

SINCLAIR RESEARCH LTD.

```

ZZZZZZZ X      X
      Z    X    X
      Z    X X
      Z    X
      Z    X X
      Z    X X
ZZZZZZZ X      X

```

```

SSSSS PPPPP PPPPP EEEEEEE CCCCC TTTTTTT RRRRRR U      U M      M
S      S P      P E      C      C      T      R      R U      U MM     MM
S      P      P E      C      C      T      R      R U      U M M M M
SSSSS PPPPP EEEEEEE C      C      T      RRRRRR U      U M      M
      S P      E      C      C      T      R      R U      U M      M
S      S P      E      C      C      T      R      R U      U M      M
SSSSS P      EEEEEEE CCCCC T      R      R UUUUU M      M

```

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ BASIC

СТИВЕН ВИКЕРС, ПОД РЕДАКЦИЕЙ РОБИНА БРЕДБИРА.
ПЕРЕВОД НА РУССКИЙ ЯЗЫК 1987.

ПЕЧАТАЕТСЯ ПО
THE LEAGRAVE PRND LONDON, GREAT BRITAIN
THIRD EDITION 1983.

ОГЛАВЛЕНИЕ

=====

СТР

ГЛАВА 1.	ВВЕДЕНИЕ.....	2
ГЛАВА 2.	ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ БЕЙСИК.....	4
ГЛАВА 3.	УСЛОВИЯ.....	8
ГЛАВА 4.	ЦИКЛЫ.....	8
ГЛАВА 5.	ПОДПРОГРАММЫ.....	10
ГЛАВА 6.	ОПЕРАТОРЫ READ, DATA, И RESTORE.....	10
ГЛАВА 7.	АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ.....	12
ГЛАВА 8.	СТРОКИ СИМВОЛОВ.....	13
ГЛАВА 9.	ФУНКЦИИ.....	14
ГЛАВА 10.	МАТЕМАТИЧЕСКИЕ ФУНКЦИИ.....	17
ГЛАВА 11.	СЛУЧАЙНЫЕ ЧИСЛА.....	19
ГЛАВА 12.	МАССИВЫ.....	20
ГЛАВА 13.	ЛОГИЧЕСКИЕ ОПЕРАЦИИ.....	21
ГЛАВА 14.	НАБОР СИМВОЛОВ.....	22
ГЛАВА 15.	ДОПОЛНИТЕЛЬНЫЕ ДЕЙСТВИЯ ОБ ОПЕРАТОРАХ "PRINT" И "INPUT".....	26
ГЛАВА 16.	ЦВЕТА.....	28
ГЛАВА 17.	ГРАФИКА.....	33
ГЛАВА 18.	УКАЗАНИЯ.....	36
ГЛАВА 19.	ПРОГРАММИРОВАНИЕ ЗВУКОВ.....	37
ГЛАВА 20.	ВНЕШНЯЯ ПАМЯТЬ НА МАГНИТНОЙ ЛЕНТЕ.....	39
ГЛАВА 21.	УСТРОЙСТВА ПЕЧАТИ.....	42
ГЛАВА 22.	ДРУГОЕ ПЕРИФЕРИЙНОЕ ОБОРУДОВАНИЕ.....	43
ГЛАВА 23.	ВВОД И ВЫВОД.....	43
	НАЗНАЧЕНИЕ ВЫВОДОВ МРУ Z80.....	45
ГЛАВА 24.	ПАМЯТЬ.....	47
ГЛАВА 25.	СИСТЕМНЫЕ ПЕРЕМЕННЫЕ.....	52
ГЛАВА 26.	ИСПОЛЬЗОВАНИЕ МАШИННЫХ КОДОВ.....	55
ПРИЛОЖЕНИЕ А.	ПОЛНЫЙ НАБОР СИМВОЛОВ.....	59
ПРИЛОЖЕНИЕ В.	СООБЩЕНИЯ.....	64
ПРИЛОЖЕНИЕ С.	(ЧАСТЬ 1) ОПИСАНИЕ МИКРОКОМПЬЮТЕРА ZX SPECTRUM.....	68
ПРИЛОЖЕНИЕ С.	(ЧАСТЬ 2) ЯЗЫК ПРОГРАММИРОВАНИЯ БЕЙСИК.....	72
ПРИЛОЖЕНИЕ Д.	ПРИМЕРЫ ПРОГРАММЫ.....	86
ПРИЛОЖЕНИЕ Е.	ШЕСТНАДЦАТИРИЧНАЯ И ДВОИЧНАЯ СИСТЕМЫ СЧИСЛЕНИЯ.....	91
УКАЗАТЕЛЬ КОМАНД БЕЙСИКА.....		92
ТЕКСТОВЫЙ РЕДАКТОР TASWORD II.....		95
АДРЕСА ПОДПРОГРАММ.....		101
ЯЗЫК ПРОГРАММИРОВАНИЯ PASCAL4 V 1.4.....		103
HISOFT C COMPILER V 1.1.....		119
КОДИРОВКА СИМВОЛОВ И КОМАНД.....		151
НАБОР КОМАНД МРУ Z80.....		157
СПИСОК КОМПЛЕКТА ZX-SPECTRUM.....		167

ГЛАВА 1
ВВЕДЕНИЕ
=====

ЕСЛИ ВЫ ЧИТАЕТЕ ЭТУ КНИГУ ВПЕРВЫЕ ИЛИ ОТКРЫЛИ ЕЕ НА ЭТОЙ СТРАНИЦЕ, ТО ВЫ ДОЛЖНЫ ИМЕТЬ ПРЕДСТАВЛЕНИЕ О ТОМ, ЧТО КОМАНДЫ БЕЙСИКА ВЫПОЛНЯЮТСЯ НЕПОСРЕДСТВЕННО (ИНТЕРПРЕТАЦИЯ), ОПЕРАТОРЫ ИМЕЮТ НОМЕР СТРОКИ И СОХРАНЯЮТСЯ В ПАМЯТИ КОМПЬЮТЕРА. ВЫ ДОЛЖНЫ ТАКЖЕ ПРЕДСТАВЛЯТЬ СЕБЕ, ЧТО ТАКИЕ КОМАНДЫ КАК PRINT, LET, И INPUT, ИСПОЛЗУЮТСЯ ВО ВСЕХ КОМПЬЮТЕРАХ, ИМЕЮЩИХ БЕЙСИК, А ТАКИЕ КОМАНДЫ, КАК BORDER, PAPER И VIEW, ИСПОЛЗУЮТСЯ ТОЛЬКО В ZX SPECTRUM.

ИЗУЧЕНИЕ БЕЙСИКА НАЧНЕМ С ПОВТОРЕНИЙ НЕКОТОРЫХ МОМЕНТОВ, ИЗЛОЖЕННЫХ ВО ВВОДНОЙ КНИГЕ, НО РАССМОТРИМ ИХ ЗНАЧИТЕЛЬНО БОЛЕЕ ПОДРОБНО, УЯСНИВ СЕБЕ, ЧТО МОЖНО ДЕЛАТЬ, А ЧЕГО НЕЛЬЗЯ.

ЧТО БЫ ВЫ НИ ЧИТАЛИ, СТАРАЙТЕСЬ В СВОЕЙ ДЕЯТЕЛЬНОСТИ ИСПОЛЬЗОВАТЬ КОМПЬЮТЕР. ЕСЛИ У ВАС ВОЗНИК ВОПРОС "ЧТО БУДЕТ, ЕСЛИ Я СДЕЛАЮ ТАК И ТАК?", ТОГДА ОТВЕТ ОЧЕНЬ ПРОСТ: ВВЕДИТЕ ЭТИ ФРАЗЫ В КОМПЬЮТЕР И ВЫ УВИДИТЕ САМИ.

ВСЯКИЙ РАЗ, КОГДА В ЭТОЙ КНИГЕ ВЫ ВОСТРЕТИТЕ ПРЕДЛОЖЕНИЕ ЧТО-НИБУДЬ ВВЕСТИ В КОМПЬЮТЕР И ВЫПОЛНИТЬ НА НЕМ, СПРАШИВАЙТЕ СЕБЯ: "ЧТО Я МОГУ СДЕЛАТЬ ВМЕСТО ЭТОГО?" И ПРОБУЙТЕ ЭТО СДЕЛАТЬ. ЧЕМ БОЛЬШЕ СОБСТВЕННЫХ ПРОГРАММ ВЫ НАПИШЕТЕ, ТЕМ ЛУЧШЕ ВЫ БУДЕТЕ ПОНИМАТЬ, КАК РАБОТАЕТ КОМПЬЮТЕР.

В КОНЦЕ ЭТОЙ КНИГИ ИМЕЕТСЯ НЕСКОЛЬКО ПРИЛОЖЕНИЙ, ОНИ СОДЕРЖАТ СВЕДЕНИЯ ПО ОРГАНИЗАЦИИ ПАМЯТИ, ПО ОПЕРАЦИЯМ С ЧИСЛАМИ, А ТАКЖЕ НЕСКОЛЬКО ПРИМЕРОВ ПРОГРАММ, ИЛЛЮСТРИРУЮЩИХ ВОЗМОЖНОСТИ SPECTRUM.

КЛАВИАТУРА
=====

В ZX SPECTRUM КЛАВИШИ СОДЕРЖАТ НЕ ТОЛЬКО ОДИНОЧНЫЕ СИМВОЛЫ (БУКВЫ, ЦИФРЫ, И Т.Д.), НО ТАКЖЕ СОСТАВНЫЕ СИМВОЛЫ (КЛЮЧЕВЫЕ СЛОВА, НАЗВАНИЯ, ФУНКЦИИ И Т.П.).

ДЛЯ ТОГО, ЧТОБЫ РЕАЛИЗОВАТЬ ВСЕ ЭТИ ФУНКЦИИ И КОМАНДЫ, НЕКОТОРЫЕ КЛАВИШИ КЛАВИАТУРЫ ИМЕЮТ 5 И БОЛЕЕ ЗНАЧЕНИЙ, ПОЛУЧАЕМЫХ ЛИБО ПУТЕМ ВЫБОРА СООТВЕТСТВУЮЩЕГО РЕГИСТРА (Т.Е. ПУТЕМ НАЖАТИЯ КЛАВИШ CAPS SHIFT ИЛИ SYMBOL SHIFT ОДНОВРЕМЕННО С КАКОЙ-ЛИБО НЕОБХОДИМОЙ КЛАВИШЕЙ), ЛИБО ПУТЕМ ПЕРЕВОДА КОМПЬЮТЕРА В ОДИН ИЗ ВОЗМОЖНЫХ РЕЖИМОВ РАБОТЫ.

РЕЖИМ ИНДИЦИРУЕТСЯ КУРСОРОМ - МЕРЦАЮЩЕЙ БУКВОЙ, КОТОРАЯ ПОКАЗЫВАЕТ, ГДЕ БУДЕТ ПОЯВЛЯТЬСЯ НА ЭКРАНЕ СЛЕДУЮЩИЙ НАБИРАЕМЫЙ СИМВОЛ.

РЕЖИМ [K] АВТОМАТИЧЕСКИ ЗАМЕНЯЕТ РЕЖИМ [L], КОГДА КОМПЬЮТЕР ОЖИДАЕТ КОМАНДУ ИЛИ ПРОГРАММНУЮ СТРОКУ (ОТЛИЧАЮЩУЮСЯ ОТ ВВОДА ДАННЫХ) И С ЭТОЙ ПОЗИЦИИ В СТРОКЕ КУРСОРОМ УКАЗЫВАЕТСЯ, ЧТО ОЖИДАЕТСЯ ВВОД КЛЮЧЕВОГО СЛОВА ИЛИ СТРОКИ. ЭТО ОТНОСИТСЯ К НАЧАЛУ СТРОКИ ИЛИ ЗНАКОМЕСТУ СРАЗУ ЖЕ ПОСЛЕ ОПЕРАТОРА THEN, ИЛИ К ЗНАКОМЕСТУ СРАЗУ ЖЕ ПОСЛЕ ДВОЕТОЧИЯ ";" (ЗА ИСКЛЮЧЕНИЕМ ДВОЕТОЧИЯ В СТРОКЕ), ЕСЛИ НЕ ИЗМЕНЕН РЕЖИМ, ТО НАЖАТИЕ СЛЕДУЮЩЕЙ КЛАВИШИ БУДЕТ ИНТЕРПРЕТИРОВАТЬСЯ КАК КЛЮЧЕВОЕ СЛОВО, НАПИСАННОЕ НА КЛАВИАТУРЕ, ЛИБО КАК ЦИФРА.

РЕЖИМ КУРСОРА [L] (ДЛЯ БУКВ) ПОЯВЛЯЕТСЯ ОБЫЧНО ВО ВСЕХ ДРУГИХ СЛУЧАЯХ. ЕСЛИ ОН НЕ МЕНЯЕТСЯ, ТО НАЖАТИЕ СЛЕДУЮЩЕЙ КЛАВИШИ БУДЕТ ИНТЕРПРЕТИРОВАНО КАК ОСНОВНОЙ СИМВОЛ НА КЛАВИШЕ, В БОЛЬШИНСТВЕ СЛУЧАЕВ ЭТО БУКВЫ.

И В [K] И В [L] РЕЖИМАХ ОДНОВРЕМЕННОЕ НАЖАТИЕ КЛАВИШИ

SYMBOL SHIFT и какой - либо другой клавиши воспринимается как вспомогательный символ, изображенный на клавише, а в случае нажатия клавиши CAPS SHIFT с цифровой клавишей - как управляющая функция, написанная на цифровой клавише.

Нажатие клавиши CAPS SHIFT с другими клавишами в режиме курсора [K] не влияет на ключевые слова, а в режиме курсора [L] вызывает появление заглавных букв.

Режим курсора [C] (для заглавных букв) - это вариант режима [L], в котором все буквы появляются на экране как заглавные.

Нажатие клавиши CAPS LOCK приводит к смене курсора [L] на [C] или наоборот.

Режим курсора [E] (расширение) используется для получения дополнительных символов (обычно знаков), курсор [E] появляется после одновременного нажатия обеих клавиш смены режима и сохраняется до нажатия какой - нибудь другой клавиши. В этом режиме нажатие клавиши дает один символ или знак, если режим сохраняется, и другой, если одновременно нажата одна из клавиш смены режима.

Одновременное нажатие цифровых клавиш с клавишей смены режима SYMBOL SHIFT вызывает появление знака, в противном случае они дают появление символов, управляющих цветом.

Режим курсора [G] возникает после нажатия клавиши GRAPHICS (CAPS SHIFT и 9) и сохраняется до тех пор, пока не будет нажата клавиша CAPS SHIFT одна или совместно с 9.

Цифровые клавиши дают также графические символы, за исключением GRAPHICS или DELETE. Каждая из буквенных клавиш, кроме V, W, X, Y и Z может вызывать появление определенных пользователем графических символов.

Если некоторая клавиша удерживается более чем 2 или 3 сек. это вызывает повторение ее действия. Ввод с клавиатуры осуществляется в нижнюю половину экрана. Каждый символ (или группа символов для ключевых слов) появляется перед курсором, сам курсор может перемещаться клавишами:

влево - CAPS SHIFT и 5

вправо - CAPS SHIFT и 6 и т.д.

Символ перед курсором может быть удален командой DELETE (CAPS SHIFT и 0).

Примечание: целая строка может быть удалена вводом EDIT (CAPS SHIFT и 1) и последующим нажатием клавиши ENTER.

При нажатии ENTER строка, набранная в нижней части экрана, либо выполняется как команда, либо выводится как очередная строка в программу, либо используется как список данных для INPUT-ввода.

Если же она содержит синтаксические ошибки, то ошибочное место указывается мерцающим знаком вопроса (?).

Когда вводятся строки программы, то листинг отображается в верхней половине экрана. Последняя введенная строка называется текущей и указывается символом (>), его можно перемещать ниже или выше, используя клавиши CAPS SHIFT и 6 или CAPS SHIFT и 7 соответственно. Если введено EDIT (CAPS SHIFT и 1), то текущая строка переносится в нижнюю часть экрана, где она может редактироваться. При выполнении команды и программы вывод осуществляется в верхнюю часть экрана и сохраняется до ввода строки программы, либо нажатия клавиши ENTER при наличии пустой строки, либо нажатия клавиш перемещения курсора вверх, вниз.

В нижнюю часть экрана выводятся также сообщения об ошибках, которые сохраняются там до нажатия любой клавиши (это индицируется переходом в режим [K]).

В ОПРЕДЕЛЕННЫХ СОСТОЯНИЯХ ЛАВИШИ CAPS SHIFT И SPACE ДЕЙСТВУЮТ КАК BREAK, ОСТАНАВЛИВАЯ КОМПЬЮТЕР С ВЫДАЧЕЙ СООБЩЕНИЯ "D" ИЛИ "L" ЭТО РАСПОЗНАЕТСЯ;
А) В КОНЦЕ ВЫПОЛНЯЕМОГО Оператора ПРОГРАММЫ
Б) ПОСЛЕ ЗАВЕРШЕНИЯ ОПЕРАЦИИ НА ПРИНТЕРЕ ИЛИ МАГНИТОФОНЕ

ЭКРАН ТЕЛЕВИЗОРА

=====

ЭКРАН СОДЕРЖИТ 24 СТРОКИ ПО 32 СИМВОЛА В КАЖДОЙ И ДЕЛИТСЯ НА ДВЕ ЧАСТИ. ВЕРХНЯЯ ЧАСТЬ ЭКРАНА (22 СТРОКИ) СЛУЖИТ ДЛЯ ОТОБРАЖЕНИЯ ЛИСТИНГА И ВЫВОДА РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММЫ, КОГДА ВЕРХНЯЯ ЧАСТЬ ЭКРАНА ЗАПОЛНЯЕТСЯ ПОЛНОСТЬЮ, ОН СВОРАЧИВАЕТСЯ НА ОДНУ СТРОКУ, КОМПЬЮТЕР ОСТАНАВЛИВАЕТСЯ С ВЫДАЧЕЙ СООБЩЕНИЯ: "SCROLL?". ОТВЕТ: N, SPACE - ИЛИ STOP ВЫЗОВЕТ ОСТАНОВ ПРОГРАММЫ С ВЫДАЧЕЙ СООБЩЕНИЯ: "D BREAK - CONT REPEAT", НАЖАТИЕ ЛЮБОЙ ДРУГОЙ ИЗ КЛАВИШ РАЗРЕШАЕТ СВЕРТКУ ЭКРАНА.

НИЖНЯЯ ЧАСТЬ ЭКРАНА ИСПОЛЬЗУЕТСЯ ДЛЯ ВВОДА КОМАНД, СТРОК ПРОГРАММЫ И ВВОДИМЫХ ДАННЫХ, А ТАКЖЕ ДЛЯ ВЫВОДА СООБЩЕНИЙ СИСТЕМЫ (ПОДРОБНЕЕ СМ. ПРИЛОЖЕНИЕ С ЧАСТЬ 1).

ГЛАВА 2

ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ БЕЙСИК

=====

КРАТКОЕ СОДЕРЖАНИЕ: ПРОГРАММЫ, НОМЕРА СТРОК, РЕДАКТИРОВАНИЕ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ КЛАВИШ:

<ВВЕРХ>, <ВНИЗ>, EDIT,
КОМАНДЫ RUN, LIST, GO TO, CONTINUE, INPUT,
NEW, REM, PRINT, STOP В INPUT ДАННЫХ, BREAK

НАБЕРИТЕ ДВЕ СТРОКИ ПРОГРАММЫ ВЫЧИСЛЕНИЯ СУММЫ ДВУХ ЧИСЕЛ
10 LET A=10
20 PRINT A
ТАК, ЧТОБЫ НА ЭКРАНЕ ПОЯВИЛОСЬ:

```
10 LET A=10  
20 > PRINT A
```

[K]

СТРОКА ПРОГРАММЫ ДОЛЖНА НАЧИНАТЬСЯ С НОМЕРА, КОТОРЫЙ НЕ ЗАПИСЫВАЕТСЯ В ПАМЯТЬ, А СЛУЖИТ ЛИШЬ ДЛЯ УКАЗАНИЯ ПОРЯДКА СЛЕДОВАНИЯ СТРОК В ПРОГРАММЕ, ЧТО ВАЖНО ПРИ ЕЕ ВЫПОЛНЕНИИ.

ТЕПЕРЬ НАБЕРИТЕ:

```
15 LET B=15
```

И ВВЕДИТЕ. ПОДОБНОГО НЕВОЗМОЖНО БЫЛО БЫ СДЕЛАТЬ, ЕСЛИ БЫ НУМЕРАЦИЯ НАЧИНАЛАСЬ С 1 И 2, А НЕ С 10 И 20, КАК В НАШЕМ СЛУЧАЕ, (НОМЕР МОЖЕТ БЫТЬ В ИНТЕРВАЛЕ ОТ 1 ДО 9999).

ДОПУСТИМ, ТЕПЕРЬ ВАМ ПОНАДОБИЛОСЬ ИЗМЕНИТЬ СТРОКУ 20 НА СЛЕДУЮЩУЮ:

```
20 PRINT A+B
```

ЭТО МОЖНО СДЕЛАТЬ, ИСПОЛЬЗУЯ КОМАНДУ EDIT.

СИМВОЛ [>] В СТРОКЕ 15 НАЗЫВАЕТСЯ ПРОГРАММНЫМ КУРСОРОМ, А СТРОКА НА КОТОРУЮ ОН УКАЗЫВАЕТ, НАЗЫВАЕТСЯ ТЕКУЩЕЙ, ЭТО ОБЫЧНО ПОСЛЕДНЯЯ ВВЕДЕННАЯ СТРОКА, НО ВЫ ИМЕЕТЕ ВОЗМОЖНОСТЬ ПЕРЕМЕСТИТЬ ПРОГРАММНЫЙ КУРСОР ВЫШЕ ИЛИ НИЖЕ, ИСПОЛЬЗУЯ СООТВЕТСТВУЮЩИЕ КЛАВШИИ УПРАВЛЕНИЯ КУРСОРОМ: УСТАНОВИТЕ ЕГО В СТРОКУ 20. КОГДА ВЫ НАЖМЕТЕ КЛАВИШУ EDIT, ТО В НИЖНЕЙ ЧАСТИ ЭКРАНА ПОЯВИТСЯ КОПИЯ ТЕКУЩЕЙ СТРОКИ, В НАШЕМ СЛУЧАЕ КОПИЯ СТРОКИ 20. НАЖМИТЕ И УДЕРЖИВАЙТЕ КЛАВИШУ ПЕРЕМЕЩЕНИЯ КУРСОРА ВПРАВО ДО ТЕХ ПОР, ПОКА КУРСОР [L] НЕ ПЕРЕМЕСТИТСЯ НА КОНЕЦ ОПЕРАТОРА И ЗАТЕМ ВВЕДИТЕ: "+B" (БЕЗ НАЖАТИЯ ENTER), СТРОКА В НИЖНЕЙ ЧАСТИ ЭКРАНА ПРИМЕТ ВИД:

```
20 PRINT A+B
```

ТЕПЕРЬ НАЖМИТЕ ENTER, - ЭТО ВЫЗОВЕТ ЗАМЕНУ СТАРОЙ СТРОКИ 20 НА НОВУЮ, ЗАПИСАННУЮ В НИЖНЕЙ ЧАСТИ ЭКРАНА, НА ЭКРАНА ЭТО БУДЕТ ВЫГЛЯДЕТЬ ТАК:

```
10 LET A=10
15 LET B=15
20 > PRINT A+B
```

[K]

ЗАПУСТИТЕ ПРОГРАММУ, НАЖАВ RUN И ENTER И ПОЛУЧИТЕ НА ЭКРАНЕ СУММУ.

ВЫПОЛНИТЕ ПРОГРАММУ PRINT A,B, ПЕРЕМЕННЫЕ СОХРАНЯЮТСЯ ДАЖЕ ПОСЛЕ ЗАВЕРШЕНИЯ ПРОГРАММЫ.

ЕСТЬ ЕЩЕ ОДНО ПРИМЕНЕНИЕ КОМАНДЫ EDIT. ДОПУСТИМ, ВАМ НАДО УДАЛИТЬ ВСЮ СТРОКУ НАБРАННУЮ В НИЖНЕЙ ЧАСТИ ЭКРАНА, ДЛЯ ЭТОГО ВЫ МОЖЕТЕ НАЖАТЬ И УДЕРЖИВАТЬ ДО КОНЦА СТРОКИ КЛАВИШУ DELETE, НО МОЖНО СДЕЛАТЬ БЫСТРЕЕ: НАЖАТЬ EDIT, ЧТО ВЫЗОВЕТ КОПИРОВАНИЕ ТЕКУЩЕЙ СТРОКИ В НИЖНЮЮ ЧАСТЬ ЭКРАНА, ЗАТЕМ НАЖАТЬ ENTER, СТРОКА ЗАМЕНИТ ТАКУЮ ЖЕ В ПРОГРАММЕ, А НИЖНЯЯ ЧАСТЬ ЭКРАНА ОЧИСТИТСЯ.

ВВЕДИТЕ СТРОКУ:

```
12 LET B=B
```

ТЕПЕРЬ ДЛЯ УДАЛЕНИЯ ЭТОЙ СТРОКИ НАБЕРИТЕ:

```
12 ( И ЗАТЕМ ENTER )
```

ПРОГРАММНЫЙ КУРСОР СТАНЕТ МЕЖДУ СТРОКАМИ 10 И 15, НО КЛАВИШАМИ УПРАВЛЕНИЯ КУРСОРОМ ВЫ МОЖЕТЕ УСТАНОВИТЬ ЕГО В ЛЕВУЮ СТРОКУ.

ЕЩЕ РАЗ ВЫПОЛНИТЕ 12 И ENTER, КУРСОР СНОВА УСТАНОВИТСЯ МЕЖДУ СТРОКАМИ 10 И 15. ТЕПЕРЬ НАЖМИТЕ EDIT И СТРОКА 15 БУДЕТ СКОПИРОВАНА В НИЖНЮЮ ЧАСТЬ ЭКРАНА. ОПЕРАТОР EDIT КОПИРУЕТ ВНИЗ СТРОКУ, СЛЕДУЮЩУЮ ЗА СТРОКОЙ С НОВЫМ НОМЕРОМ. НАЖМИТЕ ENTER ДЛЯ ОЧИСТКИ НИЖНЕЙ ЧАСТИ ЭКРАНА.

ТЕПЕРЬ ВВЕДИТЕ:

```
30 ( И ЗАТЕМ ENTER )
```

ПРОГРАММНЫЙ КУРСОР УСТАНОВИТСЯ ПОСЛЕ КОНЦА ПРОГРАММЫ. ЕСЛИ ВЫ ТЕПЕРЬ НАЖМЕТЕ EDIT, ВНИЗ БУДЕТ ПЕРЕНЕСЕНА СТРОКА 20.

И, НАКОНЕЦ, ВЫПОЛНИТЕ КОМАНДУ:

```
LIST 15
```


ТЕПЕРЬ ВЫ УВИДИТЕ НА ЭКРАНЕ: 15 LET B=15
20 PRINT A+B

СТРОКА 15 НЕ ОТОБРАЖАЕТСЯ НА ЭКРАНЕ, НО ОНА СОХРАНЯЕТСЯ В ВАШЕЙ ПРОГРАММЕ, ВЫ МОЖЕТЕ УБЕДИТЬСЯ В ЭТОМ, НАЖАВ ENTER.

КОМАНДА LIST 15 УКАЗЫВАЕТ, ЧТО НАДО ОТОБРАЗИТЬ ЛИСТИНГ СО СТРОКИ С НОМЕРОМ 15 И УСТАНОВЛИВАЕТ В ЭТУ СТРОКУ ПРОГРАММНЫЙ КУРСОР, ЭТО БЫВАЕТ УДОБНО ПРИ ПРОСМОТРЕ ОЧЕНЬ БОЛЬШИХ ПРОГРАММ.

ДРУГОЕ НАЗНАЧЕНИЕ НОМЕРОВ СТРОК - СЛУЖИТЬ ИМЕНЕМ ОПЕРАТОРА ПРИ ССЫЛКЕ К НИМУ ИЗ ДРУГОГО МЕСТА ПРОГРАММЫ (В GO TO N).

КОМАНДА LIST БЕЗ ОПЕРАНДОВ ВЫДАТ ЛИСТИНГ С ПЕРВОЙ СТРОКИ.

КОМАНДА NEW ОЧИЩАЕТ ПАМЯТЬ КОМПЬЮТЕРА ОТ СТАРЫХ ПРОГРАММ И ПЕРЕМЕННЫХ.

ТЕПЕРЬ ВЫПОЛНИМ ПРОГРАММУ, ПЕРЕВОДЯЩУЮ ЗНАЧЕНИЯ ТЕМПЕРАТУРЫ В ГРАДУСАХ ПО ФАРЕНГЕЙТУ В ГРАДУСЫ ПО ЦЕЛЬСИУ:

```
10 REM TEMPERATURE CONVERSION
20 PRINT "DEG F", "DEG C"
30 PRINT
40 INPUT "ENTER DEG F ", F
50 PRINTF ( F-32)=5/9
60 GO TO 40
```

ВЫ УВИДИТЕ, ЧТО ЗАГОЛОВОК ВЫВОДИТСЯ В СТРОКЕ 20, И У ВАС ВОЗНИКНЕТ ВОПРОС, ЧТО ЖЕ ДЕЛАЕТ СТРОКА 10? КОМПЬЮТЕР ИГНОРИРУЕТ ЭТУ СТРОКУ. ЭТО КОМЕНТАРИИ (РЕМАРК ИЛИ RENAMBER). ВСЕ ЧТО СЛЕДУЕТ ПОСЛЕ РЕМАРК КОМПЬЮТЕРОМ ИГНОРИРУЕТСЯ ДО КОНЦА СТРОКИ.

ВЫЧИСЛЕНИЯ ДОХОДЯТ ДО СТРОКИ 40 И КОМПЬЮТЕР ПЕРЕХОДИТ В ОЖИДАНИЕ ВВОДА ВАМИ ЗНАЧЕНИЯ ПЕРЕМЕННОЙ F, ВЫ МОЖЕТЕ ВВЕСТИ ЭТО ЗНАЧЕНИЕ В РЕЖИМЕ (L), НАВЕРИТЕ ЧИСЛО И НАЖМИТЕ ENTER, КОМПЬЮТЕР ВЫВЕДЕТ РЕЗУЛЬТАТ И СНОВА ПЕРЕИДЕТ В ОЖИДАНИЕ СЛЕДУЮЩЕГО ЧИСЛА, ЭТОТ ПЕРЕХОД ОБЕСПЕЧИВАЕТСЯ В СТРОКЕ 60 В ОПЕРАТОРЕ GO TO 40.

ЕСЛИ НА ЗАПРОС ОЧЕРЕДНОГО ЧИСЛА ОТВЕТИТЬ STOP, ТО КОМПЬЮТЕР ОСТАНОВИТСЯ С ВЫДАЧЕЙ СООБЩЕНИЯ: "N STOP IN INPUT LINE 40:1" КОТОРОЕ ПОЯСНЯЕТ ПРИЧИНУ ОСТАНОВА И МЕСТО ОСТАНОВА (ПЕРВЫЙ ОПЕРАТОР В СТРОКЕ 40).

ЕСЛИ ТЕПЕРЬ ВЫ ЖЕЛАЕТЕ ВНОВЬ ПРОДОЛЖИТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ, ТО ВЫВЕДИТЕ CONTINUE И КОМПЬЮТЕР ЗАПРОСИТ ОЧЕРЕДНОЕ ЧИСЛО

ПРИ ИСПОЛЬЗОВАНИИ CONTINUE КОМПЬЮТЕР ЗАПОМИНАЕТ (ДО ВЫДАЧИ "OK") НОМЕР ПОСЛЕДНЕЙ ВЫПОЛНЯЕМОЙ СТРОКИ И ПРОДОЛЖАЕТ ВЫПОЛНЕНИЕ ИМЕННО С ЭТОЙ СТРОКИ.

ПРОСМОТРИТЕ ВНИМАТЕЛЬНО ОПЕРАТОР PRINT В СТРОКЕ 50 ЗАПЯТАЯ В НЕМ ОЧЕНЬ ВАЖНА. ЗАПЯТЫЕ В ОПЕРАТОРЕ PRINT ИСПОЛЬЗУЮТСЯ ДЛЯ УКАЗАНИЯ ТОГО, ЧТО ВЫВОД, СЛЕДУЮЩИЙ ПОСЛЕ ЗАПЯТОЙ, ДОЛЖЕН ПРОДОЛЖАТЬСЯ ЛИБО С ЛЕВОГО КРАЯ ЭКРАНА, ЛИБО С ЕГО СЕРЕДИНЫ, В ЗАВИСИМОСТИ ОТ ТОГО, КАКАЯ ЭТО ПО ПОРЯДКУ ЗАПЯТАЯ В ДАННОМ ОПЕРАТОРЕ. ТАК В СТРОКЕ 50 ЗАПЯТАЯ ПРЕДПИСЫВАЕТ ВЫВОДИТЬ ЗНАЧЕНИЕ ТЕМПЕРАТУРЫ В ГРАДУСАХ ЦЕЛЬСИЯ С СЕРЕДИНЫ ЭКРАНА.

ЕСЛИ ИСПОЛЬЗОВАТЬ В ОПЕРАТОРЕ PRINT ВМЕСТО ЗАПЯТОЙ ТОЧКУ С ЗАПЯТОЙ, ТО ОЧЕРЕДНЫЕ ДАННЫЕ БУДУТ ВЫВОДИТСЯ НЕПОСРЕДСТВЕННО ПОСЛЕ ПРЕДЫДУЩИХ.

ОПЕРАТОР В СТРОКЕ 30 ВЫВОДИТ ЧИСТУЮ СТРОКУ.

ОПЕРАТОР PRINT ВСЕГДА НАЧИНАЕТ ВЫВОД С НАЧАЛА СЛЕДУЮЩЕЙ СТРОКИ, НО ЭТО МОЖНО ИЗМЕНИТЬ, ПОСТАВИВ В КОНЦЕ ПРЕДЫДУЩЕГО ОПЕРАТОРА PRINT ЗАПЯТУЮ ИЛИ ТОЧКУ С ЗАПЯТОЙ ;

```
50 PRINT F ,  
60 PRINT F
```

НЕ ПУТАЙТЕ ЭТИ ЗНАКИ С ДВОЕТОЧИЕМ ':', КОТОРОЕ ИСПОЛЬЗУЕТСЯ ТОЛЬКО ДЛЯ РАЗДЕЛЕНИЯ РАЗНЫХ ОПЕРАТОРОВ В ОДНОЙ СТРОКЕ.

ТЕПЕРЬ НАБЕРЕМ ЕЩЕ НЕСКОЛЬКО ПРОГРАММНЫХ СТРОК;

```
100 REM THIS POLITE PROGRAM REMEMBER YOUR NAME
```

```
110 INPUT N*
```

```
120 PRINT "HELLO" N* "!"
```

```
130 GO TO 110
```

ЭТА ПРОГРАММА НИКАК НЕ СВЯЗАНА С НАБРАННОЙ НАМИ РАНЕЕ ПРОГРАММОЙ, НО ИХ ОБЕ МОЖНО ДЕРЖАТЬ В ПАМЯТИ КОМПЬЮТЕРА ОДНОВРЕМЕННО.

ДЛЯ ТОГО, ЧТОБЫ ВЫПОЛНИТЬ ПОСЛЕДНЮЮ ПРОГРАММУ, НАДО ВВЕСТИ КОМАНДУ; RUN 100

ЭТА ПРОГРАММА ВВОДИТ СТРОКУ СИМВОЛОВ, ЧТО ДОЛЖНО УКАЗЫВАТЬ СТРОКОВЫМИ КОВЫЧКАМИ. ЕСЛИ ИХ ОПУСТИТЬ, ТО КОМПЬЮТЕР ПОПЫТАЕТСЯ НАЙТИ ПЕРЕМЕННУЮ С ТАКИМ ИМЕНЕМ И ИСПОЛЬЗОВАТЬ ЕЕ ЗНАЧЕНИЕ В КАЧЕСТВЕ INPUT-ДАННЫХ. НАПРИМЕР, ОТВЕЬТЕ ПРОГРАММЕ:

N* (УДАЛИВ КАВЫЧКИ)

ЭТО СДЕЛАЕТ ОПЕРАТОР INPUT В СТРОКЕ 110 ПОДОБНЫМ ОПЕРАТОРУ

```
LET N*=N*
```

ЕСЛИ ВЫ РЕШИЛИ ВВЕСТИ STOP ПОД СТРОКОВЫЙ ВВОД, ТО ДОЛЖНЫ УСТАНОВИТЬ КУРСОР В НАЧАЛО СТРОКИ, ИСПОЛЬЗУЯ КЛАВИШУ УПРАВЛЕНИЯ КУРСОРОМ < ВЛЕВО>.

ДЕЙСТВИЕ КОМАНДЫ 'RUN 100' ПОДОБНО ДЕЙСТВИЮ ОПЕРАТОРА 'GO TO', НО ИМЕЮТСЯ И ОТЛИЧИЯ. RUN 100 ОЧИЩАЕТ ВСЕ ПЕРЕМЕННЫЕ И ЭКРАН И ПОСЛЕ ЭТОГО ВЫПОЛНЯЕТ GO TO 100. ДРУГОЕ ОТЛИЧИЕ В ТОМ, ЧТО ВЫ МОЖЕТЕ УКАЗАТЬ RUN БЕЗ НОМЕРА СТРОКИ, И ТОГДА ВЫПОЛНЕНИЕ НАЧНЕТСЯ С ПЕРВОЙ СТРОКИ, А ОПЕРАТОР GO TO ВСЕГДА ДОЛЖЕН СОДЕРЖАТЬ НОМЕР СТРОКИ.

ОБЕ ПРИВЕДЕННЫЕ ПРОГРАММЫ ОСТАНАВЛИВАЛИСЬ НАМИ ВВОДОМ КОМАНДЫ STOP, НО МОГУТ БЫТЬ ПРОГРАММЫ, КОТОРЫЕ НЕВОЗМОЖНО ОСТАНОВИТЬ ПОДОБНЫМ ОБРАЗОМ, НАПРИМЕР:

```
200 GO TO 200
```

```
RUN 200
```

ОСТАНОВИТЬ ЭТУ ПРОГРАММУ МОЖНО, ЕСЛИ НАЖАТЬ КЛАВИШИ CAPS SHIFT И SPACE. ЭТО ВЫЗОВЕТ ВВОД ПРОГРАММЫ BREAK, КОТОРАЯ ОСТАНОВИТ ВЫЧИСЛЕНИЯ С ВЫДАЧЕЙ СООБЩЕНИЯ 'L BREAK INTO PROGRAM'. КОМАНДА BREAK МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНА И ВО ВРЕМЯ ВЫПОЛНЕНИЯ ОПЕРАЦИИ НА МАГНИТОФОНЕ ИЛИ ПРИНТЕРЕ. В ЭТОМ СЛУЧАЕ ВЫДАЕТСЯ СООБЩЕНИЕ 'O BREAK CONT REPEATS'. КОМАНДА CONTINUE В ЭТОМ СЛУЧАЕ (КАК И В БОЛЬШИНСТВЕ ДРУГИХ), ВЫЗОВЕТ ПОВТОРЕНИЕ ОПЕРАТОРА, В КОТОРОМ ПРОИЗОШЕЛ ОСТАНОВ. ВВОД КОМАНДЫ CONTINUE ПОСЛЕ СООБЩЕНИЯ 'L BREAK INTO PROGRAM' ПРОДОЛЖИТ ВЫПОЛНЕНИЕ СО СЛЕДУЮЩЕГО ОПЕРАТОРА.

ЗАПУСТИТЕ ВТОРУЮ ПРОГРАММУ СНОВА, И КОГДА ОНА ЗАПРОСИТ ВВОД, ВВЕДИТЕ:

N* (УДАЛИВ ПЕРЕМЫЧКИ)

ПОСКОЛЬКУ ЗНАЧЕНИЕ N* НЕ ОПРЕДЕЛЕНО, ТО БУДЕТ ВЫДАНО СООБЩЕНИЕ: '2 VARIABLE NOT FOUND'. ЕСЛИ ТЕПЕРЬ ВЫ ВЫПОЛНИТЕ:

```
LET N*="SOMETHING DEFINITE"
```

НА ЧТО КОМПЬЮТЕР ОТВЕТИТ '3 OK,0:1'.

А ЗАТЕМ ВВЕДЕТЕ CONTINUE, ТО УВИДИТЕ, ЧТО ПРОГРАММА ЗАВЕРНИТСЯ НОРМАЛЬНО. КАК УЖЕ ОТМЕЧАЛОСЬ, СООБЩЕНИЕ 'L BREAK INTO PROGRAM' ОСОБОЕ, ТАК КАК ВЫДАЧА ПОСЛЕ НЕГО CONTINUE НЕ ВЫЗЫВАЕТ ПОВТОРЕНИЕ КОМАНДЫ, ВЫЗВАВШЕЙ ОСТАНОВ.

ВСЕ ПРИВЕДЕННЫЕ В ЭТОЙ ГЛАВЕ УТВЕРЖДЕНИЯ PRINT, LET, INPUT, RUN, LIST, GO TO, CONTINUE, NEW И REM МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ЛИБО КАК ОПЕРАТОРЫ В ПРОГРАММЕ, ЛИБО КАК КОМАНДЫ, ХОТЯ RUN, LIST, CONTINUE И REM ЧАЩЕ ИСПОЛЬЗУЮТСЯ КАК КОМАНДЫ, НО МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ И В ПРОГРАММЕ.

-8-
ГЛАВА 3
УСЛОВИЯ
=====

КРАТКОЕ СОДЕРЖАНИЕ: IF, STOP, =, >, <, <=, >=, <>

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ОПЕРАТОРОВ ПРОГРАММЫ НЕ ВСЕГДА ПРЕДСКАЗУЕМА. В ОПРЕДЕЛЕННЫХ МЕСТАХ ПРОГРАММЫ КОМПЬЮТЕР МОЖЕТ ПРИНИМАТЬ РЕШЕНИЕ О ДАЛЬНЕЙШЕМ ХОДЕ ВЫЧИСЛЕНИЯ. ОПЕРАТОР, РЕАЛИЗУЮЩИЙ ЭТО ИМЕЕТ ФОРМУ:

IF - НЕКОТОРОЕ ИСТИННОЕ ИЛИ ЛОЖНОЕ ВЫРАЖЕНИЕ,
THEN - НЕКОТОРОЕ ДЕЙСТВИЕ.

НАПРИМЕР ВЫПОЛНИТЕ КОМАНДУ NEW, А ЗАТЕМ НАБЕРИТЕ И ВЫПОЛНИТЕ ПРОГРАММУ (ЭТО ИГРА ДЛЯ ДВУХ ЧЕЛОВЕК):

```
10 REM GUESS THE NUMBER (УГАДЫВАНИЕ ЧИСЛА)
20 INPUT A:CLS
30 INPUT "GUESS THE NUMBER",B (УГАДАЙТЕ ЧИСЛО)
40 IF A=B THEN PRINT "THIS IS CORRECT":STOP
50 IF B<A THEN PRINT "THIS IS TOO SMALL, TRY AGAIN"
60 IF B>A THEN PRINT "THIS IS TOO BIG, TRY AGAIN"
70 GO TO 30
```

ЗДЕСЬ IF ОПЕРАТОР ИМЕЕТ ФОРМУ: IF УСЛОВИЕ THEN....

ГДЕ "...." - ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАТОРОВ, РАЗДЕЛЕННЫХ ДВОЕТОЧИЕМ ОБЫЧНЫМ ОБРАЗОМ. ЕСЛИ "УСЛОВИЕ" ИСТИННО, ТО ВЫПОЛНЯЮТСЯ ОПЕРАТОРЫ СЛЕДУЮЩИЕ ПОСЛЕ THEN, В ПРОТИВНОМ СЛУЧАЕ ОНИ ПРОПУСКАЮТСЯ И ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРОДОЛЖАЕТСЯ ДО СЛЕДУЮЩЕГО ОПЕРАТОРА. ПРОСЛЕДИТЕ ЗА УСЛОВИЕМ МОЖЕТ БЫТЬ СРАВНЕНИЕ ДВУХ ЧИСЕЛ ИЛИ ДВУХ СТРОК ЧИСЛА МОГУТ БЫТЬ ЛИБО РАВНЫ, ЛИБО ОДНО БОЛЬШЕ ДРУГОГО, А СТРОКИ ЛИБО РАВНЫ, ЛИБО ОДНА СЛЕДУЕТ ПОСЛЕ ДРУГОЙ В АЛФАВИТНОМ ПОРЯДКЕ ДЛЯ ЗАДАНИЯ УСЛОВИЯ ИСПОЛЪЗУЮТСЯ ОТНОШЕНИЯ:

=, <, >, <=, >=, <>

НАПРИМЕР, ВЫРАЖЕНИЯ $1 < 2$, $-2 < 1$, $-3 < 1$ ИСТИННЫ, А ВЫРАЖЕНИЯ $1 < 0$, $0 < 2$ ЛОЖНЫ. СТРОКА ПРОГРАММЫ 40 СРАВНИВАЕТ ЧИСЛА 'A' И 'B'. И, ЕСЛИ ОНИ РАВНЫ, ЗАВЕРШАЕТ РАБОТУ, ВЫПОЛНЯЯ КОМАНДУ STOP. ПРИ ЭТОМ БУДЕТ ВЫДАНО СООБЩЕНИЕ '9 STOP, STATEMENT, 30:3', ПОКАЗЫВАЮЩЕЕ, ЧТО КОМАНДА STOP БЫЛА ВЫДАНА В 3-ЕМ ОПЕРАТОРЕ В 30-И СТРОКЕ.

ЗНАКИ УСЛОВИЯ НАБИРАЮТ НА КЛАВИАТУРЕ СЛЕДУЮЩИМ ОБРАЗОМ:

```
> - SYMBOL SHIFT ВМЕСТЕ С Т - БОЛЬШЕ
< - SYMBOL SHIFT ВМЕСТЕ С R - МЕНЬШЕ
<= - SYMBOL SHIFT ВМЕСТЕ С G - МЕНЬШЕ ИЛИ РАВНО
( НЕЛЬЗЯ НАБИРАТЬ < И == )
>= - SYMBOL SHIFT ВМЕСТЕ С E - БОЛЬШЕ ИЛИ РАВНО
<> - SYMBOL SHIFT ВМЕСТЕ С W - НЕ РАВНО
```

ГЛАВА 4
• ЦИКЛЫ
=====

КРАТКОЕ СОДЕРЖАНИЕ: FOR, NEXT, TO, STEP

ДОПУСТИМ НАМ НЕОБХОДИМО СОСТАВИТЬ ПРОГРАММУ, ПОДСЧИТЫВАЮЩУЮ СУММУ ВВОДИМЫХ ПЯТИ ЧИСЕЛ. ЭТО МОЖНО БЫЛО ВЫ СДЕЛАТЬ ТАК:

```
10 LET TOTAL=0
20 INPUT A
30 LET TOTAL=TOTAL+A
40 INPUT A
50 LET TOTAL=TOTAL+A
60 INPUT A
70 LET TOTAL=TOTAL+A
80 INPUT A
```

```

90 LET TOTAL=TOTAL+A
100 INPUT A
110 LET TOTAL=TOTAL+A
120 PRINT TOTAL

```

ПОЛУЧИЛАСЬ БОЛЬШАЯ И НЕ ОЧЕНЬ ОПТИМАЛЬНАЯ ПРОГРАММА. МОЖНО РЕШИТЬ ЭТУ ЗАДАЧУ БОЛЕЕ РАЦИОНАЛЬНО, ЕСЛИ ВВЕСТИ СЧЕТЧИК И ОПЕРАТОР GO TO

```

10 LET TOTAL=0
20 LET COUNT=1
30 INPUT A
40 REM COUNT= NUMBER OF TIME THAT A HAS
   BEEN INPUT SO FAR
50 LET TOTAL=TOTAL+A
60 LET COUNT=COUNT+1
70 IF COUNT<=5 THEN GO TO 30
80 PRINT TOTAL

```

ТЕПЕРЬ, ИЗМЕНИВ УСЛОВИЕ В СТРОКЕ 70, МОЖНО ВВЕСТИ НЕ ТОЛЬКО 5, НО И ЛЮБОЕ КОЛИЧЕСТВО ЧИСЕЛ. ДЛЯ ОРГАНИЗАЦИИ В ПРОГРАММЕ ТАКИХ СЧЕТЧИКОВ СУЩЕСТВУЮТ СПЕЦИАЛЬНЫЕ ОПЕРАТОРЫ FOR И NEXT, КОТОРЫЕ ВСЕГДА ИСПОЛЗУЮТСЯ ВМЕСТЕ.

НАША ПРОГРАММА ПРИ ИСПОЛЗОВАНИИ ЭТИХ ОПЕРАТОРОВ БУДЕТ ВЫГЛЯДЕТЬ ТАК:

```

10 LET TOTAL=0
20 FOR C=1 TO 5
30 INPUT A
40 REM C= NUMBER OF TIMES THAT A HAS
   BEEN INPUT SO FAR
50 LET TOTAL=TOTAL+A
60 NEXT C
70 PRINT TOTAL

```

ЗДЕСЬ 'С' УПРАВЛЯЮЩАЯ ПЕРЕМЕННАЯ ЦИКЛА ДОЛЖНА ИМЕТЬ ИМЯ В ОДНУ БУКВУ. 'С' ПОСЛЕДОВАТЕЛЬНО ПРИНИМАЕТ ЗНАЧЕНИЯ 1,2,3,4,5 (ПРЕДЕЛ - КОНЕЧНОЕ ЗНАЧЕНИЕ ПЕРЕМЕННОЙ ЦИКЛА) И ПРИ КАЖДОМ ПРОХОДЕ ВЫПОЛНЯЮТСЯ СТРОКИ 30, 40, 50. ЗАТЕМ, ПОСЛЕ ТОГО КАК 'С' ПРИМЕТ ПЯТОЕ ЗНАЧЕНИЕ, ВЫПОЛНИТСЯ 70-Я СТРОКА.

ПРИРАЩЕНИЕ ЗНАЧЕНИЯ УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ СОСТАВЛЯЕТ 1, НО ЭТО ЗНАЧЕНИЕ МОЖНО ИЗМЕНИТЬ, ИСПОЛЬЗУЯ УКАЗАНИЕ STEP КАК ЧАСТЬ ОПЕРАТОРА FOR. ТАКИМ ОБРАЗОМ ОБЩАЯ ФОРМА ОПЕРАТОРА FOR ВЫГЛЯДИТ СЛЕДУЮЩИМ ОБРАЗОМ:

FOR 'УПР. ПЕРЕМ.' = 'НАЧ.ЗНАЧ.' TO 'ПРЕДЕЛ' 'STEP' 'ШАГ ПРИРАЩ.' ЗДЕСЬ 'НАЧАЛЬНОЕ ЗНАЧЕНИЕ', 'ПРЕДЕЛ', 'ШАГ ПРИРАЩЕНИЯ' - ЕСТЬ ВЫРАЖЕНИЯ, ПРИНИМАЮЩИЕ ЧИСЛОВОЕ ЗНАЧЕНИЕ. ИТАК, ЕСЛИ ВЫ ЗАМЕНИТЕ СТРОКУ 20 ПРОГРАММЫ НА

```
20 C=1 TO 5 STEP 3/2
```

ТО 'С' ПОСЛЕДОВАТЕЛЬНО ПРИНИМАЕТ ЗНАЧЕНИЯ 1,2,5 И 4.

ВЫПОЛНИТЕ ПРОГРАММУ, ВЫВОДЯЩУЮ ЧИСЛА ОТ 1 ДО 10 В УБЫВАЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ:

```

10 FOR N=10 TO 1 STEP -1
20 PRINT N
30 NEXT N

```

СЛЕДУЮЩАЯ ПРОГРАММА ВЫВОДИТ ЧИСЛА ДОМИНО:

```

10 FOR M=0 TO 6
20 FOR N=0 TO M
30 PRINT M " " N " "
40 NEXT N
50 PRINT
60 NEXT M

```

ЗНАЧЕНИЕ STEP, РАВНОЕ 0, ВЫЗОВЕТ БЕСКОНЕЧНОЕ ПОВТОРЕНИЕ ЦИКЛА, ЭТОГО НЕ РЕКОМЕНДУЕТСЯ ДЕЛАТЬ.

КРАТКОЕ СОДЕРЖАНИЕ: GO SUB, RETURN

Иногда бывает удобно некоторые фрагменты программы представлять в виде отдельных частей, но несколько раз используемых в различных местах программы. Такие части оформляются как подпрограммы, которые могут вызываться в любом месте программы.

Для этого используются операторы GO SUB (GO TO SUBROUTINE) и RETURN в форме: GO SUB N, где 'N' номер первой строки в подпрограмме. Этот оператор подобен GO TO с той разницей, что при использовании GO SUB компьютер запоминает следующий после GO SUB оператор, которому и передается управление после выполнения программы.

Делается это посредством помещения номера оператора (адреса возврата) в специальную область памяти, называемую GO SUB стек. RETURN выбирает верхний адрес возврата из GO SUB стека и продолжает выполнение программы с оператора, следующего после оператора с этим номером. Приведем пример:

```
100 LET X=10
110 GO SUB 500
120 PRINT S
130 LET X=X+4
140 GO SUB 500
150 PRINT S
160 LET X=X+2
170 GO SUB 500
180 PRINT S
190 STOP
500 LET S=0
510 FOR U=1 TO X
520 LET S=S+U
530 NEXT U
540 RETURN
```

В общем случае подпрограмма может вызывать другие подпрограммы и даже саму себя (такая подпрограмма называется рекурсивной).

ГЛАВА 6

ОПЕРАТОРЫ READ, DATA, RESTORE =====

В некоторых предыдущих программах мы видели, что информация или данные могут быть введены в компьютер при помощи оператора INPUT. Иногда это может быть очень утомительно, особенно если многие данные повторяются каждый раз при выполнении программы. Вы можете сэкономить много времени используя команды READ, DATA и RESTORE. Например:

```
10 READ A, B, C
20 PRINT A, B, C
30 DATA 10, 20, 30
40 STOP
```

Оператор READ состоит из слова READ, за которым следует список имен переменных, разделенных запятыми. Он выполняется значительно эффективнее оператора INPUT, особенно когда вводимые значения присваиваются переменным. В этом случае компьютер ищет величины переменных в утверждении DATA.

Каждый оператор DATA — это список значения — числовых или строчных — разделенных запятыми. Вы можете вводить их где

УГОДНО В ПРОГРАММЕ, ТАК КАК КОМПЬЮТЕР ИГНОРИРУЕТ ИХ, ЗА ИСКЛЮЧЕНИЕМ ТЕХ СЛУЧАЕВ, КОГДА ИХ ИСПОЛЬЗУЕТ ОПЕРАТОР READ. ВЫ ДОЛЖНЫ ПРЕДСТАВЛЯТЬ ВЫРАЖЕНИЯ ДЛЯ ВСЕХ ОПЕРАТОРОВ DATA В ПРОГРАММЕ, ЧТО ПОЗВОЛИТ ВАМ ВВОДИТЬ СОВМЕСТНО ФОРМЫ ОДНОЙ ДЛИННЫ СПИСКА ВЫРАЖЕНИЙ (СПИСКА DATA). СНАЧАЛА КОМПЬЮТЕР ВЫБИРАЕТ ПЕРВОЕ ЗНАЧЕНИЕ ИЗ СПИСКА DATA ДЛЯ ВЕЛИЧИНЫ ИЗ ОПЕРАТОРОВ READ, В СЛЕДУЮЩИЙ РАЗ ОН БЕРЕТ ВТОРОЕ ЗНАЧЕНИЕ ИЗ СПИСКА DATA И, ТАКИМ ОБРАЗОМ, ВЫБИРАЕМЫЕ ПОСЛЕДОВАТЕЛЬНО ОПЕРАТОРЫ READ ОБРАБАТЫВАЮТСЯ С ИСПОЛЬЗОВАНИЕМ СПИСКА DATA (ЕСЛИ ОКАЗАЛОСЬ, ЧТО СПИСОК ПРОСМОТРЕН ДО КОНЦА, ТО ВОЗНИКАЕТ ОШИБКА).

ЗАМЕЬТЕ ДЛЯ СЕБЯ, ЧТО ЯВЛЯЕТСЯ БЕСПОЛЕЗНЫМ ВВЕДЕНИЕ ОПЕРАТОРА DATA В ВИДЕ УПРАВЛЯЮЩЕГО ОПЕРАТОРА, ТАК КАК ОПЕРАТОР READ НЕ ОБНАРУЖИТ ЕГО. ОПЕРАТОР DATA ВЫПОЛНЯЕТСЯ ТОЛЬКО В ПРОГРАММЕ.

МОЖНО ПОСМОТРЕТЬ, КАК ЭТИ ТРЕБОВАНИЯ ВЫПОЛНЯЮТСЯ В ВЕСЬМА ХАРАКТЕРНОЙ ПРОГРАММЕ; СТРОКА 10 ДАЕТ УКАЗАНИЕ КОМПЬЮТЕРУ ЧИТАТЬ ТРИ ЗНАЧЕНИЯ ДАННЫХ И ПРИСВОИТЬ ИХ ПЕРЕМЕННЫМ A, B, C. СТРОКА 20 ГОВОРИТ О ТОМ, ЧТО НАДО ВЫВЕСТИ (PRINT) ТРИ ЗНАЧЕНИЯ. В СТРОКЕ 30 ОПЕРАТОР DATA ЗАДАЕТ ЗНАЧЕНИЯ ДЛЯ A, B, C. СТРОКА 40 ЗАКАНЧИВАЕТ ПРОГРАММУ.

ИНФОРМАЦИЯ ОБ ОПЕРАТОРЕ DATA МОЖЕТ БЫТЬ ЧАСТЬЮ FOR..NEXT ЦИКЛА. НАПРИМЕР;

```
10 FOR N=1 TO 6
20 READ D
30 DATA 2,4,6,8,10,12
40 PRINT D
50 NEXT N
60 STOP
```

КОГДА ВЫПОЛНЯЕТСЯ ЭТА ПРОГРАММА, ТО МОЖНО УВИДЕТЬ, КАК READ ОПЕРАТОР ПЕРЕБИРАЕТ ВЕСЬ СПИСОК DATA.

ОПЕРАТОР DATA МОЖЕТ СОДЕРЖАТЬ ТАКЖЕ И СТРОЧНЫЕ ПЕРЕМЕННЫЕ. НАПРИМЕР:

```
10 READ D$
20 PRINT " THE DATE IS", D$
30 DATA " JUNE 1ST, 1982"
40 STOP
```

ЭТО ПРОСТОЙ СПОСОБ ПОЛУЧЕНИЯ ВЫРАЖЕНИЯ ИЗ DATA СПИСКА СТАРТ И ВЫПОЛНЕНИЕ ОТ НАЧАЛА ДО ТЕХ ПОР, ПОКА НЕ БУДЕТ ДОСТИГНУТ КОНЕЦ, ОДНАКО ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ И ПРОГРАММНЫЙ ПЕРЕХОД ДЛЯ DATA СПИСКОВ, ИСПОЛЬЗУЯ ОПЕРАТОР RESTORE. В ЭТОМ СЛУЧАЕ ИСПОЛЬЗУЕТСЯ ОПЕРАТОР RESTORE С УКАЗАНИЕМ ПОСЛЕ НЕГО НОМЕРА СТРОКИ С ОПЕРАТОРОМ DATA, И ВСЕ ПОСЛЕДОВАТЕЛЬНО ВСТРЕЧАЮЩИЕСЯ В ПРОГРАММЕ ОПЕРАТОРЫ READ ВВОДЯТ ДАННЫЕ ПОДРЯД, НАЧИНАЯ С ПЕРВОГО ОПЕРАТОРА DATA. ВООБЩЕ-ТО, ВЫ МОЖЕТЕ НЕ УКАЗЫВАТЬ НОМЕР СТРОКИ В ОПЕРАТОРЕ RESTORE, И, В ЭТОМ СЛУЧАЕ, УКАЗАТЕЛЬ ДАННЫХ СТАНОВИТСЯ НА ПЕРВЫЙ ОПЕРАТОР В ПРОГРАММЕ. ПОПРОБУЙТЕ ВЫПОЛНИТЬ ТАКУЮ ПРОГРАММУ;

```
10 READ A,B
20 PRINT A,B
30 RESTORE 10
40 READ X,Y,Z
50 PRINT X,Y,Z
60 DATA 1,2,3
70 STOP
```

В ЭТОЙ ПРОГРАММЕ ПЕРЕМЕННЫМ, ВВОДИМЫМ В СТРОКЕ 10, БУДУТ ПРИСВОЕНЫ ЗНАЧЕНИЯ A=1 И B=2. ОПЕРАТОР RESTORE 10 СБРАСЫВАЕТ УКАЗАТЕЛЬ ДАННЫХ В НАЧАЛЬНОЕ ПОЛОЖЕНИЕ И СТРОКА 40 ПРИСВОИТ ЗНАЧЕНИЯ ПЕРЕМЕННЫМ X,Y И Z НАЧИНАЯ С ПЕРВОГО ЗНАЧЕНИЯ В DATA.

ВЫПОЛНИТЕ ПРОГРАММУ БЕЗ СТРОКИ 30 И ВЫ УВИДИТЕ САМИ, ЧТО ИЗ ЭТОГО ПОЛУЧИТСЯ.

КРАТКОЕ СОДЕРЖАНИЕ: ОПЕРАЦИИ $+$, $-$, $*$, $/$

ВЫРАЖЕНИЯ, УСЛ. ОБОЗНАЧЕНИЯ, ИМЕНА ПЕРЕМЕННЫХ

ВЫ ВИДЕЛИ НЕСКОЛЬКО ПРИМЕРОВ В КОТОРЫХ ZX SPECTRUM МОЖЕТ

ОПЕРИРОВАТЬ ЧИСЛАМИ. ТАКЖЕ МОЖНО ВЫПОЛНЯТЬ ЧЕТЫРЕ АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ: $+$, $-$, $*$, $/$. ПРИМЕР:

LET TAX= SUM*15/100

ОТСЮДА ВИДНО, ЧТО ВЫЧИСЛЕНИЯ МОГУТ БЫТЬ КОМБИНИРОВАННЫМИ. КОМБИНАЦИИ ТАКОГО ТИПА, КАК SUM * 15/100 НАЗЫВАЮТСЯ ВЫРАЖЕНИЯМИ. ВЫРАЖЕНИЕ - ЭТО САМЫЙ КОРОТКИЙ ПУТЬ ДЛЯ УКАЗАНИЯ КОМПЬЮТЕРУ НА ТО, ЧТО ВЫЧИСЛЕНИЯ НАДО ДЕЛАТЬ ОДНО ЗА ДРУГИМ. В НАШЕМ ПРИМЕРЕ ВЫРАЖЕНИЕ

SUM*15/100

УКАЗЫВАЕТ: ВОЗЬМИ ЗНАЧЕНИЕ ПЕРЕМЕННОЙ С ИМЕНЕМ "SUM", УМНОЖЬ ЕГО НА 15 И ЗАТЕМ РАЗДЕЛИ НА 100. ЕСЛИ ВЫ НЕ МОЖЕТЕ ЕЩЕ ЭТОГО СДЕЛАТЬ, МЫ РЕКОМЕНДУЕМ ПРОСМОТРЕТЬ ВВОДНУЮ ЧАСТЬ ЭТОЙ КНИГИ, ЧТОБЫ ОЗНАКОМИТЬСЯ С ТЕМ, КАК ZX SPECTRUM РАБОТАЕТ С ЧИСЛАМИ И КАКОВ ПОРЯДОК, В КОТОРОМ ВЫПОЛНЯЮТСЯ МАТЕМАТИЧЕСКИЕ ВЫРАЖЕНИЯ.

КРАТКОЕ ПОВТОРЕНИЕ:

УМНОЖЕНИЕ И ДЕЛЕНИЕ ВЫПОЛНЯЮТСЯ ПЕРВЫМИ, ОНИ ИМЕЮТ БОЛЕЕ ВЫСОКИЙ ПРИОРИТЕТ, ЧЕМ СЛОЖЕНИЕ И ВЫЧИТАНИЕ.

ОТНОСИТЕЛЬНО ДРУГ ДРУГА УМНОЖЕНИЕ И ДЕЛЕНИЕ ИМЕЮТ РАВНЫЕ ПРИОРИТЕТЫ. СУЩЕСТВУЕТ ПРАВИЛО, ПО КОТОРОМУ УМНОЖЕНИЕ И ДЕЛЕНИЕ ВЫПОЛНЯЮТСЯ ПОСЛЕДОВАТЕЛЬНО СЛЕВА НА ПРАВО. КОГДА ВСЕ ОНИ ВЫПОЛНЯЮТСЯ, БУДУТ ВЫПОЛНЯТЬСЯ СЛОЖЕНИЕ И ВЫЧИТАНИЕ ПО ПОРЯДКУ ТАКЖЕ СЛЕВА НАПРАВО.

ДЛЯ ЗАДАНИЯ ПРИОРИТЕТА В КОМПЬЮТЕРЕ ZX SPECTRUM ИСПОЛЗУЮТСЯ ЧИСЛА ОТ 1 ДО 16. НАПРИМЕР, ОПЕРАЦИЯ "*" И "/" ИМЕЕТ ПРИОРИТЕТ 8, А "+" И "-" - 6. ЭТОТ ПОРЯДОК ВЫЧИСЛЕНИЯ ЯВЛЯЕТСЯ ЖЕСТКИМ, НО ЕГО МОЖНО ИЗМЕНИТЬ ПРИ ПОМОЩИ СКОБОК. ВЫРАЖЕНИЕ В СКОБКАХ ВЫЧИСЛЯЕТСЯ ПЕРВЫМ, А ЗАТЕМ ПОДСТАВЛЯЕТСЯ В ОБЩЕЕ ВЫРАЖЕНИЕ КАК ОДНО ЧИСЛО.

ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ОПЕРАЦИЮ СЛОЖЕНИЕ "+" ДЛЯ СЦЕПЛЕНИЯ СТРОК (КОНКАТЕНЦИИ) В ВЫРАЖЕНИЯХ.

ИМЯ СТРОКОВОЙ ПЕРЕМЕННОЙ СОСТОИТ ИЗ БУКВЫ С ПОСЛЕДУЮЩИМ ЗНАКОМ "X". ИМЯ УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ В FOR-NEXT ЦИКЛЕ ДОЛЖНО СОСТОЯТЬ ИЗ ОДНОЙ БУКВЫ, А ИМЕНА ОБЫЧНЫХ ЧИСЛОВЫХ ПЕРЕМЕННЫХ МОГУТ ВЫБИРАТЬСЯ ПРОИЗВОЛЬНО. ОНИ МОГУТ СОДЕРЖАТЬ НЕСКОЛЬКО БУКВ И ЦИФР, НО ПЕРВОЙ ВСЕГДА ДОЛЖНА БЫТЬ БУКВА.

ВЫ МОЖЕТЕ ВСТАВЛЯТЬ В ИМЕНА ПРОБЕЛЫ ДЛЯ УДОБСТВА ЧТЕНИЯ, ПОСКОЛЬКУ КОМПЬЮТЕР НЕ СЧИТАЕТ ИХ ЧАСТЬЮ ИМЕНИ. ЗАПИСЬ ИМЕНИ ПРОПИСНЫМИ ИЛИ ЗАГЛАВНЫМИ БУКВАМИ НЕ ДЕЛАЕТ ИХ РАЗЛИЧНЫМИ.

ПРИМЕРЫ ДОПУСТИМЫХ ИМЕН ПЕРЕМЕННЫХ:

X

T42

THIS NAME IS SO LONG THAT I SHALL NEVER BE
ABLE TO TYPE
IN OUT AGAIN WITHOUT
MAKING A MISTAKE

NOW WE ARE SIX ЭТИ ДВА ИМЕНА УКАЗЫВАЮТ
NOWWEARE SIX НА ОДНУ И ТУ ЖЕ ПЕРЕМЕННУЮ

ПРИМЕРЫ НЕДОПУСТИМЫХ ИМЕН ПЕРЕМЕННЫХ:

2001

(НАЧИНАЕТСЯ С ЦИФРЫ)

3 BEARS

--//--

M*A*C*X

ЗНАК "*" НЕ БУКВА И НЕ ЦИФРА

FOTHERINGTON-TOMAS (СОДЕРЖИТ ЗНАК "-")

ЧИСЛА В ВЫРАЖЕНИЯХ МОГУТ ЗАДАВАТЬСЯ В ЭКСПОНЕНЦИАЛЬНОЙ

ФОРМЕ, ПОПРОБУЙТЕ ВЫПОЛНИТЬ:

PRINT 2,34E0
PRINT 2,34E1
PRINT 2,34E2 И Т.Д ДО
PRINT 2,34E18

ПОМНИТЕ ЧТО ОПЕРАТОР PRINT ДАЕТ ЛИШЬ 8 ЗНАЧАЩИХ ЦИФР ЧИСЛА, ПОПРОБУЙТЕ ВЫПОЛНИТЬ ЕЩЕ:

PRINT 4294967296, 4294967295-429E7

И ВЫ УВИДИТЕ, ЧТО КОМПЬЮТЕР МОЖЕТ ВОСПРИНЯТЬ ТОЛЬКО ЧИСЛО 4294967296.

КОМПЬЮТЕР ZX SPECTRUM ИСПОЛЬЗУЕТ АРИФМЕТИКУ С ПЛАВАЮЩЕЙ ТОЧКОЙ (ЗАПЯТОЙ), ПРИ ЭТОМ РАЗЛИЧНЫЕ ЧАСТИ ЧИСЛА (МАНТИСА И ПОРЯДОК) ХРАНЯТСЯ В ОТДЕЛЬНЫХ БАЙТАХ, ЧТО ПРИВОДИТ К НЕ ВСЕГДА ТОЧНЫМ РЕЗУЛЬТАТАМ ДАЖЕ ДЛЯ ЦЕЛЫХ ЧИСЕЛ. ВЫПОЛНИТЕ:

PRINT 1E10+1-1E10, 1E10-1E10+1

1E10 И 1E10+1 НЕ РАЗЛИЧАЮТСЯ КОМПЬЮТЕРОМ КАК РАЗНЫЕ ЧИСЛА (1E10 УСЕКАЕТСЯ СПРАВА)

ЕЩЕ ОДИН БОЛЕЕ НАГЛЯДНЫЙ ПРИМЕР:

PRINT 5E9+1-5E9

ПОГРЕШНОСТЬ В 5E9 СОСТАВЛЯЕТ ОКОЛО 1, А С ПРИБАВЛЕНИЕМ 1 ФАКТИЧЕСКИ ОКРУГЛИТСЯ ДО 2.

ЧИСЛА 5E9+1 И 5E9+2 ДЛЯ КОМПЬЮТЕРА РАВНЫ, НАИБОЛЬШЕЕ ЦЕЛОЕ КОТОРОЕ МОЖЕТ ВОСПРИНЯТЬ КОМПЬЮТЕР, РАВНО 2**32-1 (4 294 957 295).

СТРОКА"" БЕЗ ЕДИНОГО СИМВОЛА НАЗЫВАЕТСЯ ПУСТОЙ ИЛИ НУЛЬ СТРОКОЙ, НЕ ПУТАЙТЕ ЕЕ С ПРОБЕЛОМ, НАБЕРИТЕ:

PRINT"" HAVE YOU FINISHED""FINNEGANS WAKE"" YET?""

КОГДА ВЫ НАЖМЕТЕ КЛАВИШУ ENTER, ВЫ ПОЛУЧИТЕ МЕРЦАЮЩИЙ ЗНАК ВОПРОСА, УКАЗЫВАЮЩИЙ ОШИБОЧНОЕ МЕСТО В СТРОКЕ. КОГДА ПРИ ИНТЕРПРЕТАЦИИ ВСЕЙ СТРОКИ КОМПЬЮТЕР НАЙДЕТ ДВОЙНУЮ КАВЫЧКУ, ОТКРЫВАЮЩУЮ ""FINNEGANS WAKE"", ТО СОЧТЕТ ЕЕ ЗАКРЫВАЮЩЕЙ КАВЫЧКОЙ ДЛЯ СТРОКИ ""HAVE YOU FINISHED"" И ЗАТЕМ НЕ СМОЖЕТ ВЫВЕСТИ ""FINNEGANS WAKE"" ЗДЕСЬ НАДО ПОМНИТЬ СПЕЦИАЛЬНОЕ ПРАВИЛО: ЕСЛИ ВЫ ХОТИТЕ ВЫВЕСТИ КАВЫЧКИ ВНУТРИ СТРОКИ, ОНИ ДОЛЖНЫ УДВАИВАТЬСЯ. НАПРИМЕР: PRINT"" HAVE YOU FINISHED"" "" FINNEGANS WAKE"" YET?"" В ДАННОМ СЛУЧАЕ БУДЕТ ВЫВЕДЕНА ТОЛЬКО ФРАЗА, ОБРАМЛЕННАЯ ДВОЙНЫМИ КАВЫЧКАМИ ""FINNEGANS WAKE"".

ГЛАВА 8

СТРОКИ СИМВОЛОВ

=====

КРАТКОЕ СОДЕРЖАНИЕ: СЕЧЕНИЯ, ИСПОЛЬЗОВАНИЕ ТО

ПРИМЕЧАНИЕ: ЭТИ ОПЕРАЦИИ ОТСУТСТВУЮТ В СТАНДАРТНОМ БЕЙСИКЕ.

ПУСТЬ ИМЕЕТСЯ СТРОКА СИМВОЛОВ, ТОГДА ЕЕ ПОДСТРОКОЙ БУДЕТ НЕКОТОРАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ ИЗ ЭТОЙ СТРОКИ, ТАК ""STRING"" ЯВЛЯЕТСЯ ПОДСТРОКОЙ ОТ ""BIGGEN STRING"", А ""V STRING"" И ""BIG STRING"" НЕ ЯВЛЯЕТСЯ. СУЩЕСТВУЕТ ДЕЙСТВИЕ, НАЗЫВАЕМОЕ СЕЧЕНИЕМ ДЛЯ ОПРЕДЕЛЕНИЯ ПОДСТРОК И КОТОРОЕ МОЖЕТ ПРИМЕНЯТЬСЯ К СТРОКОВЫМ ВЫРАЖЕНИЯМ, ОБЩАЯ ЕГО ФОРМА:

"СТРОКОВОЕ ВЫРАЖЕНИЕ" ("НАЧАЛО" ТО "КОНЕЦ")

СЛЕДУЮЩЕЕ ВЫРАЖЕНИЕ ИСТИННО:

""ABCDEF"" (2 TO 5) = ""BCDEF"".

ЕСЛИ ОПУЩЕНО " НАЧАЛО", ТО ПО УМОЛЧАНИЮ ПОДРАЗУМЕВАЕТСЯ 1,

ЕСЛИ "КОНЕЦ", ТО ТО ПОДРАЗУМЕВАЕТСЯ ДЛИНА ВСЕЙ СТРОКИ, ТАК:

""ABCDEF"" (TO 5) = ""ABCDEF"" (1 TO 5) = ""ABCDEF""

""ABCDEF"" (2 TO) = "" ABCDEF"" (2 TO 6) = ""BCDEF""

"ABCDEF" (TO) = "ABCDEF" (TO 6) = "ABCDEF"

ПОСЛЕДНЕЕ ВЫРАЖЕНИЕ МОЖНО БЫЛО БЫ ЗАПИСАТЬ И ТАК.

"ABCDEF" ()

ЧТО ТОЖЕ ВЕРНО, МОЖНО ОПУСКАТЬ И СЛОВО TO:

"ABCDEF" (3) = "ABCDEF" (3 TO 3) = "C"

"НАЧАЛО" И "КОНЕЦ" ДОЛЖНЫ НАХОДИТЬСЯ В ПРЕДЕЛАХ СТРОКИ, ИНАЧЕ БУДЕТ ВЫДАНО СООБЩЕНИЕ ОБ ОШИБКЕ. ТАК, ВЫРАЖЕНИЕ

"ABCDEF" (5 TO 7)

ВЫЗЫВАЕТ СООБЩЕНИЕ "SUBSCRIPT WRONG", ТАК КАК "КОНЕЦ" ПРЕВЫШАЕТ ДЛИНУ СТРОКИ (6).

ЕСЛИ "НАЧАЛО" БОЛЬШЕ ЧЕМ "КОНЕЦ", ЛИБО ОБЕ СТРАНИЦЫ ЛЕЖАТ ЗА ПРЕДЕЛАМИ СТРОКИ, ТО РЕЗУЛЬТАТОМ БУДЕТ ПУСТАЯ СТРОКА:

"ABCDEF" (8 TO 7) = ""

"ABCDEF" (1 TO 0) = ""

"НАЧАЛО" И "КОНЕЦ" НЕ МОГУТ БЫТЬ ОТРИЦАТЕЛЬНЫМИ, ИНАЧЕ ВЫДАЕТСЯ СООБЩЕНИЕ "INTEGER OUT OF RANGE".

СЛЕДУЮЩАЯ ПРОГРАММА ИЛЛЮСТРИРУЕТ ЭТИ ПРАВИЛА:

```
10 LET A$ = "ABCDEF"
20 FOR N=1 TO 6
30 PRINT A$ ( N TO 6 )
40 NEXT N
50 STOP
```

МОЖНО ТАКЖЕ ПРИСВАИВАТЬ ЗНАЧЕНИЕ ПОДСТРОКЕ, ПОПРОБУЙТЕ:

```
10 LET A$ = " I AM THE ZX SPECTRUM "
20 PRINT A$
30 LET A$ ( 5 TO 8 ) = " ***** "
40 PRINT A$
```

ПОДСТРОКА A\$ (5 TO 8) ИМЕЕТ ДЛИНУ ТОЛЬКО В 4 СИМВОЛА, ПО ЭТОМУ БУДУТ ИСПОЛЬЗОВАНЫ ТОЛЬКО ПЕРВЫЕ ЧЕТЫРЕ ЗВЕЗДОЧКИ. ЭТО ОСОБЕННОСТЬ ПРИСВОЕНИЯ ЗНАЧЕНИЯ ПОДСТРОКЕ: ДЛИННЫЕ ДАННЫЕ УСЕКАЮТСЯ СПРАВА, А КОРОТКИЕ ДОПОЛНЯЮТСЯ ПРОБЕЛАМИ ДО ДЛИНЫ ПОДСТРОКИ. ДЕЙСТВИЕ НАЗЫВАЕТСЯ ПРОКРУСТИАНОМ В ЧЕСТЬ МИФИЧЕСКОГО РАЗБОЙНИКА ПРОКРУСТА, КОТОРЫЙ СВОИМ ГОСТЯМ ЛИБО ОТРУБАЛ НОГИ, ЛИБО ВЫТАГИВАЛ ИХ ЕСЛИ ОНИ НЕ ПОДХОДИЛИ ПО ДЛИНЕ К ЕГО КРОВАТИ.

ЕСЛИ ВЫ ТЕПЕРЬ ВЫПОЛНИТЕ:

LET A\$ () = "HELLO THERE" И PRINT A\$

ВЫ УВИДИТЕ ЧТО БУДУТ ВЫВЕДЕНЫ ДОПОЛНИТЕЛЬНЫЕ ПРОБЕЛЫ, ТАК "A\$ ()" СЧИТАЕТСЯ ПОДСТРОКОЙ. ДЛЯ ПРАВИЛЬНОГО ЗАПОЛНЕНИЯ СЛЕДУЕТ ПИСАТЬ:

LET A\$ = "HELLO THERE"

МОЖНО ИСПОЛЬЗОВАТЬ СКОБКИ, ЧТО ПОЗВОЛЯЕТ ВЫЧИСЛЯТЬ ЗНАЧЕНИЕ СТРОКОВОГО ВЫРАЖЕНИЯ ПЕРЕД ТЕМ, КАК БРАТЬ СЕЧЕНИЕ, НАПРИМЕР:

```
"ABC" + "DEF" ( 1 TO 2 ) = "ABCDE"
( "ABC" + "DEF" ) ( 1 TO 2 ) = "AB"
```

ГЛАВА 9

ФУНКЦИИ

=====

КРАТКОЕ СОДЕРЖАНИЕ: DEF, LEN, STR\$, VAL, SGN, ABS, INT, SQR, FN
ФУНКЦИИ — ЭТО ЗАШИТЫЕ В БЕЙСИК — СИСТЕМУ ПОДПРОГРАММЫ, КОТОРЫЕ ПОЛУЧАЯ НА ВХОД ОДНИ ЗНАЧЕНИЯ, НАЗЫВАЕМЫЕ АРГУМЕНТАМИ, ВОЗВРАЩАЮТ ДРУГИЕ ЗНАЧЕНИЯ-РЕЗУЛЬТАТЫ.

ФУНКЦИИ ИСПОЛЬЗУЮТСЯ В ВЫРАЖЕНИИ ПРОСТЫМ ВКЛЮЧЕНИЕМ В НЕГО ИМЕНИ ФУНКЦИИ С ПОСЛЕДУЮЩИМИ АРГУМЕНТАМИ.

ПРИ ВЫЧИСЛЕНИИ ВЫРАЖЕНИЯ, ВЫЧИСЛЯЕТСЯ И ЗНАЧЕНИЕ ФУНКЦИИ
НАПРИМЕР: ФУНКЦИЯ LEN ВОЗВРАЩАЕТ ДЛИНУ ЗАДАННОГО В НЕИ СТРОКОВОГО АРГУМЕНТА, ВЫ МОЖЕТЕ ЗАПИСАТЬ:

```
PRINT LEN "SINKLAIR"
```


А КОМПЬЮТЕР ВЫВЕДЕТ ОТВЕТ " 8 ", Т.Е. КОЛИЧЕСТВО БУКВ В СЛОВЕ "SINKLAIR" (ДЛЯ ВВОДА С КЛАВИАТУРЫ ИМЕНИ ФУНКЦИИ LEN, ВЫ ДОЛЖНЫ ВОЙТИ В НЕОБХОДИМЫЙ РЕЖИМ, НАЖАВ КЛАВИШИ CAPS SHIFT И SUM SHIFT, КУРСОР ИЗМЕНИТСЯ С (L) НА (E), И НАЖАТЬ КЛАВИШУ R).

ЕСЛИ В ОДНОМ ВЫРАЖЕНИИ ИСПОЛЬЗУЮТСЯ И ФУНКЦИИ И ОПЕРАЦИИ ФУНКЦИИ БУДУТ ВЫЧИСЛЕНЫ ПЕРЕД ВЫПОЛНЕНИЕМ ЛЮБЫХ ОПЕРАЦИЙ, ОДНАКО ВЫ МОЖЕТЕ ИЗМЕНИТЬ ЭТОТ ПОРЯДОК, ПРИМЕНЯЯ СКОБКИ.

ФУНКЦИЯ STR* ПРЕОБРАЗУЕТ ЧИСЛО В СИМВОЛЬНЫЙ ВИД, ПОДОБНО
ФОРМАТУ ВЫВОДА ЧИСЕЛ ОПЕРАТОРОМ PRINT;

LET A $\times$ = STR $\times$ 1E2

АНАЛОГИЧНО ПО ДЕЙСТВИЮ КОМАНДЕ LET A* = " 100"

ИЛИ ВЫПОЛНИТЕ

PRINT LEN STR * 100,0000

И ПОЛУЧИТЕ ОТВЕТ 3, Т.К. $\text{СТР} \times 100,0000 = '' 100''$

ФУНКЦИЯ VAL ОБРАТИТСЯ В ФУНКЦИЮ STR* И ПРЕОБРАЗУЕТ СТРО-
КУ В ЧИСЛО, ТАК:

```
VAL "3.5" = 3.5
```

ИЛИ

VAL " 2*3" = 6

ИЛИ ДАЖЕ ТАК

```
VAL ( ' ' 2 ' ' + ' ' * 3 ' ' ) = 6
```

В ПОСЛЕДНЕМ СЛУЧАЕ ПРОИСХОДИТ ВЫЧИСЛЕНИЕ ДВУХ ВЫРАЖЕНИЙ:
СНАЧАЛА СТРОКОВОГО С ПОЛУЧЕНИЕМ СТРОКИ "2*3", ЗАТЕМ ЧИСЛОВОГО
С ПОЛУЧЕНИЕМ СТРОКИ "6".

МОЖНО ПОПАСТЬ В ЗАТРУДНИТЕЛЬНОЕ ПОЛОЖЕНИЕ, НАПРИМЕР:
PRINT VAL " VAL" " VAL " " " " "2" " " " " "

ПОМНЯ, ЧТО ВНУТРИ СТРОКИ КАВЫЧКИ УДВАИВАЮТСЯ, МЫ ВИДИМ, ЧТО В НАШЕМ СЛУЧАЕ МОЖЕТ ПОНАДОБИТЬСЯ УЧЕТВЕРЕНИЕ ИЛИ ДАЖЕ УВОСМИРЕНИЕ.

ИМЕЕТСЯ ЕЩЕ ОДНА ФУНКЦИЯ ПОДОБНАЯ VAL - ЭТО VAL *, И АРГУМЕНТОМ И РЕЗУЛЬТАТОМ ЭТОЙ ФУНКЦИИ ЯВЛЯЕТСЯ СТРОКА СИМВОЛОВ. ОНА РАБОТАЕТ КАК VAL, ПРИМЕНЕННАЯ ДВАЖДЫ, РАСКРЫВАЯ ВСЕ КАВЫЧКИ В СТРОКАХ. VAL* " " " " FRUIT PUNCH " " " " = " " FRUIT PUNCH " "

СДЕЛАЙТЕ ЛЕТ АЖ 99

И ЗАТЕМ ВЫВЕДИТЕ ВСЕ СЛЕДУЮЩИЕ ЗНАЧЕНИЯ

VAL AX

VAL " A x "

VAL " " " " AX " " " "

VALX AX

VAL * * * A * *

VALX " " " " AX " " " " "

НЕКОТОРЫЕ ИЗ НИХ СРАБОТАЮТ, А НЕКОТОРЫЕ НЕТ, ПРОАНАЛИЗИ-
РУЙТЕ ЭТИ ОТВЕТЫ.

ФУНКЦИЯ SGN - ЭТО ТАК НАЗЫВАЕМАЯ МАТЕМАТИЧЕСКАЯ ФУНКЦИЯ СИГНУМ(ЗНАК), И АРГУМЕНТ И РЕЗУЛЬТАТ ЕГО ЧИСЛОВЫЕ, РЕЗУЛЬТАТ РАВЕН:

1. ЕСЛИ АРГУМЕНТ ПОЛОЖИТЕЛЕН

1. ЕСЛИ АРГУМЕНТ ПОЛОЖИТЕЛЕН.

0. ЕСЛИ АРГУМЕНТ РАВЕН 0

1. ЕСЛИ АРГУМЕНТ ОТРИЦАТЕЛЕН.

ФУНКЦИЯ ABS ПРЕОБРАЗУЕТ АРГУМЕНТ В ПОЛОЖИТЕЛЬНОЕ ЧИСЛО:

$$\text{ABS } -3,2 = \text{ABS } 3,2 = 3,2$$

ФУНКЦИЯ INT (ОТ "INTEGER PART" - ЦЕЛАЯ ЧАСТЬ) ПРЕОБРАЗУЕТ ДРОБНОЕ ЧИСЛО К ЦЕЛОМУ ОТБРАСЫВАНИЕМ ДРОБНОЙ ЧАСТИ:

INT 3.9 = 3

СЛОЖНОСТИ ВОЗНИКАЕТ ПРИ ОТРИЦАТЕЛЬНОМ АРГУМЕНТЕ, ТАК КАК
ОКРУГЛЕНИЕ ПРОИСХОДИТ К БЛИЖАЙШЕМУ ЦЕЛОМУ НЕ БОЛЬШЕМУ, ЧЕМ АРГУ-
МЕНТ:

INT -3.9 == -4

НАПРИМЕР:

	ФУНКЦИЯ	SQR	ВЫЧИСЛЯЕТ	КОРЕНЬ	КВАДРАТНЫЙ	ОТ	ЧИСЛА,
			$\text{SQR } 4 = 2$				

$$SQ_4 = 2$$
$$SQR \ 0,25 = 0,5$$

$SQR\ 2 = 1,4142136$ (ПРИБЛИЗИТЕЛЬНО)

ЕСЛИ АРГУМЕНТ ОТРИЦАТЕЛЬНЫЙ, ТО ВЫДАЕТСЯ СООБЩЕНИЕ:

' A INVALID ARGUMENT'.

ВЫ ТАКЖЕ МОЖЕТЕ САМИ ОПРЕДЕЛИТЬ ДЛЯ СЕБЯ КАКУЮ. НИБУДЬ

ФУНКЦИЮ, УКАЗАВ FN И ИМЯ ЭТОЙ ФУНКЦИИ (БУКВУ, ЕСЛИ АРГУМЕНТ ЧИСЛОВОЙ ИЛИ БУКВУ И *, ЕСЛИ СТРОКОВЫЙ). АРГУМЕНТЫ ДОЛЖНЫ БЫТЬ ОБЯЗАТЕЛЬНО ЗАКЛЮЧЕНЫ В СКОБКИ.

ВЫ МОЖЕТЕ ОПРЕДЕЛИТЬ ФУНКЦИЮ ВВОДОМ ОПЕРАТОРА DEF В НЕКОТОРОМ МЕСТЕ ПРОГРАММЫ. НАПРИМЕР: ЗАДАДИМ ФУНКЦИЮ, ВЫЧИСЛЯЮЩУЮ КВАДРАТ ЧИСЛА 10 DEF FN S(X) = X*2: REM THE SQUARE OF X

DEF ВВОДИТСЯ В СООТВЕТСТВУЮЩЕМ РЕЖИМЕ (SYMBOL SHIFT И1), ТЕПЕРЬ ФУНКЦИЯ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ В ПРОГРАММЕ.

PRINT FN S(2)

PRINT FN S(3+4)

PRINT 1+ INT FN S(LEN " CHICREN " /2+3)

ФУНКЦИЯ INT ВСЕГДА ОКРУГЛЯЕТ ДО ЦЕЛОГО. ДЛЯ ОКРУГЛЕНИЯ С ТОЧНОСТЬЮ 0,5 НАДО ДОБАВИТЬ К РЕЗУЛЬТАТУ ',5', ВЫ МОЖЕТЕ ЗАДАТЬ ДЛЯ СЕБЯ ТАКУЮ ФУНКЦИЮ: DEF FN R(X) = INT(X+0,5): REM GIVES X ROUNDED TO THE NEAREST INTEGER

И МОЖЕТЕ ЗАТЕМ ПОПРОБОВАТЬ ВВЕСТИ :

FN R(2,9)=3

FN R(2,4)=2

FN R(-2,9)=-3

FN R(-2,4)=-2

ВВЕДИТЕ И ВЫПОЛНИТЕ СЛЕДУЮЩЕЕ:

10 LET X=0: LET Y=0: LET A=10

20 DEF FN P(X,Y)=A+X*Y

30 DEF FN R()=A+X*Y

40 PRINT FN P(2,3), FN Q()

ЕСТЬ ОДНА ТОЧНОСТЬ В ЭТОЙ ПРОГРАММЕ, ВО-ПЕРВЫХ, ФУНКЦИЯ FN () НЕ ИСПОЛЬЗУЕТ АРГУМЕНТОВ, НО СКОБКИ ПРИ ЭТОМ ДОЛЖНЫ ОБЯЗАТЕЛЬНО ИСПОЛЬЗОВАТЬСЯ. ВО-ВТОРЫХ, ОПЕРАТОРЫ DEF НЕ ВЫПОЛНИМЫЕ, КОМПЬЮТЕР ПОСЛЕ ВЫПОЛНЕНИЯ СТРОКИ 10, ПРОСТО ПЕРЕХОДИТ К ВЫПОЛНЕНИЮ СТРОКИ 40, ПОМНИТЕ, ЧТО DEF МОЖЕТ БЫТЬ ТОЛЬКО ОПЕРАТОРОМ, НО НЕ КОМАНДОЙ. В ТРЕТЬИХ, 'X' И 'Y' - ИМЕНА ЦЕЛЫХ ПЕРЕМЕННЫХ В ПРОГРАММЕ И В ТО ЖЕ ВРЕМЯ ИМЕНА АРГУМЕНТОВ В ФУНКЦИИ FN P.

ФУНКЦИЯ FN P ИСПОЛЬЗУЕТ В ВЫЧИСЛЕНИИ РЕЗУЛЬТАТА ЗНАЧЕНИЯ АРГУМЕНТОВ 'X', 'Y' И ПЕРЕМЕННОЙ 'A', НЕ ЯВЛЯЮЩЕЙСЯ АРГУМЕНТОМ. ТАК, КОГДА ВЫЧИСЛЯЕТСЯ FN P(2,3), ЗНАЧЕНИЕ 'A' РАВНО 10, КАК ОПРЕДЕЛЕНО В ПРОГРАММЕ, А ЗНАЧЕНИЯ 'X' И 'Y' СООТВЕТСТВЕННО 2 И 3, ТАК КАК ОНИ АРГУМЕНТЫ И РЕЗУЛЬТАТ БУДЕТ $10+2*3=16$.

ПРИ ВЫЧИСЛЕНИИ FN Q() УЧАСТВУЮТ ТОЛЬКО ПЕРЕМЕННЫЕ ПРОГРАММЫ, ТАК КАК АРГУМЕНТОВ НЕТ, И ОТВЕТ В ЭТОМ СЛУЧАЕ БУДЕТ $10+0*0$. ТЕПЕРЬ ИЗМЕНИМ СТРОКУ 20 НА 20 DEF FN P(X,Y)=FNQ() В ЭТОМ СЛУЧАЕ FN P(2,3) БУДЕТ ВОЗВРАЩАТЬ ЗНАЧЕНИЕ 10.

НЕКОТОРЫЕ ВЕРСИИ ЯЗЫКА БЕЙСИК ИМЕЮТ ФУНКЦИИ LEFT*, RIGHT*, TL*

LEFT*(A*,N) - ВОЗВРАЩАЕТ ПОДСТРОКУ, СОДЕРЖАЩУЮ 'N' ПЕРВЫХ СИМВОЛОВ СТРОКИ 'A*'

RIGHT*(A*,N) - ВОЗВРАЩАЕТ ПОДСТРОКУ, СОДЕРЖАЩУЮ 'N' ПОСЛЕДНИХ СИМВОЛОВ В СТРОКЕ 'A*'

TL*(A*) - ВОЗВРАЩАЕТ ПОДСТРОКУ, СОДЕРЖАЩУЮ ВСЕ СИМВОЛЫ СТРОКИ 'A*', КРОМЕ ПЕРВОГО

ВЫ МОЖЕТЕ ОПРЕДЕЛИТЬ ТАКИЕ ФУНКЦИИ НА СВОЕМ КОМПЬЮТЕРЕ:

10 DEF FN T*(A*)=A*(2 TO): REM TL*

20 DEF FN L*(A*,N)=A*(TO N): REM LEFT*

ПРОВЕРЬТЕ ИХ РАБОТУ СО СТРОКАМИ ДЛИНОЙ 0 И 1.

ПРИМЕЧАНИЕ: ФУНКЦИЯ МОЖЕТ ИМЕТЬ ДО 26 ЧИСЛОВЫХ АРГУМЕНТОВ И В ТОЖЕ ВРЕМЯ ДО 26 СТРОКОВЫХ.

ГЛАВА 10 МАТЕМАТИЧЕСКИЕ ФУНКЦИИ =====

КРАТКОЕ СОДЕРЖАНИЕ: **, PI, EXP, LN, SIN, COS, TAN, ASN, ACS, ATN
В ЭТОЙ ГЛАВЕ ОПИСЫВАЮТСЯ МАТЕМАТИЧЕСКИЕ ФУНКЦИИ, КОТОРЫЕ МОГУТ БЫТЬ ВЫПОЛНЕНЫ НА ZX SPECTRUM, ВПОЛНЕ ВОЗМОЖНО, ЧТО ВАМ НИКОГДА НЕ ПРИДЕТСЯ ВОСПОЛЬЗОВАТЬСЯ НИМИ И ЕСЛИ ВЫ СОЧТЕТЕ ИХ СЛИШКОМ СЛОЖНЫМИ, МОЖЕТЕ ПРОПУСТИТЬ ЭТУ ГЛАВУ, ВСЕ СКАЗАННОЕ ОТНОСИТСЯ К ФУНКЦИЯМ: ** (ВОЗВЕДЕНИЕ В СТЕПЕНЬ), EXP, LN, ТРИГОНОМЕТРИЧЕСКИМ ФУНКЦИЯМ: SIN, COS, TAN, И ОБРАТНЫХ К НИМ: ASN, ACS, ATN, ** И EXP.

ВЫ МОЖЕТЕ ВОЗВЕСТИ ЧИСЛО В НЕКОТОРУЮ СТЕПЕНЬ ПУТЕМ МНОЖИТЕЛЬНОГО УМНОЖЕНИЯ ЕГО САМОГО НА СЕБЯ НЕОБХОДИМОЕ ЧИСЛО РАЗ. ЭТО ОБЫЧНО ИЗОБРАЖАЕТСЯ ЗАПИСЬЮ ЧИСЛА ОБОЗНАЧАЮЩЕГО СТЕПЕНЬ СПРАВА ВВЕРХ ОТ ЧИСЛА ОБОЗНАЧАЮЩЕГО ОСНОВАНИЕ, НО ТАКУЮ ФОРМУ ЗАПИСИ ТРУДНО РЕАЛИЗОВАТЬ В КОМПЬЮТЕРЕ, ПОЭТОМУ ТАМ ИСПОЛЬЗУЕТСЯ СПЕЦИАЛЬНЫЙ СИМВОЛ (НАПРАВЛЕННАЯ ВВЕРХ СТРЕЛКА, В ДАННОМ ОПИСАНИИ ЗАМЕНЕННАЯ ДВУМЯ ЗВЕЗДОЧКАМИ: '**'), НАПРИМЕР СТЕПЕНИ ДВОЙКИ МОЖНО ПРЕДСТАВИТЬ

$$\begin{aligned} 2^{**1} &= 2 \\ 2^{**2} &= 2 * 2 = 4 \quad (\text{ДВА В КВАДРАТЕ}) \\ 2^{**3} &= 2 * 2 * 2 = 8 \quad (\text{ДВА В КУБЕ}) \end{aligned}$$

ТАКИМ ОБРАЗОМ, ЗАПИСЬ A**B ОЗНАЧАЕТ: УМНОЖЬ 'A' САМО НА СЕБЯ 'B' РАЗ, НО ЭТО ПРЕДПОЛАГАЕТ, ЧТО 'B' ПОЛОЖИТЕЛЬНОЕ ЦЕЛОЕ ЧИСЛО.

ДЛЯ НАХОЖДЕНИЯ ОПРЕДЕЛЕНИЯ ДЛЯ ЭТОГО ДЕЙСТВИЯ ПРИ ДРУГИХ ЗНАЧЕНИЯХ A И B, МЫ ЗАПИШЕМ ВЫРАЖЕНИЕ:

$$A^{** (B+C)} = A^{**B} * A^{**C}$$

ЗДЕСЬ НАДО ПОМНИТЬ, ЧТО ОПЕРАЦИЯ '**' ИМЕЕТ БОЛЕЕ ВЫСОКИЙ ПРИОРИТЕТ, ЧЕМ '*' И '/'. ВЫ МОЖЕТЕ БЫТЬ УВЕРЕНЫ В ПРАВИЛЬНОСТИ ЭТОГО ВЫРАЖЕНИЯ, ЕСЛИ 'B' И 'C' ЦЕЛЫЕ ПОЛОЖИТЕЛЬНЫЕ ЧИСЛА, НО ЕСЛИ ЭТО НЕ ТАК, А ВЫ ВСЕ-ТАКИ РЕШИЛИ ВЫПОЛНИТЬ ВОЗВЕДЕНИЕ В СТЕПЕНЬ, ТО ВЫ ДОЛЖНЫ ЗНАТЬ ЧТО:

$$\begin{aligned} A^{**0} &= 1 \\ A^{**(-B)} &= 1 / (A^{**B}) \\ A^{**(1/B)} &= \text{КОРЕНЬ B-ОЙ СТЕПЕНИ ИЗ A} \\ A^{** (B * C)} &= (A^{**B})^{**C} \\ A^{**(-1)} &= 1 / A \\ A^{**(1/2)} &= \text{SQRT A} \end{aligned}$$

ПОЛЕЗНО ПОМНИТЬ, ЧТО:

ПОЭКСПЕРИМЕНТИРУЙТЕ С ЭТИМ, ПОПРОБОВАВ ВЫПОЛНИТЬ ТАКУЮ ПРОГРАММУ:

```
10 PRINT A, B, C
20 PRINT A** (B+C), A**B * A**C
30 GO TO 10
```

КОМПЬЮТЕР СТАНЕТ ВЫВОДИТЬ ДВА ЧИСЛА, ЕСЛИ ВЫ ПРАВИЛЬНО, КОНЕЧНО НАБРАЛИ ПРОГРАММУ, ЧИСЛО A, КСТАТИ, НЕ ДОЛЖНО БЫТЬ ОТРИЦАТЕЛЬНЫМ.

ДРУГОЙ ТИПИЧНЫЙ ПРИМЕР ИСПОЛЬЗОВАНИЯ ЭТОЙ ОПЕРАЦИИ, ЭТО ВЫЧИСЛЕНИЕ ДОХОДА. ПРЕДПОЛОЖИМ, ЧТО ВЫ ВЛОЖИЛИ ЧАСТЬ СВОИХ ДЕНЕГ В ОБЩЕСТВЕННОЕ СТРОИТЕЛЬСТВО, КОТОРОЕ ПРИНОСИТ ВАМ 15% ГОДОВЫХ, ПОСЛЕ ГОДА ВЫ БУДЕТЕ ИМЕТЬ УЖЕ НЕ ТОЧНО 100% ОТ ТОГО, ЧТО ИМЕЛИ В НАЧАЛЕ, А ПЛЮС 15% ДОХОДА, ЧТО СОСТАВИТ 115%. ДЛЯ ВЫЧИСЛЕНИЯ ДРУГИМ СПОСОБОМ, ВЫ УМНОЖАЕТЕ ВАШУ СУММУ ДЕНЕГ НА 1,15 И ПОЛУЧАЕТЕ ТОТ ЖЕ РЕЗУЛЬТАТ. В КОНЦЕ СЛЕДУЮЩЕГО ГОДА ВЫ СНОВА ПОЛУЧИТЕ ПРИБЫЛЬ, ЧТО В СУММЕ СОСТАВИТ $1,15 * 1,15 = 1,15^{**2} = 1,3225$ ОТ ВАШЕЙ ПЕРВОНАЧАЛЬНОЙ СУММЫ. В ИТОГЕ ПОСЛЕ Y ЛЕТ ВЫ БУДЕТЕ ИМЕТЬ В $1,15^{**Y}$ РАЗ БОЛЬШЕ ДЕНЕГ. ВЫПОЛНИВ :

```
FOR Y=0 TO 100: PRINT Y, 10 * 1,15**Y: NEXT Y
```

ВЫ УВИДИТЕ, ЧТО НАЧИНАЯ С 10 ФУНТОВ МОЖНО ПОЛУЧАТЬ ВСЕ БОЛЬШИЙ И БОЛЬШИЙ ДОХОД С КАПИТАЛА.

ТАКОЙ ТИП ПОВЕДЕНИЯ ФУНКЦИИ, КОГДА ПОСЛЕ ФИКСИРОВАННОГО ЧИСЛА ИНТЕРВАЛОВ ВРЕМЕНИ, ЗНАЧЕНИЯ ФУНКЦИИ ПРОПОРЦИОНАЛЬНЫ КОЛИЧЕСТВУ УМНОЖЕНИЯ ЭТОГО ЧИСЛА САМОГО НА СЕБЯ, НАЗЫВАЕТСЯ ЭКСПОНЕНЦИАЛЬНЫМ ЗАКОНОМ. ПРЕДПОЛОЖИМ ВМ НАПИСАЛИ: 10 DEF FN A(N)=A**N. ЗДЕСЬ А ОПРЕДЕЛЕННО В ОПЕРАТОРЕ LET, ЕГО ЗНАЧЕНИЕ ПЕРЕДАЕТСЯ ДЛЯ ВЫЧИСЛЕНИЯ СТЕПЕНИ.

ИМЕЕТСЯ ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЕ А, КОТОРОЕ ДЕЛАЕТ ФУНКЦИЮ FN A ИЛЛЮСТРИРУЮЩЕЙ СПЕЦИАЛЬНУЮ МАТЕМАТИЧЕСКУЮ ФУНКЦИЮ, ЭТО ЗНАЧЕНИЕ НАЗЫВАЕТСЯ 'Е'.

ZX СПЕКТРУМ ИМЕЕТ СПЕЦИАЛЬНУЮ ФУНКЦИЮ, НАЗЫВАЕМУЮ EXP И ОПРЕДЕЛЯЕМУЮ КАК: $EXP\ X = E^{**} X$

К СОЖАЛЕНИЮ, 'Е' НЕ МОЖЕТ БЫТЬ ОПРЕДЕЛЕНО ТОЧНЫМ ЧИСЛОМ. ВЫ МОЖЕТЕ УВИДЕТЬ ПЯТЬ ЕГО ПЕРВЫХ ДЕСЯТИЧНЫХ ЗНАКОВ, ВЫПОЛНИВ $PRINT\ EXP\ 1$

ТАК КАК $EXP\ 1 = E^{**} 1 = E$, КОНЕЧНО, ЭТО ЛИШЬ ПЕРВОЕ ПРИБЛИЖЕНИЕ, ВЫ НИКОГДА НЕ СМОЖЕТЕ ЗАПИСАТЬ 'Е' АБСОЛЮТНО ТОЧНО.

LN — ОБРАТНОЙ К ЭКСПОНЕНЦИАЛЬНОЙ ЯВЛЯЕТСЯ ЛОГАРИФИЧЕСКАЯ ФУНКЦИЯ. ЛОГАРИФМ (ПО ОСНОВАНИИ А) ЧИСЛА X ЕСТЬ СТЕПЕНЬ, В КОТОРУЮ НАДО ВОЗВЕСТИ А, ЧТОБЫ ПОЛУЧИТЬ X, ЭТО ЗАПИСЫВАЕТСЯ ТАК $LOGA\ X$, (ВЫРАЖЕНИЕ $A^{**} LOGA X = X$ ТАК ЖЕ ВЕРНО КАК И $LOGA(A^{**} X) = X$).

ВАМ ДОЛЖНО БЫТЬ УЖЕ ИЗВЕСТНО КАК ИСПОЛЬЗУЕТСЯ ЛОГАРИФМ ПО ОСНОВНИЮ 10 ДЛЯ УМНОЖЕНИЯ, ТАКОЙ ЛОГАРИФМ НАЗЫВАЕТСЯ ОБЩИМ. ZX СПЕКТРУМ ИМЕЕТ ФУНКЦИЮ LN, КОТОРАЯ ВЫЧИСЛЯЕТ ЛОГАРИФМ ПО ОСНОВАНИЮ 'Е', НАЗЫВАЕМЫЙ НАТУРАЛЬНЫМ, ДЛЯ ВЫЧИСЛЕНИЯ ЛОГАРИФМА С ДРУГИМ ОСНОВАНИЕМ, НАДО РАЗДЕЛИТЬ НАТУРАЛЬНЫЙ ЛОГАРИФМ ИСКОМОГО ЧИСЛА НА НАТУРАЛЬНЫЙ ЛОГАРИФМ ОСНОВАНИЯ:

$$LOGA\ X = LN\ X / LN\ A$$

ДОПУСТИМ ИМЕЕТСЯ НЕКОТОРЫЙ КРУГ, ВЫ МОЖЕТЕ НАЙТИ ЕГО ПЕРИМЕТР (ДЛИНУ ОКРУЖНОСТИ), УМНОЖИВ ЕГО ДИАМЕТР НА ЧИСЛО, НАЗЫВАЕМОЕ PI, ПОДОБНО 'Е', PI ПРЕДСТАВЛЯЕТСЯ БЕСКОНЕЧНОЙ ДЕСЯТИЧНОЙ ДРОБЬЮ, НАЧАЛО 3,141592653569...

СЛОВО 'PI' В ZX СПЕКТРУМ ОБОЗНАЧАЕТ ЭТО ЧИСЛО. ВЫПОЛНИТЕ, НАПРИМЕР: $PRINT\ PI$

SIN, COS, TAN, ASN, ACS, ATN — ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ ИЗМЕРЯЮТ ТЕ СЛУЧАИ, КОГДА ТОЧКА ПЕРЕМЕЩАЕТСЯ ВОКРУГ ОКРУЖНОСТИ ЕДИНИЧНОГО РАДИУСА. ТОЧКА СТАРТУЕТ С ПОЗИЦИИ 3-Х ЧАСОВ И ПЕРЕМЕЩАЕТСЯ ПРОТИВ ЧАСОВОЙ СТРЕЛКИ. НАЧАЛО КООРДИНАТ НАХОДИТСЯ В ЦЕНТРЕ ЭТОЙ ОКРУЖНОСТИ, ТОГДА SIN УГЛА МЕЖДУ РАДИУСОМ, СОЕДИНЯЮЩИМ ДВИЖУЩУЮСЯ ПО ОКРУЖНОСТИ ТОЧКУ С НАЧАЛОМ КООРДИНАТ, БУДЕТ ОРДИНАТА ЭТОЙ ТОЧКИ, А COS — АБСЦИСА. НЕОБХОДИМО ПОМНИТЬ, ЧТО ЕСЛИ ТОЧКА НАХОДИТСЯ СЛЕВА ОТ ОСИ ХУ, ТО КОСИНУС ОТРИЦАТЕЛЬНЫЙ, А ЕСЛИ ТОЧКА НАХОДИТСЯ ПОД ОСЬЮ Х, ТО ОТРИЦАТЕЛЬНЫЙ СИНУС. НЕОБХОДИМО ТАКЖЕ ПОМНИТЬ, ЧТО:

$$SIN(A+2*PI)=SINA$$

$$COS(A+2*PI)=COSA$$

ИМЕЮТСЯ И ДРУГИЕ ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ;

TAN — ТАНГЕНС

ASN — АРКСИНУС

ACS — АРКОСИНУС

ATN — АРКТАНГЕНС,

ПОМНИТЕ, В ZX СПЕКТРУМ ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ ВЫЧИСЛЯЮТ В РАДИАНАХ. ДЛЯ ПЕРЕВОДА ИЗ ГРАДУСОВ В РАДИАНЫ НЕОБХОДИМО ЧИСЛО РАЗДЕЛИТЬ НА 180 И УМНОЖИТЬ ЕГО НА PI, А ДЛЯ ОБРАТНОГО ПРЕОБРАЗОВАНИЯ НЕОБХОДИМО РАЗДЕЛИТЬ НА PI И УМНОЖИТЬ НА 180.

ГЛАВА 11 СЛУЧАЙНЫЕ ЧИСЛА =====

КРАТКОЕ СОДЕРЖАНИЕ: RANDOMIZE, RND.

В ЭТОЙ ГЛАВЕ ОПИСЫВАЕТСЯ ФУНКЦИЯ RND И КЛЮЧЕВОЕ СЛОВО RANDOMIZE, ИХ НЕ НАДО ПУТАТЬ, ХОТЯ ОНИ ОБЕ РАСПОЛОЖЕНЫ НА КЛАВИШЕ 'T', ДЛЯ RANDOMIZE ДОПУСТИМО СОКРАЩЕНИЕ RAND.

ПРИ ОБРАЩЕНИИ К ФУНКЦИИ RND, ОНА ВОЗВРАЩАЕТ СЛУЧАЙНОЕ ЧИСЛО В ИНТЕРВАЛЕ ОТ 0 ДО 1 (МОЖЕТ ПРИНИМАТЬ ЗНАЧЕНИЕ 0, НО НИ КОГДА 1), ПОПРОБУЙТЕ ВЫПОЛНИТЬ: 10 PRINT RND
20 GO TO 10.

ВЫ УВИДИТЕ КАК МЕНЯЕТСЯ РЕЗУЛЬТАТ.

ФАКТИЧЕСКИ RND НЕ АБСОЛЮТНО СЛУЧАЙНОЕ ЧИСЛО, А ВЫБИРАЕТСЯ В ОПРЕДЕЛЕННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ДЛИНОЙ В 65536 ЧИСЕЛ, ПОЭТОМУ ОБЫЧНО ГОВОРЯТ ЧТО RND - ПСЕВДОСЛУЧАЙНОЕ ЧИСЛО.

ДЛЯ ПОЛУЧЕНИЯ СЛУЧАЙНОГО ЧИСЛА В ИНТЕРВАЛЕ ОТЛИЧНОМ ОТ 0 МОЖНО ИСПОЛЬЗОВАТЬ ВЫРАЖЕНИЕ, НАПРИМЕР: $1,3+0,7*RND$ ДАСТ ИНТЕРВАЛ ОТ 1,3 ДО 2.

ДЛЯ ПОЛУЧЕНИЯ СЛУЧАЙНЫХ ЦЕЛЫХ ЧИСЕЛ ИСПЬЗУЙТЕ ФУНКЦИЮ
 $1+INT(RND*N)$ ГДЕ N ЛЮБОЕ ЧИСЛО
НАПРИМЕР: $1+INT(RND*6)$ ДАСТ 1,2,3,4,5,6.

ПУСТЬ ИМЕЕТСЯ ПРОГРАММА:

```
10.REM DICE THROWING PROGRAM, ( ВЫБРАСЫВАНИЕ КОСТИ)
20 CLS
30 FOR N=1 TO 2
40 PRINT 1+INT(RND*6 " ")
50 NEXT N
60 INPUT A*: GO TO 20
```

НАЖИМАЯ ENTER, ВЫ КАЖДЫЙ РАЗ БУДЕТЕ ПОЛУЧАТЬ НОМЕР, ВЫПАВШИЙ НА КОСТИ.

УТВЕРЖДЕНИЕ RANDOMIZE ИСПОЛЬЗУЕТСЯ ДЛЯ УСТАНОВЛЕНИЯ НАЧАЛА ПОСЛЕДОВАТЕЛЬНОСТИ СЛУЧАЙНЫХ ЧИСЕЛ ДЛЯ ФУНКЦИИ RND. КАК МОЖНО УВИДЕТЬ ИЗ ПРОГРАММЫ:

```
10 RANDOMIZE 1
20 FOR N=1 TO 5:PRINT RND:NEXT N
30 PRINT : GO TO 10
```

ПОСЛЕ КАЖДОГО ВЫПОЛНЕНИЯ RANDOMIZE 1 СЛУЧАЙНАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ БУДЕТ НАЧИНАТЬСЯ С ЧИСЛА 0,0022735596. В УТВЕРЖДЕНИИ RANDOMIZE ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ЛЮБЫЕ ЧИСЛА В ИНТЕРВАЛЕ ОТ 1 ДО 65535. НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ RANDOMIZE БЕЗ ЧИСЛА, А ТАКЖЕ RANDOMIZE 0, НАПРИМЕР, ИМЕЕТСЯ ПРОГРАММА:

```
10 RANDOMIZE
20 PRINT RND: GO TO 10
```

В КАЖДОЙ ИТЕРАЦИИ БУДЕТ ПЕЧАТАТЬСЯ НЕ СЛУЧАЙНОЕ ЧИСЛО. ДЛЯ УЛУЧШЕНИЯ СЛУЧАЙНОСТИ РАСПРЕДЕЛЕНИЯ МОЖНО ЗАМЕНИТЬ GO TO 10 НА GO TO 20. В ДОПОЛНЕНИЕ БОЛЬШИНСТВО ВЕРСИЙ БЕЙСИКА ИСПОЛЬЗУЕТ RND И RANDOMIZE ДЛЯ ГЕНЕРАЦИИ СЛУЧАЙНЫХ ЧИСЕЛ, НО ЭТО НЕ ЕДИНСТВЕННОЕ ИХ ПРИМЕНЕНИЕ.

НИЖЕ ПРИВОДИТСЯ ТЕКСТ ПРОГРАММЫ, МОДЕЛИРУЮЩЕЙ ВЫБРАСЫВАНИЕ МОНЕТЫ И ПОДСЧЕТА ЧИСЛА ВЫПАДЕНИЙ 'ОРЛА' И 'РЕШКИ'.
(ПЕРЕВОД ИМЕН ПРОГРАММЫ: HEADS-ОРЛЫ, TAILS-РЕШКИ, COIN-МОНЕТА)

```
10 LET HEADS=0: LET TAILS=0
20 LET COLN=INT(RND*2)
30 IF COLN=0 THEN LET HEADS=HEADS+1
40 IF COLN=1 THEN LET TAILS=TAILS+1
50 PRINT HEADS:',';TAILS
60 IF TAILS <> 0 THEN PRINT HEADS/TAILS
70 PRINT: GO TO 20
```

ЕСЛИ ПРОГРАММА ВЫПОЛНЯЕТСЯ ДОСТАТОЧНО ДОЛГО, ТО ОТНОШЕНИЕ 'ОР-
ЛОВ' К 'РЕШКАМ' ПРИБЛИЗИТЕЛЬНО РАВНО 1.

ГЛАВА 12 МАССИВЫ =====

КРАТКОЕ СОДЕРЖАНИЕ: DIM

ДОПУСТИМ У ВАС ИМЕЕТСЯ СПИСОК ИЗ ЧИСЕЛ, КАКИМ-ТО ОБРАЗОМ ОПИСЫВАЮЩИХ 10 ЧЕЛОВЕК. ДЛЯ ЗАПИСИ ИХ В ПАМЯТЬ КОМПЬЮТЕРА, ВЫ ДОЛЖНЫ БУДЕТЕ ЗАНЕСТИ ПЕРЕМЕННУЮ НА КАЖДОГО ЧЕЛОВЕКА. ЭТО НЕ УДОБНО, ТАК КАК ПРИХОДИТСЯ ОБРАЩАТЬСЯ К ДАННЫМ, НАЗЫВАЯ КАЖДЫЙ РАЗ НОВУЮ ПЕРЕМЕННУЮ. НАПРИМЕР BLOGGS1, BLOGGS2 И Т.Д. ДО BLOGGS10. КАК ЭТО НЕУДОБНО, ВЫ МОЖЕТЕ УБЕДИТСЯ ИЗ ПРОГРАММЫ:

```
5 REM THIS PROGRAM WILL NOT WORK
10 FOR N=1 TO 10
20 READ BLOGGSN
30 NEXT N
40 DATA 10, 2, 5, 19, 16, 3, 11, 1, 0, 6
```

ИМЕЕТСЯ СПЕЦИАЛЬНЫЙ АППАРАТ ДЛЯ ПОДОБНОГО СЛУЧАЯ, ЭТО ПРИМЕНЕНИЕ МАССИВОВ, ПЕРЕМЕННЫЕ В МАССИВЕ ЯВЛЯЮТСЯ ЕГО ЭЛЕМЕНТАМИ, ОБЛАДАЮТ ОБЩИМ ИМЕНЕМ И РАЗЛИЧАЮТСЯ ТОЛЬКО НОМЕРОМ, ЗАПИСЫВАЕМЫМ ПОСЛЕ ИМЕНИ (ИНДЕКСОМ).

В НАШЕМ ПРИМЕРЕ ИМЯ БУДЕТ В (ПОДОБНО УПРАВЛЯЮЩИМ ПЕРЕМЕННЫМ В FOR-NEXT УТВЕРЖДЕНИЯХ, ИМЯ МАССИВА ДОЛЖНО БЫТЬ УНИКАЛЬНЫМ В ДАННОЙ ПРОГРАММЕ), И ДЕСЯТЬЮ ПЕРЕМЕННЫМИ БУДУТ В(1), В(2) И Т.Д. ДО В(10).

ЭЛЕМЕНТЫ МАССИВОВ НАЗЫВАЮТ ИНДЕКСИРУЕМЫМИ ПЕРЕМЕННЫМИ. ПЕРЕД ИСПОЛЬЗОВАНИЕМ МАССИВА НЕОБХОДИМО ЗАРЕЗЕРВИРОВАТЬ ПОД НЕГО ПАМЯТЬ. ЭТО ДЕЛАЕТСЯ В ОПЕРАТОРЕ DIM (ОТ АНГЛИЙСКОГО DIMENSION). В НАШЕМ СЛУЧАЕ ЭТО БУДЕТ ОПЕРАТОР DIM В(10), КОТОРЫЙ ОПРЕДЕЛЯЕТ МАССИВ С ИМЕНЕМ В И РАЗМЕРНОСТЬЮ 10 (Т.Е. 10 ИНДЕКСИРУЕМЫХ ПЕРЕМЕННЫХ В(1), В(2), В(10) И ПРИСВАИВАЕТ ВСЕМ ЭЛЕМЕНТАМ МАССИВА ЗНАЧЕНИЕ 0.

ИТАК, ТЕПЕРЬ МЫ МОЖЕМ ЗАПИСАТЬ:

```
10 FOR N=1 TO 10
20 READ V(N)
30 NEXT N
40 DATA 10, 2, 5, 19, 3, 11, 1, 0, 5
```

МОЖНО ТАКЖЕ ОБЪЯВЛЯТЬ МАССИВЫ С БОЛЕЕ ЧЕМ ОДНОЙ РАЗМЕРНОСТЬЮ. НАПРИМЕР В ДВУХМЕРНОМ МАССИВЕ ПЕРВЫЙ ИНДЕКС МОЖНО СРАВНИТЬ С НОМЕРОМ СТРОКИ, А ВТОРОЙ С ПОЗИЦИЕЙ В СТРОКЕ. ТАКОЙ МАССИВ ОПИСЫВАЕТ СТРАНИЦУ, ЕСЛИ ВВЕСТИ ТРЕТЬЕ ИЗМЕРЕНИЕ ДЛЯ НОМЕРА СТРАНИЦЫ, ТО МАССИВ БУДЕТ ОПИСЫВАТЬ КНИГУ В ВИДЕ: (НОМЕР СТРАНИЦЫ, НОМЕР СТРОКИ, НОМЕР СТОЛБЦА).

ОБЪЯВИМ ДВУХМЕРНЫЙ МАССИВ 'С' С РАЗМЕРНОСТЬЮ 3 И 5;

```
DIM C(3,5)
```

ЧТО ДАСТ 3*5=15 ИНДЕКСИРУЕМЫХ ПЕРЕМЕННЫХ:

```
C(1,1), C(1,2).....C(1,5)
C(2,1), C(2,2).....C(2,5)
C(3,1), C(3,2).....C(3,5)
```

МОГУТ БЫТЬ ТАКЖЕ СТРОКОВЫЕ МАССИВЫ, СТРОКИ В ТАКИХ МАССИВАХ ОТЛИЧАЮТСЯ ОТ СКАЛЯРНЫХ ТЕМ, ЧТО ИМЕЮТ ФИКСИРОВАННУЮ ДЛИНУ А ПРИСВАИВАНИЕ ИМ ЗНАЧЕНИЯ ОСУЩЕСТВЛЯЕТСЯ С УСЕЧЕНИЕМ СПРАВА И ДОБАВЛЕНИЕМ ДО ПОЛНОЙ ДЛИНЫ ПРОБЕЛАМИ. ИМЯ СТРОКОВОГО МАССИВА ОБРАЗУЕТСЯ ДОБАВЛЕНИЕМ СПРАВА К ИМЕНИ СПЕЦИАЛЬНОГО СИМВОЛА - * (ПЕРЕЧЕРКНУТОЙ БУКВЫ S).

ДОПУСТИМ ВАМ НЕОБХОДИМО ОБЪЯВИТЬ МАССИВ А* НА 5 СТРОК ПО 10 СИМВОЛОВ В КАЖДОЙ, ВЫ ДОЛЖНЫ ЗАПИСАТЬ:

```
DIM A*(5,10)
```

ТЕПЕРЬ ВЫ МОЖЕТЕ ОБРАЩАТЬСЯ КАК ЦЕЛИКОМ К ОТДЕЛЬНОЙ СТРОКЕ, ТАК

И К КАЖДОМУ СИМВОЛУ В СТРОКЕ:

A*(1)=A*(1,1)A*(1,2)...A*(1,10)

A*(2)=A*(2,1)A*(2,2)...A*(2,10)

A*(5)=A*(5,1)A*(5,2)...A*(5,10)

МОЖНО ТАКЖЕ РАССМАТРИВАТЬ ЭЛЕМЕНТ СТРОКОВОГО МАССИВА КАК МАССИВ СИМВОЛОВ. ПУСТЬ ОБЪЯВЛЕН МАССИВ A*(,7), МОЖНО ЗАПИСАТЬ И ТАК A(2)(7). СЛЕДУЮЩАЯ ПРОГРАММА:

10 LET A*(2)='1234567890'

20 PRINT A*(2),A*(2,7)

ДАСТ '123456789 7'.

МОЖНО ИСПОЛЬЗОВАТЬ ТАКЖЕ СЕЧЕНИЯ МАССИВОВ:

A*(2,4 TO 8)=A*(2)(4 TO 8)='45678'

ПОМНИТЕ, ЧТО В СТРОКОВЫХ МАССИВАХ ВСЕ СТРОКИ ИМЕЮТ ФИКСИРОВАННУЮ ДЛИНУ, ЭТУ ДЛИНУ ОПРЕДЕЛЯЕТ ПОСЛЕДНЕЕ ЧИСЛО РАЗМЕРНОСТИ МАССИВА В ОПЕРАТОРЕ DIM. ЕСЛИ ОБЪЯВЛЕН ОДНОМЕРНЫЙ МАССИВ, ТО ОН ОПРЕДЕЛЯЕТ МАССИВ СИМВОЛОВ: DIM A*(10).

ГЛАВА 13

ЛОГИЧЕСКИЕ ОПЕРАЦИИ

=====

КРАТКОЕ СОДЕРЖАНИЕ: AND, OR, NOT.

ЕСЛИ МЫ ВЗГЛЯНЕМ НА ОПИСАННУЮ В ТРЕТЬЕЙ ГЛАВЕ ФОРМУ ОПЕРАТОРА IF : IF УСЛОВИЕ THEN, ТО УВИДИМ, ЧТО 'УСЛОВИЕ' ОПИСЫВАЕТСЯ ОТНОШЕНИЯМИ (=,<,>,>=,<=,<>),

СВЯЗЫВАЮЩИМИ ДВА ЧИСЛА ИЛИ СТРОКИ, ЗДЕСЬ МОЖНО ТАКЖЕ ИСПОЛЬЗОВАТЬ ЛОГИЧЕСКИЕ ОПЕРАЦИИ AND(И), OR(ИЛИ), И NOT(НЕ).

НЕКОТОРОЕ ВЫРАЖЕНИЕ 'И', НЕКОТОРОЕ ДРУГОЕ ВЫРАЖЕНИЕ ИСТИННЫ, ЕСЛИ ИСТИННЫ ОБА ЭТИХ ВЫРАЖЕНИЯ, НАПРИМЕР:

IF A='YES' AND X>0 THEN PRINT X

'X' БУДЕТ НАПЕЧАТАНО ТОЛЬКО ТОГДА, КОГДА

A='YES' И X>0

НЕКОТОРОЕ ВЫРАЖЕНИЕ 'ИЛИ', НЕКОТОРОЕ ДРУГОЕ ВЫРАЖЕНИЕ ИСТИННЫ, ЕСЛИ ИСТИННО ХОТЯ БЫ ОДНО ИЗ ЭТИХ ВЫРАЖЕНИЙ. 'НЕ' ВЫРАЖЕНИЕ ИСТИННО, ЕСЛИ ЛОЖНО САМО ВЫРАЖЕНИЕ И НАОБОРОТ.

OR ИМЕЕТ НИЖШИЙ ПРИОРИТЕТ, ЗАТЕМ ИДЕТ AND, ЗАТЕМ NOT.

УСЛОВИЕ '<>' ОБРАТНО В ЛОГИЧЕСКОМ СМЫСЛЕ УСЛОВИЮ '=', ТО ЕСТЬ: A<>B ТОЖЕ, ЧТО И NOT A=B

NOT A<>B ТОЖЕ, ЧТО И A=B

ТЕМ, КТО БОИТСЯ СЛОЖНОСТЕЙ, СЛЕДУЮЩИЕ РАЗДЕЛЫ МОЖНО ОПУСТИТЬ

1. УСЛОВИЯ =,<,>,<=,>=,<> ДАЮТ ЧИСЛОВЫЙ РЕЗУЛЬТАТ 1 ДЛЯ ИСТИННЫ И 0 ДЛЯ '1=2', КОТОРОЕ ЛОЖНО, И 1 ДЛЯ '1<>2', КОТОРОЕ ИСТИННО.

2. В ОПЕРАТОРЕ 'IF УСЛОВИЕ THEN', САМО УСЛОВИЕ МОЖЕТ БЫТЬ ЧИСЛОВЫМ ВЫРАЖЕНИЕМ, ЕСЛИ ЕГО ЗНАЧЕНИЕ ПОСЛЕ ВЫЧИСЛЕНИЯ РАВНО 0, ТО СЧИТАЕТСЯ, ЧТО ЛОЖЬ, ЕСЛИ ДРУГОЕ ЗНАЧЕНИЕ (ВКЛЮЧАЯ И 1), ТО СЧИТАЕТСЯ ЧТО ЭТО ИСТИНА, ТАКИМ ОБРАЗОМ IF ОПЕРАТОР МОЖНО ПРЕДСТАВИТЬ:

IF УСЛОВИЕ <> 0 THEN...

ОПЕРАЦИИ AND,OR,NOT МОГУТ ТАКЖЕ ИСПОЛЬЗОВАТЬСЯ И В ЧИСЛОВЫХ ВЫРАЖЕНИЯХ:

X AND Y ИМЕЕТ ЗНАЧЕНИЕ X, ЕСЛИ Y=0 И 0, ЕСЛИ Y≠0

X OR Y ИМЕЕТ ЗНАЧЕНИЕ 1, ЕСЛИ Y=0 И X, ЕСЛИ Y=0

NOT Y ИМЕЕТ ЗНАЧЕНИЕ 0, ЕСЛИ Y=0 И 1, ЕСЛИ Y≠0

НАПРИМЕР;

10 INPUT A

20 INPUT B

30 PRINT(A AND A>B)+(B AND A<B)

40 GO TO 10

В КАЖДОЙ ИТЕРАЦИИ БУДЕТ ВЫВОДИТЬСЯ БОЛЬШЕЕ ИЗ ДВУХ ЧИСЕЛ А ИЛИ В ПРИМЕР ИСПОЛЬЗОВАНИЯ OR:

LET TOTAL PRICE=PRICE LESS TAX=(1,15 OR Y*='ZERO RATED'

В УСЛОВНОМ ВЫРАЖЕНИИ МОЖНО ТАКЖЕ ИСПОЛЬЗОВАТЬ СИМВОЛЬНЫЕ СТРОКИ, НО ТОЛЬКО С ОПЕРАЦИЕЙ AND:

X* AND Y ИМЕЕТ ЗНАЧЕНИЕ X*, ЕСЛИ Y=0, И "", ЕСЛИ Y=0

ГДЕ "" - ПУСТАЯ СТРОКА.

ВЫПОЛНИТЕ СЛЕДУЮЩУЮ ПРОГРАММУ, КОТОРАЯ ВВОДИТ ДВЕ СТРОКИ А ЗАТЕМ ВЫВОДИТ ИХ В АЛФАВИТНОМ ПОРЯДКЕ:

10 INPUT "" TYPE IN TWO STRINGS"" A*, B*

20 IF A*>B* THEN LET C*=A*: LET A*=B*: LET B*=C*

30 PRINT A*, "" ("<" AND A*<B*)+("=" AND A*=B*)

40 PRINT "" B*

50 GO TO 10

ГЛАВА 14 НАБОР СИМВОЛОВ =====

КРАТКОЕ СОДЕРЖАНИЕ: CODE, CHR*, ROK, REEK, USR, VIN.

БУКВЫ, ЦИФРЫ, ЗНАКИ ПУНКТУАЦИИ ОБОЗНАЧАЮТСЯ СИМВОЛАМИ И ОБРАЗУЮТ АЛФАВИТ ИЛИ НАБОР СИМВОЛОВ, ИСПОЛЬЗУЕМЫЙ КОМПЬЮТЕРОМ. ОТДЕЛЬНЫЕ СИМВОЛЫ, НАЗЫВАЕМЫЕ ЗНАКАМИ, ОБРАЗУЮТ ЦЕЛЫЕ СЛОВА, НАПРИМЕР PRINT, STOP И Т.Д.

КОМПЬЮТЕР ZX SPECTRUM ИСПОЛЬЗУЕТ 256 СИМВОЛОВ С КОДАМИ ОТ 0 ДО 255, ВСЕ ОНИ ПРИВЕДЕННЫ В ПРИЛОЖЕНИИ А. ДЛЯ ПРЕОБРАЗОВАНИЯ ИЗ СИМВОЛЬНОЙ ФОРМЫ ВО ВНУТРЕННЮЮ КОДОВУЮ И НАОБОРОТ СЛУЖАТ ДВЕ ФУНКЦИИ CODE И CHR*.

CODE ПРИМЕНЯЕТСЯ К СТРОКЕ СИМВОЛОВ И ВОЗВРАЩАЕТ КОД ВНУТРЕННЕГО ПРЕДСТАВЛЕНИЯ ПЕРВОГО СИМВОЛА В СТРОКЕ ИЛИ 0, ЕСЛИ СТРОКА ПУСТАЯ.

CHR* ПРИМЕНЯЕТСЯ К ЧИСЛУ И ВОЗВРАЩАЕТ ОДИН СИМВОЛ, КОД КОТОРОГО ПРЕДСТАВЛЕН ЭТИМ ЧИСЛОМ.

СЛЕДУЮЩАЯ ПРОГРАММА ВЫВОДИТ ВСЕ ОТОБРАЖАЕМЫЕ СИМВОЛЬНЫЕ НАБОРЫ: 10 FOR A=32 TO 255: PRINT CHR* A :NEXT A

ВСЕ ЭТИ СИМВОЛЫ (КРОМЕ ЗНАКА ФУНТА И "C" В КРУЖОЧКЕ) ОБРАЗУЮТ КОД ASCII (AMERICAN STANDART CODES FOR INFORMATION INTERCHANGE).

СЛЕДУЮЩИЕ СИМВОЛЫ НЕ ВХОДЯТ В ASCII, НО ИСПОЛЬЗУЮТСЯ В ZX-SPECTRUM, ПЕРВЫЕ ИЗ НИХ ЭТО 15 ЧЕРНО-БЕЛЫХ ЗНАЧКОВ, НАЗЫВАЕМЫХ ГРАФИЧЕСКИМИ СИМВОЛАМИ И ИСПОЛЬЗУЕМЫХ ДЛЯ ИЗОБРАЖЕНИЯ РИСУНКОВ. ИХ МОЖНО ВВЕСТИ С КЛАВИАТУРЫ ИСПОЛЬЗУЯ ТАК НАЗЫВАЕМЫЙ, ГРАФИЧЕСКИЙ РЕЖИМ, ЕСЛИ ВЫ НАЖМЕТЕ GRAPHICS (CAPS SHIFT 9), ТО КУРСОР ИЗМЕНЯЕТСЯ НА "G". ТЕПЕРЬ ЦИФРОВЫЕ КЛАВИШИ С 1 ПО 8 ВЫДАЮТ ГРАФИЧЕСКИЕ СИМВОЛЫ, ОБОЗНАЧЕННЫЕ НА КЛАВИШАХ, А ЕСЛИ ПРИ ЭТОМ УДЕРЖИВАТЬ SHIFT, ТО ОНИ БУДУТ ВЫДАВАТЬ ИНВЕРСНЫЕ СИМВОЛЫ, Т.Е. ЧЕРНОЕ СТАНОВИТСЯ БЕЛЫМ, А БЕЛОЕ ЧЕРНЫМ.

НЕЗАВИСИМО ОТ SHIFT, КЛАВИША С ЦИФРОЙ "9" ОБЕСПЕЧИВАЕТ ВАМ ВОЗВРАТ К ОБЫЧНОМУ ("L" - КУРСОР) РЕЖИМУ, А КЛАВИША "0" ФУНКЦИЮ DELETE. ПОСЛЕ ГРАФИЧЕСКИХ СИМВОЛОВ НА КЛАВИАТУРЕ РАСПОЛОЖАЮТСЯ СИМВОЛЫ АЛФАВИТА ОТ А ДО У. ГРАФИЧЕСКИЕ ЗНАЧЕНИЯ ЭТИХ КЛАВИШ МОГУТ ОПРЕДЕЛЯТЬСЯ САМИМ ПОЛЬЗОВАТЕЛЕМ, А ЗАТЕМ ИСПОЛЬЗОВАТЬСЯ В ГРАФИЧЕСКОМ РЕЖИМЕ, ОПРЕДЕЛЕНИЕ ГРАФИКИ ЭТИХ КЛАВИШ ПРОИЛЛЮСТРИРУЕМ НА ПРИМЕРЕ ОПРЕДЕЛЕНИЯ БУКВЫ ГРЕЧЕСКОГО АЛФАВИТА "ПН".

1. КАЖДЫЙ СИМВОЛ ПРЕДСТАВЛЯЕТСЯ ТОЧКАМИ В МАТРИЦЕ 8x8, ПОЭТОМУ МЫ ВНАЧАЛЕ НАЧЕРТИМ ДИАГРАММУ, ПРИВЕДЕННУЮ НА РИСУНКЕ. МЫ ОСТАВИМ ПО ОДНОЙ КЛЕТКЕ ПО ПЕРИМЕТРУ СИМВОЛА ДЛЯ ОТДЕЛЕНИЯ

ЕГО ОТ СОСЕДНИХ ЗНАКОВ.

2. ЗАКРЕПИМ ДАННЫЙ СИМВОЛ ЗА КЛАВИШЕЙ 'Р', ТАК, ЧТОБЫ ПРИ НАЖАТИИ КЛАВИШИ В ГРАФИЧЕСКОМ РЕЖИМЕ ВЫДАВАЛСЯ СИМВОЛ 'П'

3. ЗАПРОГРАММИРУЕМ ЭТО ИЗОБРАЖЕНИЕ, КАЖДЫЙ ОПРЕДЕЛЯЕМЫЙ ПОЛЬЗОВАТЕЛЕМ СИМВОЛ ЗАПОМИНАЕТСЯ В ПАМЯТИ ВОСЕМЬЮ ЗНАКАМИ, ПО ОДНОМУ НА КАЖДЫЙ РЯД, ВЫ МОЖЕТЕ ЗАПИСАТЬ ИХ, ИСПОЛЬЗУЯ ФУНКЦИЮ BIN С ОБОЗНАЧЕНИЕМ ЦИФРОЙ 0 ЧИСТОЙ ТОЧКИ И 1 ЗАКРАШЕННОЙ ТОЧКИ.

```
BIN 00000000
BIN 00000000
BIN 00000010
BIN 00111100
BIN 01010100
BIN 00010100
BIN 00010100
BIN 00010100
BIN 00000000
```

ЭТИ ВОСЕМЬ ДВОИЧНЫХ ЧИСЕЛ ЗАПОМИНАЮТСЯ В ПАМЯТИ В ВОСЬМИ ЯЧЕЙКАХ, КАЖДАЯ ИЗ КОТОРЫХ ИМЕЕТ СВОЙ АДРЕС. ДЛЯ НАШЕГО СИМВОЛА АДРЕС ПЕРВОГО ИЗ ВОСЬМИ БАЙТОВ В ГРУППЕ БУДЕТ `USR "P"`, ВТОРОЙ БАЙТ ИМЕЕТ АДРЕС `USR "P"+1` И Т.Д. ДО `USR "P"+7`.

`USR` = ФУНКЦИЯ ПРЕОБРАЗОВАНИЯ СТРОКИ СИМВОЛОВ В АДРЕС ПЕРВОГО БАЙТА В СТРОКЕ. СТРОКОВЫЙ АРГУМЕНТ МОЖЕТ СОДЕРЖАТЬ ЕДИНИЦЕННЫЙ СИМВОЛ, КОТОРЫЙ БУДЕТ ОБОЗНАЧАТЬ СИМВОЛ ОПРЕДЕЛЯЕМЫЙ ПОЛЬЗОВАТЕЛЕМ. ИМЕЮТСЯ И ДРУГИЕ ПРИМЕНЕНИЯ ФУНКЦИИ `USR` С ЧИСЛОВЫМ АРГУМЕНТОМ, НО ОБ ЭТОМ ПОЗЖЕ.

ПОЯСНИМ ВСЕ СКАЗАННОЕ ПРОГРАММОЙ:

```
10 FOR N=0 TO 7
20 INPUT ROW :POKE USR"P"+N:ROW
30 NEXT N
```

ДАННАЯ ПРОГРАММА ВВОДИТ 8 ДВОИЧНЫХ ЧИСЕЛ, ОПРЕДЕЛЯЮЩИХ ГРАФИКУ СИМВОЛА, ЗАКРЕПЛЕННОГО ЗА КЛАВИШЕЙ 'Р'.

ОПЕРАТОР 'POKE' ЗАПИСЫВАЕТ ДАННЫЕ НЕПОСРЕДСТВЕННО В ПАМЯТЬ ОБЫЧНЫЙ АППАРАТ БЕЙСИКА. ОБРАТНЫМ ОПЕРАТОРУ 'POKE' ЯВЛЯЕТСЯ ОПЕРАТОР 'PEEK', КОТОРЫЙ СЛУЖИТ ДЛЯ ОТОБРАЖЕНИЯ СОДЕРЖИМОГО ОБЛАСТИ ПАМЯТИ, НО ОБ ЭТОМ ПОДРОБНЕЕ БУДЕМ ГОВОРИТЬ В ГЛАВЕ 24.

ГРАФИЧЕСКИЕ СИМВОЛЫ

СИМВОЛ	КОД	НАБОР	СИМВОЛ	КОД	НАБОР
	128	<G> 8		143	<G> SH8
	129	<G> 1		142	<G> SH1
	150	<G> 2		141	<G> SH2
	131	<G> 3		140	<G> SH3
	132	<G> 4		139	<G> SH4
	133	<G> 5		138	< G> SH5
	134	<G> 6		137	<G> SH6
	135	<G> 7		136	<G> SH7

ВЕРНЕМСЯ К ЗНАКАМ. МЫ ЕЩЕ НЕ ПОГОВОРИЛИ О НЕ НАПЕЧАТАННЫХ ПЕРВЫХ 32 ЗНАКАХ С КОДАМИ ОТ 0 ДО 32, ЭТО УПРАВЛЯЮЩИЕ СИМВОЛЫ, ОНИ НЕ ОТОБРАЖАЮТСЯ. ВМЕСТО НИХ НА ЭКРАНЕ ТЕЛЕВИЗОРА ОТОБРАЖАЕТСЯ ЗНАК '7'. НАЗНАЧЕНИЕ ЭТИХ СИМВОЛОВ ОПИСАНО В ПРИЛОЖЕНИИ А.

ТРИ СИМВОЛА С КОДАМИ 6, 8 И 13 ИМЕЮТ СПЕЦИАЛЬНОЕ НАЗНАЧЕНИЕ ПРИ РАБОТЕ С ТЕЛЕВИЗОРОМ.

CHRx 6 ПЕЧАТАЕТ ПРОБЕЛ, ИСПОЛЬЗУЕМЫЙ КАК ЗАПЯТАЯ В ОПЕРАТОРЕ PRINT; PRINT 1; CHRx 6; 2 ДАСТ ТОТ ЖЕ РЕЗУЛЬТАТ, ЧТО И ОПЕРАТОР; PRINT 1,2

НО ЭТО НЕ СОВСЕМ КОРРЕКТНОЕ ИСПОЛЬЗОВАНИЕ, ВЕРНЕЕ БУДЕТ СДЕЛАТЬ
 10 LET A="1"+CHR 6+"2"
 20 PRINT A
 CHR 8 - ЭТО СИМВОЛ ЗАБОЯ, ОБЕСПЕЧИВАЕТ ВОЗВРАТ НА ОДНУ
 ПОЗИЦИЮ НАЗАД. ОПЕРАТОР PRINT "1234" CHR 8 "5" ДАСТ СТРОКУ:
 "1234".

CHR 13 - ПЕРЕВОД СТРОКИ, ПРОДОЛЖАЕТ ВЫВОД С НОВОЙ СТРО-
 КИ.

С ТЕЛЕВИЗОРОМ ТАКЖЕ ИСПОЛЬЗУЮТСЯ СИМВОЛЫ С КОДАМИ 16 И
 23 НО ОБ ЭТОМ ПОГОВОРИМ В ГЛАВАХ 15,16.

ВСЕ СИМВОЛЫ РАСПОЛОЖЕНЫ В КОДОВОЙ ТАБЛИЦЕ В АЛФАВИТНОМ
 ПОРЯДКЕ ПО ВОЗРАСТАНИЮ КОДОВ, ПРИЧЕМ ВСЕ ПРОПИСНЫЕ БУКВЫ РАСПО-
 ЛОЖЕНЫ ПОСЛЕ ЗАГЛАВНЫХ, ТАК, ЧТО 'А' СЛЕДУЕТ ПОСЛЕ 'Z'.

CHR 3+"ZOOLOGICAL GARDENS"

CHR 8+"AARDYARK HURTING"

"(PARENTHETICAL REMARK)"

"100"

"129,95 INC,YAT"

"AASYOGEL"

"AARDYARK"

"PRINT"

"ZOO"

"LINTERPOLATIONJ"

"AARDYARK"

"AASYOGEL"

"ZOO"

"ZOOLOGU"

СУЩЕСТВУЕТ ПРАВИЛО ПО КОТОРОМУ СОРТИРУЮТСЯ ДВЕ СТРОКИ.
 СНАЧАЛА СРАВНИВАЮТСЯ ПЕРВЫЕ СИМВОЛЫ. ЕСЛИ ОНИ РАЗЛИЧАЮТСЯ, ТО
 СТРОКА, СОДЕРЖАЩАЯ СИМВОЛ С МЕНЬШИМ КОДОМ ЯВЛЯЕТСЯ 'МЕНЬШЕЙ', А
 ЕСЛИ РАВНЫ ТО ВЫБИРАЕТСЯ ДЛЯ СРАВНЕНИЯ СЛЕДУЮЩАЯ ПАРА СИМВОЛОВ.
 ТАК ДО ТЕХ ПОР ПОКА ВСТРЕТЯТСЯ НЕСОВПАДАЮЩИЕ СИМВОЛЫ, ЛИБО ПОКА
 ОДНА ИЗ СТРОК НЕ КОНЧИТСЯ, ОНА И БУДЕТ МЕНЬШЕЙ, В ПРОТИВНОМ СЛУ-
 ЧАЕ СТРОКИ СЧИТАЮТСЯ РАВНЫМИ.

ОТНОШЕНИЯ =,<,>,<=,>=<,> ПРИМЕНЯЮТСЯ К СТРОКАМ СИМВОЛОВ
 ТАКЖЕ КАК И К ЧИСЛАМ: ЗНАК '<' ОЗНАЧАЕТ 'НАХОДИТСЯ ВПЕРЕДИ В КО-
 ДОВОЙ ТАБЛИЦЕ', А '>' - 'НАХОДИТСЯ ПОЗАДИ', ТАК, ЧТО ВЫРАЖЕНИЯ

"AAMAN" < "AARDYARK"

"AARDYARK" > "AAMAN" ОБА ИСТИНЫ.

ДЛЯ ИЛЛЮСТРАЦИИ ВСЕГО СКАЗАННОГО ПРИВЕДЕМ ПРОГРАММУ, КО-
 ТОРАЯ ВВОДИТ ДВЕ СТРОКИ, А ЗАТЕМ ВЫВОДИТ ИХ В УПОРЯДОЧЕННОМ ВИДЕ

```
10 INPUT "TYPE IN TWO STRING ", A,B
20 IF A>B THEN LET C=A:LET A=B:LET B=C
30 PRINT A " "
40 IF A<B THEN PRINT " < " : GO TO 60
50 PRINT " = "
60 PRINT " " B
70 GO TO 10
```

СЛЕДУЮЩАЯ ПРОГРАММА ЗАКРЕПЛЯЕТ ОПРЕДЕЛЕННЫЕ ПОЛЬЗОВАТЕ-
 ЛЕМ СИМВОЛЫ ДЛЯ ИГРЫ В ШАХМАТЫ ЗА КЛАВИШАМИ:

F - ЗА ПЕШКОЙ (PAWN)
 R - ЗА ЛАДЬЕЙ (ROCK)
 N - ЗА КОНЕМ (KNIGHT)
 B - ЗА СЛОНОМ (BISHOP)
 K - ЗА КОРОЛЕМ (KING)
 Q - ЗА КОРОЛЕВОЙ (QUEEN)

```
5 LET B=BIN 01111100:LET S= BIN 00111000:LET D=BIN 00010000
10 FOR N=1 TO 6 : READ P: REM 6 PIECES
20 FOR N=1 TO 7 :REM READ PIECE INTO 8 BUTES
30 READ A: POKE USR P+1,A
```

```

40 NEXT F
50 NEXT N
100 REM BISHOP
110 DATA " B ", 0, 0, BIN 00101000, BIN 01000100
120 DATA BIN 01101100, C, B, 0
130 REM KING
140 DATA " K ", 0, 0, D, C, D
150 DATA C, BIN 01000100, C, 0
160 REM ROCK
170 DATA " R ", 0, BIN 01010100
180 DATA C, B, B, 0
190 REM QUEEN
200 DATA " Q ", 0, BIN 01010100, BIN 00101000, D
210 DATA BIN 01101100, B, B, 0
220 REM POWN
230 DATA " P ", 0, 0, D, C
240 DATA C, D, 0, 0
250 REM KNIGHT
260 DATA " K ", 0, D, C, BIN 01111000
270 DATA BIN 00011000, C, B, 0

```

ГЛАВА 15 ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ ОБ ОПЕРАТОРАХ PRINT И INPUT

=====

ФИЗИЧЕСКОЕ СОДЕРЖАНИЕ: CLS, PRINT - ПАРАМЕТРЫ: ИХ ОТСУТ-
СТВИЕ - ВООБЩЕ, ВЫРАЖЕНИЕ (ЧИСЛОВОЕ ИЛИ СТРОКОВОЕ)

ТАВ ЧИСЛОВОЕ ВЫРАЖЕНИЕ, АТ ЧИСЛОВОЕ ВЫРАЖЕНИЕ,

PRINT РАЗДЕЛИТЕЛИ: ", ", " ", " ", " ",

INPUT - ПАРАМЕТРЫ,

LINE СТРОЧНАЯ ПЕРЕМЕННАЯ,

SCREEN* - СВЕРТКА

ВЫРАЖЕНИЯ, ЗНАЧЕНИЕ КОТОРЫХ ИСПОЛЬЗУЕТСЯ В ОПЕРАТОРЕ
PRINT, НАЗЫВАЮТСЯ PRINT-ПАРАМЕТРАМИ, ОНИ РАЗДЕЛЯЮТСЯ ЗАПЯТЫМИ
ИЛИ ТОЧКОЙ С ЗАПЯТОЙ, НАЗЫВАЕМЫМИ PRINT-РАЗДЕЛИТЕЛЯМИ.

В ОПЕРАТОРЕ PRINT ВОЗМОЖНО ОТСУТСТВИЕ НЕКОТОРЫХ ПАРАМЕТ-
РОВ, В ЭТОМ СЛУЧАЕ СТАВЯТСЯ ДВЕ ЗАПЯТЫЕ ПОДРЯД.

PRINT AT 11,16 "*"

ВЫВЕДЕТ "*" В СЕРЕДИНЕ ЭКРАНА.

АТ "СТРОКА", "СТОЛБЕЦ"

ЭТОТ ПАРАМЕТР ПЕРЕМЕЩАЕТ ПОЗИЦИЮ ВЫВОДА В МЕСТО, ОПРЕДЕЛЯЕМОЕ
НОМЕРОМ СТРОКИ И СТОЛБЦА, НОМЕР СТРОКИ МЕНЯЕТСЯ ОТ 0 (ВЕРХНЯЯ)
ДО 21, А НОМЕР СТОЛБЦА ОТ 0 (ЛЕВЫЙ) ДО 31,

ОПЕРАТОР SCREEN*, ЕГО ДЕЙСТВИЕ ПРОТИВОПОЛОЖНО ДЕЙСТВИЮ

ОПЕРАТОРА PRINT AT, ОН ИСПОЛЬЗУЕТ ТЕ ЖЕ ПАРАМЕТРЫ, НОМЕР СТРОКИ
И СТОЛБЦА, НО ИХ ЗНАЧЕНИЯ ЗАКРЫВАЮТСЯ В СКОБКИ.

ОПЕРАТОР SCREEN* СООБЩАЕТ ВАМ КАКОЙ СИМВОЛ НАХОДИТСЯ НА ЭКРАНЕ В
УКАЗАННОЙ ПОЗИЦИИ, ВЫПОЛНИВ: PRINT SCREEN* (11,16)

МЫ ПОЛУЧИМ "*" ВЫВЕДЕННУЮ ПРЕДЫДУЩИМ ОПЕРАТОРОМ.

В КАЧЕСТВЕ ВОЗВРАЩАЕМОГО ЗНАЧЕНИЯ МОГУТ ИСПОЛЬЗОВАТЬСЯ
АЛФАВИТНО-ЦИФРОВЫЕ СИМВОЛЫ, СПЕЦИАЛЬНЫЕ СИМВОЛЫ, ПРОБЕЛЫ, ЛИНИИ,
НАРИСОВАННЫЕ С ПОМОЩЬЮ ОПЕРАТОРОВ: PLOT, DRAW, CIRCLE, ОПРЕДЕЛЯЕ-
МЫЕ ПОЛЬЗОВАТЕЛЕМ СИМВОЛЫ И ГРАФИЧЕСКИЕ СИМВОЛЫ, ВОЗВРАЩАЕМЫЕ
КАК ПУСТАЯ СТРОКА.

ТО ЖЕ, КОГДА ФУНКЦИЯ OVER ИСПОЛЬЗУЕТСЯ ДЛЯ ПОЛУЧЕНИЯ КОМБИНИРОВА-
ННЫХ ЗНАКОВ.

ТАВ СТОЛБЕЦ

ЭТОТ ПАРАМЕТР ПЕРЕМЕЩАЕТ ПОЗИЦИЮ ВЫВОДА В УКАЗАННЫЙ СТОЛБЕЦ В ТОЙ
ЖЕ СТРОКЕ, ИЛИ ПЕРЕХОДИТ НА НОВУЮ СТРОКУ, ЕСЛИ СТОЛБЕЦ БЫЛ ПОСЛЕ
ДНИМ.

ПОМНИТЕ, ЧТО КОМПЬЮТЕР УМЕНЬШАЕТ НОМЕР ПОЗИЦИИ ПО МОДУЛЮ 32 (Т.Е. ДЕЛИТ НА 32 И ИСПОЛЬЗУЕТ ОСТАТОК). ТАК 'ТАБ 33' РАВНОЗНАЧНО 'ТАБ 1'. К ПРИМЕРУ:

```
PRINT TAB 30,1 TAB12 "CONTENTS" AT 3,1 "CHAPTER" TAB 24 "PAGE"
ВЫВЕДЕТ НА ЭКРАН ОГЛАВЛЕНИЕ ДЛЯ ПЕРВОЙ СТРАНИЦЫ КНИГИ.
```

РАССМОТРИМ ПРИМЕР, ИЛЛЮСТРИРУЮЩИЙ УМЕНЬШЕНИЕ ПО МОДУЛЮ 32;

```
10 FOR N=0 TO 20
20 PRINT TAB 8*N N
30 NEXT N
```

БОЛЕЕ НАГЛЯДНЫЙ ПРИМЕР ПОЛУЧАЕТСЯ ПРИ ЗАМЕНЕ В СТРОКЕ 20 '8' НА '6'. НЕСКОЛЬКО ЗАМЕЧАНИЙ:

1) В РАССМОТРЕННЫХ ПРИМЕРАХ В КАЧЕСТВЕ ОГРАНИЧИТЕЛЕЙ ИСПОЛЬЗОВАЛОСЬ ' ', МОЖНО ИСПОЛЬЗОВАТЬ ' ' (ИЛИ ВОООБЩЕ НИЧЕГО).

ПРИ ЭТОМ НЕОБХОДИМО СЛЕДИТЬ ЗА УСТАНОВКОЙ ТЕКУЩЕЙ ПОЗИЦИИ .

2) НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ ДЛЯ ВЫВОДА ДВЕ НИЖНИЕ СТРОКИ ЭКРАНА Т.К. ОНИ ИСПОЛЬЗУЮТСЯ ДЛЯ ПОЛУЧЕНИЯ ОПЕРАТОРОМ INPUT ДАННЫХ, ПОСЛЕДНЯЯ ИСПОЛЬЗУЕМАЯ СТРОКА -21.

3) МОЖНО ИСПОЛЬЗОВАТЬ ПАРАМЕТР AT ДЛЯ УСТАНОВКИ ПОЗИЦИИ ВЫВОДА В ТО МЕСТО, ГДЕ УЖЕ ИМЕЕТСЯ ВЫВЕДЕННАЯ ИНФОРМАЦИЯ, ПРИ ЭТОМ КАЖДЫЙ НОВЫЙ СИМВОЛ УНИЧТОЖАЕТ СТАРЫЙ.

ЕЩЕ ОДНИМ ОПЕРАТОРОМ ИСПОЛЬЗУЕМЫМ СОВМЕСТНО С PRINT ЯВЛЯЕТСЯ CLS. ОН ПРОИЗВОДИТ ОЧИСТКУ ЭКРАНА ПОДОБНО ОПЕРАТОРАМ CLEAR И 9.

ПРИ ЗАПОЛНЕНИИ ВСЕГО ЭКРАНА ПРОИСХОДИТ ЕГО СВЕРТКА. В ЭТОМ МОЖНО УБЕДИТСЯ ПРОДЕЛАВ:

```
CLS: FOR N=1 TO 22: PRINT N: NEXT N
```

И ДАЛЕЕ ВЫПОЛНИТЬ 'PRINT 99' НЕКОТОРОЕ КОЛИЧЕСТВО РАЗ, ИЛИ ВАРИАНТ С ОСТАНОВКОЙ ВЫВОДА, ДЛЯ ПРОСМОТРА ТЕКСТА. ЧТОБЫ УБЕДИТЬСЯ В ЭТОМ ВЫПОЛНИМ:

```
CLS: FOR N=1 TO 100: PRINT N: NEXT N
```

КОГДА ЭКРАН ЗАПОЛНИТСЯ, ВЫВОД ОСТАНОВИТСЯ И В НИЖНЕЙ ЧАСТИ ЭКРАНА ПОЯВИТСЯ ЗАПРОС: "SCROLL Y", ПОСЛЕ ПРОСМОТРА НАЖМИТЕ "Y" (ДА). ВЫВОД ПРОДОЛЖИТСЯ. ВОЗМОЖЕН ОТРИЦАТЕЛЬНЫЙ ОТВЕТ 'N' (НЕТ) (SYMBOL SHIFT И A) ИЛИ SPACE (BREAK). В ПОСЛЕДНЕМ СЛУЧАЕ КОМПЬЮТЕР ОСТАНОВИТ ПРОГРАММУ И ВЫДАСТ СООБЩЕНИЕ: 'D BREAK-CONT PER?

ОПЕРАТОР INPUT ИСПОЛЬЗУЕТСЯ ДЛЯ ВВОДА РАЗЛИЧНЫХ ЗНАЧЕНИЙ, НАПРИМЕР:

```
INPUT "NO" OLD ARE YOU?", AGE
```

КОМПЬЮТЕР ВЫВЕДЕТ НА ЭКРАН (В НИЖНЕЙ ЧАСТИ) ВОПРОС, В ОТВЕТ НА КОТОРЫЙ ВЫ ДОЛЖНЫ ВВЕСТИ СВОЙ ВОЗРАСТ.

ФАКТИЧЕСКИ INPUT СОДЕРЖИТ ТЕ ЖЕ ПАРАМЕТРЫ, ЧТО И PRINT. ТАК "NO-OLD ARE YOU?" И "AGE" ОБА ЯВЛЯЮТСЯ ПАРАМЕТРАМИ INPUT, ОДНАКО СУЩЕСТВУЮТ И НЕКОТОРЫЕ ОТЛИЧИЯ.

ПЕРВОЕ: ЭТО ДОПОЛНИТЕЛЬНЫЙ ПАРАМЕТР-ПЕРЕМЕННАЯ, ЗНАЧЕНИЕ КОТОРОЙ ВЫ ДОЛЖНЫ ПРИСВОИТЬ (В НАШЕМ ПРИМЕРЕ " AGE ").

ВТОРОЕ: ВЫ МОЖЕТЕ ВЫВОДИТЬ ЗНАЧЕНИЕ ПЕРЕМЕННОЙ, КАК ЧАСТЬ ЗАПРОСА, ЗАКЛЮЧИВ ЕЕ ДЛЯ ЭТОГО В СКОБКИ.

```
ПРИМЕР: LET MY AGE=INT(RND*100):INPUT("I AM, MY AGE ",")
```

```
"HOW OLD ARE YOU?, YOUR AGE?"
```

ЗНАЧЕНИЕ 'MY AGE' ВЫДАЕТ КОМПЬЮТЕР, ЗНАЧЕНИЕ 'YOUR AGE' ВВОДИТЕ ВЫ САМИ. ПО МЕРЕ ВЫДАЧИ ОПЕРАТОРОВ INPUT ПРОИСХОДИТ СВЕРТКА ЭКРАНА. РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ 'AT' В INPUT ОПЕРАТОРЕ:

```
10 INPUT "THIS IS LINE 1", A* AT 0,0 "THIS LINE IS 0",
A* AT 2,0 " THIS IS LINE 2", A* AT 1,0
"THIS IS STILL LINE 1" A*
```

КОГДА "THIS IS LINE 2" БУДЕТ ВЫВЕДЕНО, НИЖНИЕ СТРОКИ СТАНУТ СДВИГАТЬСЯ ВВЕРХ, ОСВОБОЖДАЯ МЕСТО ДЛЯ НУМЕРАЦИИ. ВЫПОЛНИМ:

```
10 FOR N=0 TO 19: PRINT AT N,0 N NEXT N
20 INPUT AT 0,0 A* AT 1,0 A* AT 2,0 A* AT 3,0 A* AT 4,0
A* AT 5,0 A*
```

КОГДА ИНФОРМАЦИЯ НАЧНЕТ СМЕЩАТЬСЯ В ОБЛАСТЬ ДЕЙСТВИЯ ОПЕРАТОРОВ PRINT ПРОИЗОЙДЕТ СВЕРТКА ЭКРАНА. ЕДИНСТВЕННЫМ ПАРАМЕТРОМ ОПЕРАТОРА INPUT ЯВЛЯЕТСЯ LINE, ОН НЕ ПРЕДНАЗНАЧЕН ДЛЯ ВВОДА СТРОЧНЫХ ПЕРЕМЕННЫХ, РАССМОТРИМ ПРИМЕР: INPUT LINE A* ЕСЛИ ВВЕСТИ КАКУЮ ЛИБО СТРОЧНУЮ ПЕРЕМЕННУЮ БЕЗ СТРОКОВЫХ КАВЫЧЕК ТО ЕЕ ЗНАЧЕНИЕ БУДЕТ ПРИСВОЕНО A*. ЗАМЕТИМ, ЧТО МЫ НЕ МОЖЕМ ИСПОЛЬЗОВАТЬ ПАРАМЕТР LINE ДЛЯ ЧИСЛОВЫХ ПЕРЕМЕННЫХ.

УПРАВЛЯЮЩИЕ СИМВОЛЫ CHR*22 И CHR*23 ВЫПОЛНЯЮТ ФУНКЦИИ ПОДОБНЫЕ ПАРАМЕТРАМ TAB И AT. ИХ ПРЕИМУЩЕСТВО СОСТОИТ В ТОМ, ЧТО МОЖНО ИСПОЛЬЗОВАТЬ ИМЕНА ПЕРЕМЕННЫХ, А ДЛЯ TAB И AT ЭТО НЕВОЗМОЖНО, ЭТИ УПРАВЛЯЮЩИЕ СИМВОЛЫ ОБРАБАТЫВАЮТСЯ КАК ЧИСЛА. АНАЛОГОМ AT ЯВЛЯЕТСЯ УПРАВЛЯЮЩИЙ СИМВОЛ CHR*22, ПЕРВОЕ ЗНАЧЕНИЕ ОПРЕДЕЛЯЕТ СТРОКУ, ВТОРОЕ СТОЛБЕЦ: PRINT CHR*22+CHR*1+CHR*C ТО ЖЕ, ЧТО И PRINT AT 1,C КАК ЗНАЧЕНИЯ ПАРАМЕТРОВ РАССМАТРИВАЮТСЯ ТОЛЬКО CHR*1 И CHR*C. (CHR*22 НЕ УЧИТЫВАЕТСЯ).

АНАЛОГОМ TAB ЯВЛЯЕТСЯ УПРАВЛЯЮЩИЙ СИМВОЛ CHR*23, ЗНАЧЕНИЯ ЗАДАВАЕМЫХ ИМ ПАРАМЕТРОВ НАХОДЯТСЯ В ПРЕДЕЛАХ ОТ 0 ДО 65535 *23+CHR*A+CHR*B

ТО ЖЕ, ЧТО И PRINT TAB A+256*B

ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ РОКЕ ДЛЯ ОСТАНОВКИ КОМПЬЮТЕРА, ЗАПРАШИВАЮЩЕГО СВЕРТКУ, ВЫПОЛНИВ: РОКЕ 23692,255. КОМПЬЮТЕР СТАНЕТ СВОРАЧИВАТЬ ЭКРАН БЕЗ ЗАПРОСА 255 РАЗ, ПРЕЖДЕ ЧЕМ ЗАПРОСИТ СВЕРТКУ. ТАК, НАПРИМЕР, ЗАПУСТИТЕ:

```
10 FOR N=0 TO 10000
20 PRINT N: РОКЕ 23692,255
30 NEXT N
```

И СЛЕДИТЕ, СКОЛЬКО СВЕРТОК СДЕЛАЛ КОМПЬЮТЕР.

СТАРТУЕМ СЛЕДУЮЩУЮ ПРОГРАММУ ПРОВЕРЯЮЩУЮ ЗНАНИЕ ТАБЛИЦЫ УМНОЖЕНИЯ

```
10 LET A*= " "
20 LET A=INT(RND*12)+1 LET B=INT(RND*12)+1
30 INPUT(A)* " WHAT IS" (A) "*" (B) "?" C
100 IF C=A*B THEN LET A*="RIGHT": GO TO 20
111 LET A*="WRONG, TRY AGAIN,": GO TO 30
```

МОЖНО НЕСКОЛЬКО ИЗМЕНИТЬ ПРОГРАММУ, ТАК ЧТОБЫ НЕ ЗНАЯ ОТВЕТА МОЖНО БЫЛО УЗНАТЬ ЕГО. К ПРИМЕРУ, КОМПЬЮТЕР СПРАШИВАЕТ СКОЛЬКО БУДЕТ 2*3, НЕ ЗНАЯ ОТВЕТА ВЫ ВВОДИТЕ 2*3 И ПОЛУЧАЕТЕ ЕГО. ДЛЯ ЭТОГО ЗАМЕНИТЕ В 30 СТРОКЕ 'C' НА 'C*', В 100 СТРОКЕ НА 'VAL C*' И ДОПОЛНИТЕЛЬНО ВВЕДИТЕ СТРОКУ:

```
40 IF C* < > STR* VAL C* THEN LET N*="TYPE
IT PROPERLU, A NUMBER,": GO TO
```

ДЛЯ ИСКЛЮЧЕНИЯ ПОДСКАЗКИ ПОМЕНЯЙТЕ 'C*' В СТРОКЕ 30 НА 'LINE C*'

ГЛАВА 16

ЦВЕТА

=====

КРАТКОЕ СОДЕРЖАНИЕ: INK, PAPER, FLASH, BRIGHT, INVERSE, OVER, BORDER
ВЫПОЛНИМ СЛЕДУЮЩУЮ ПРОГРАММУ:

```
10 FOR M=0 TO 1: BRIGHT M
20 FOR N=1 TO 10
30 FOR C=0 TO 7
40 PAPER C:PRINT"" REM 4 COLOURED SPACES
50 NEXT C. NEXT N: NEXT M
60 FOR C=0 TO 1: BRIGHT M: PAPER 7
70 FOR C=0 TO 3
80 INK C:PRINT C " "
90 NEXT C:PAPER 0
100 FOR C=4 TO 7
110 INK C:PRINT C " "
```

120 NEXT C:NEXT M

130 PAPER 7: INK 0: BRIGHT 0

ОНА ПРОДЕМОНСТРИРУЕТ ВАМ ВОЗМОЖНОСТИ ВЫВОДА КОМПЬЮТЕРОМ ZX-SPECTRUM НА ЦВЕТНОЙ ТЕЛЕВИЗОР ВОСЬМИ ЦВЕТОВ (ВКЛЮЧАЯ ЧЕРНЫЙ И БЕЛЫЙ) И ДВУХ УРОВНЕЙ ЯРКОСТИ. ЕСЛИ ТЕЛЕВИЗОР ЧЕРНО-БЕЛЫЙ, ВЫ УВИДИТЕ РАЗЛИЧНЫЕ ГРАДАЦИИ СЕРОГО ЦВЕТА.

НИЖЕ ДАНА КОДИРОВКА ЦВЕТОВ:

- 0 - ЧЕРНЫЙ
- 1 - СИНИЙ
- 2 - КРАСНЫЙ
- 3 - ФИОЛЕТОВЫЙ
- 4 - ЗЕЛЕНый
- 5 - ГОЛУБОЙ
- 6 - ЖЕЛТЫЙ
- 7 - БЕЛЫЙ

ДЛЯ ЧЕРНО-БЕЛОГО ТЕЛЕВИЗОРА ЭТОТ РЯД ПРЕДСТАВЛЯЕТ СОБОЙ ПОСЛЕДОВАТЕЛЬНОСТЬ ПЕРЕХОДА СЕРЫХ ПОЛУТОНОВ ОТ ЧЕРНОГО ДО БЕЛОГО. ДЛЯ ИСПОЛЬЗОВАНИЯ ЦВЕТОВ УЯСНИМ СТРОЕНИЕ ГРАФИЧЕСКОГО ЭКРАНА. ОН СОСТОИТ ИЗ 768 ПОЗИЦИЙ (24 СТРОКИ ПО 32 ЗНАКОМЕСТА), КАЖДАЯ ИЗ КОТОРЫХ ПРЕДСТАВЛЯЕТ ИЗ СЕБЯ МАТРИЦУ 5 НА 8 ПИКСЕЛЕЙ. ВСПОМНИМ

- 0 - БЕЛАЯ ТОЧКА
- 1 - ЧЕРНАЯ ТОЧКА

ПОЗИЦИЯ СИМВОЛА (ЗНАКОМЕСТО) ТАКЖЕ РАССМАТРИВАЕТСЯ С ЭТИХ ПОЗИЦИЙ: INK-ЦВЕТ ТОНА, PAPER-ЦВЕТ ФОНА. Т.О. ЗНАКОМЕСТО СОСТОИТ ИЗ INK И PAPER (РИСУНОК НА БУМАГЕ), МОЖНО ТАКЖЕ ГОВОРИТЬ О INK И PAPER ОБЫЧНОЙ И ПОВЫШЕННОЙ ЯРКОСТИ, А ТАКЖЕ МЕРЦАЮЩИХ И НЕМЕРЦАЮЩИХ ЗНАКОМЕСТАХ. ВСЕ ЭТО ИМЕЕТ КОДИРОВКУ:

1) ДЛЯ ЗНАКОМЕСТА (8 НА 8 ПИКСЕЛЕЙ) ФОРМУ СИМВОЛА ОПРЕДЕЛЯЮТ ЧИСТЫЕ И ЗАКРАШЕННЫЕ ТОЧКИ (0 И 1), ЦВЕТА ФОНА И ТОНА ОПРЕДЕЛЯЮТСЯ PAPER И INK.

2) ЦВЕТА ФОНА И ТОНА КОДИРУЕТСЯ ОТ 0 ДО 7 КАЖДЫЙ.

3) ЯРКОСТЬ: 0-ОБЫЧНАЯ, 1-ПОВЫШЕННАЯ

4) МЕРЦАНИЕ: 0-ПОСТОЯННО, 1-МЕРЦАНИЕ.

ЗАМЕТИМ, ЧТО ДЛЯ ОДНОГО ЗНАКОМЕСТА В 64 ПИКСЕЛЯ МЫ НЕ МОЖЕМ УСТАНОВИТЬ БОЛЕЕ ОДНОГО ЦВЕТА ДЛЯ ФОНА И ОДНОГО ЦВЕТА ДЛЯ ТОНА. ЭТО ЖЕ ОТНОСИТСЯ И К ЯРКОСТИ, И К МЕРЦАНИЮ. ЦВЕТ, ЯРКОСТЬ И МЕРЦАНИЕ ЗАДАЮТСЯ ДЛЯ ЗНАКОМЕСТА (А НЕ ДЛЯ ОТДЕЛЬНОГО ПИКСЕЛЯ) И ЯВЛЯЮТСЯ ЕГО АТТРИБУТАМИ. ДЛЯ ИЗМЕНЕНИЯ ЭТИХ АТТРИБУТОВ ПРЕДНАЗНАЧЕНЫ ОПЕРАТОРЫ: INK, PAPER, BRIGHT, FLASH.

ВЫПОЛНИМ: PAPER 5

НА ЭКРАНЕ БУДЕТ ГОЛУБОЙ ФОН (Т.К. 5- КОД ГОЛУБОГО ЦВЕТА).
ФОРМАТ ОПЕРАТОРОВ:

PAPER ЧИСЛО ОТ 0 ДО 7

INK ЧИСЛО ОТ 0 ДО 7

BRIGHT ЧИСЛО 0 ИЛИ 1 (0-ВЫКЛ. 1-ВКЛ.)

FLASH ЧИСЛО 0 ИЛИ 1 (0-ВЫКЛ. 1-ВКЛ.)

ОТМЕТИМ ЧТО ИСПОЛЬЗОВАНИЕ ЧИСЕЛ БОЛЬШИХ, ЧЕМ УКАЗЫВАЛОСЬ ВЫШЕ, ДОПУСТИМО, НО ДАЕТ ДРУГОЙ ЭФФЕКТ.

К ПРИМЕРУ - '8' МОЖЕТ ИСПОЛЬЗОВАТЬСЯ ВО ВСЕХ ЧЕТЫРЕХ ОПЕРАТОРАХ КАК СРЕДСТВО, ПОЗВОЛЯЮЩЕЕ ОПРЕДЕЛИТЬ ЗНАЧЕНИЕ РАНЕЕ УСТАНОВЛЕННЫХ АТТРИБУТОВ. ТАК PAPER 8 НЕ ИЗМЕНИТ ЦВЕТА ФОНА (Т.К. ТАКОГО ЦВЕТА НЕТ), А ПОМОЖЕТ ВЫЯСНИТЬ ЗНАЧЕНИЕ ПРЕДЫДУЩЕГО PAPER. ОПЕРАТОРЫ INK 8, BRIGHT 8, FLASH 8 ВЫДАДУТ ЗНАЧЕНИЯ ЭТИХ АТТРИБУТОВ. "9" МОЖЕТ ИСПОЛЬЗОВАТЬСЯ ТОЛЬКО ДЛЯ INK И PAPER, КАК СРЕДСТВО "КОНТРАСТА". ЦВЕТА INK И PAPER, КОТОРЫЕ ВЫ ИСПОЛЬЗУЕТЕ, ДОЛЖНЫ БЫТЬ КОНТРАСТНЫ ДРУГ ДРУГУ. ТАК К БЕЛОМУ ЦВЕТУ ПОДХОДЯТ ТЕМНЫЕ ТОНА: ЧЕРНЫЙ, СИНИЙ, КРАСНЫЙ, ФИОЛЕТОВЫЙ К ЧЕРНОМУ ЦВЕТУ ПОДХОДЯТ

СВЕТЛЫЕ ТОНА: ЗЕЛЕНый, ГОЛУБОЙ, ЖЕЛТЫЙ, БЕЛЫЙ.

ВЫПОЛНИМ :

INK 9 : FOR C = 0 TO 7 : PAPER C : PRINT C : NEXT C

МОЖНО ЗАПУСТИТЬ ПРОГРАММУ ВЫДАВЛУК НА ЭКРАН ДИСПЛЕЯ ЦВЕТНЫЕ ПОЛОСЫ; INK 9:PAPER 6:PRINT AT 0,0 :FOR N=1 TO 1000:

PRINT N : NEXT N

ЦВЕТ ФОНА БУДЕТ КОНТРАСТЕН ЦВЕТУ ТОНА В КАЖДОЙ ВЫВОДИМОЙ ПОЗИЦИИ ЦВЕТНОЙ ТЕЛЕВИЗОР ПОСТРОЕН НА СПОСОБНОСТИ ЧЕЛОВЕЧЕСКОГО ГЛАЗА ВОСПРИНИМАТЬ ТОЛЬКО ТРИ ПЕРВИЧНЫХ ЦВЕТА: СИНИЙ, КРАСНЫЙ, ЗЕЛЕНый ДРУГИЕ ЦВЕТА ОБРАЗУЮТСЯ ИЗ ИХ СОЧЕТАНИЙ, К ПРИМЕРУ, ФИОЛЕТОВЫЙ ЦВЕТ ОБРАЗУЕТСЯ, КАК КОМБИНАЦИЯ СИНЕГО С КРАСНЫМ (КОД ФИОЛЕТОВОГО ЦВЕТА '3', ОН ЯВЛЯЕТСЯ СУММОЙ КОДОВ СИНЕГО '1' И КРАСНОГО '2' ВИДЕТЬ ВСЕ ВОСЕМЬ ЦВЕТОВ НА ОДНОМ УЧАСТКЕ ЭКРАНА НЕВОЗМОЖНО Т.К. ЭТО БУДЕТ ТЕМНОЕ ПЯТНО, НО ТАМ, ГДЕ ЦВЕТА ЧАСТИЧНО НАКЛАДЫВАЮТСЯ ДРУГ НА ДРУГА МЫ УВИДИМ ЦВЕТОВУЮ ГАММУ. В КАЧЕСТВЕ ПРИМЕРА ВЫПОЛНИМ ПРОГРАММУ (ОТМЕТИМ, ЧТО INK ПОЛУЧЕНА С ИСПОЛЬЗОВАНИЕМ SHIFT И 8 В G-РЕЖИМЕ):

```
10 BORDER 0:PAPER 0: INK 7: CLS
20 FOR A=1 TO 6
30 PRINT TAB 6 INK 1 " ( ) ( ) . . . ( ) ( ) ":REM 18 INK SQUARES
40 NEXT A
50 LET DATALINE=200
60 GO SUB 1000
70 LET DATALINE=210
80 GO SUB 1000
90 STOP
200 DATA 2,3,7,5,4
210 DATA 2,2,6,4,4
1000 FOR A=1 TO 6
1010 RESTORE DATALINE
1020 FOR B=1 TO 5
1030 READ C: PRINT INK C " ( ) . . . ( ) ": REM 6 INK SQUARES
1040 NEXT B: PRINT :NEXT A
1050 RETURN
```

СУЩЕСТВУЕТ ФУНКЦИЯ ATTR, ПОЗВОЛЯЮЩАЯ ОПРЕДЕЛИТЬ КАКИЕ АТТРИБУТЫ БЫЛИ ЗАДАНЫ ДЛЯ ПОЗИЦИЙ ЭКРАНА. ЭТО СЛОЖНАЯ ФУНКЦИЯ, И ОНА БУДЕТ РАССМОТРЕНА В КОНЦЕ ГЛАВЫ.

ОПЕРАТОРЫ: INVERSE И OVER НЕ УПРАВЛЯЮТ АТТРИБУТАМИ, НО ТЕМ НЕ МЕНЕЕ ОПРЕДЕЛЯЮТ СПОСОБ ВЫВОДА НА ЭКРАН. В ЭТИХ ОПЕРАТОРАХ ИСПОЛЬЗУЕТСЯ ЗНАЧЕНИЯ ПАРАМЕТРОВ '2' И '1'. ЕСЛИ ВЫ ДАДИТЕ: INVERSE 1, ТО ВЫВОДИМЫЙ СИМВОЛ ИЗМЕНИТ СВОЮ ОБЫЧНУЮ ФОРМУ (ВЫВОД БУДЕТ ОСУЩЕСТВЛЯТЬСЯ В НЕГАТИВНОМ ИЗООБРАЖЕНИИ). В ОБЫЧНОМ ВИДЕ МЫ ПИШЕМ ЧЕРНЫМ ПО БЕЛОМУ, В ИНВЕРСНОМ - БЕЛЫМ ПО ЧЕРНУ.

ОПЕРАТОР : OVER 1 - УСТАНОВЛИВАЕТ РЕЖИМ РАСШИРЕННОГО ВЫВОДА, В ОБЫЧНОМ РЕЖИМЕ ПРИ ВЫВОДЕ СИМВОЛА НА ЗНАКОМЕСТО, ТАМ СТИРАЕТСЯ ВСЕ ВЫВЕДЕННОЕ РАНЕЕ, ПРИ РАСШИРЕННОМ ВЫВОДЕ МОЖНО НАКЛАДЫВАТЬ СИМВОЛЫ ДРУГ НА ДРУГА, ЭТО ПОЗВОЛЯЕТ ВЫВОДИТЬ СОСТАВНЫЕ СИМВОЛЫ, НАПРИМЕР СТИЛИЗОВАННЫЕ ТЕКСТЫ.

ПРОГРАММА ДЛЯ ВЫВОДА ГОТИЧЕСКОГО ШРИФТА:

```
10 OVER 1
20 FOR N=1 TO 32
30 PRINT "O" CHR*8 " "
40 NEXT N
```

ОПРЕДЕЛИМ, ЧТО УПРАВЛЯЮЩИЙ СИМВОЛ CHR*8 ВОЗВРАЩАЕТ НА ОДНУ ПОЗИЦИЮ.

ВОЗМОЖЕН ЕЩЕ ОДИН СПОСОБ ИСПОЛЬЗОВАНИЯ INK И PAPER, ИХ МОЖНО ВВОДИТЬ КАК ПАРАМЕТРЫ PRINT. ТОЧНО ТАК ЖЕ МОЖНО ИСПОЛЬЗОВАТЬ И ДРУГИЕ ОПЕРАТОРЫ, РАССМОТРЕННЫЕ В ЭТОЙ ГЛАВЕ, ОТМЕТИВ ПРИ ЭТОМ, ЧТО ИХ ДЕЙСТВИЕ РАСПРОСТРАНЯЕТСЯ ТОЛЬКО ДО КОНЦА PRINT.

В ПРИМЕРЕ: PRINT PAPER 8 "X" ; PRINT"Y" ТОЛЬКО "X" БУ-
ДЕТ ВЫВЕДЕН НА ЖЕЛТЫЙ ФОН.

(K),(L),(G)			(G)		(E)		
НЕТ	SYMBOL	CAPS	НЕТ	ЛЮБОЙ	НЕТ	CAPS	SYMB
1	1	EDIT			ФОН ГОЛУБОЙ	ТОН ГОЛУБОЙ	DEF FN
2	@	CAPS LOCK			ФОН КРАСНЫЙ	ТОН КРАСНЫЙ	FN
3	*	TRUE VIDEO			ФОН ФИОЛЕТО ВЫЙ	ТОН ФИОЛЕТО ВЫЙ	LINE
4	x . .	INVEN VIDEO			ФОН ЗЕЛЕНый	ТОН ЗЕЛЕНый	OPEN
5	%	КУРСОР ВЛЕВО			ФОН СИНИЙ	ТОН СИНИЙ	CLOSE
6	&	КУРСОР ВНИЗ			ФОН ЖЕЛТЫЙ	ТОН ЖЕЛТЫЙ	MOVE
7	,	КУРСОР ВВЕРХ			ФОН БЕЛЫЙ	ТОН БЕЛЫЙ	ERASE
8	(	КУРСОР ВПРАВО			НОРМАЛ, ЯРКОСТЬ	НОРМАЛ, МЕРЦАН.	POINT
9	)	ГРАФИЧ РЕЖИМ	ГРАФИЧЕСК ВЫХОД	ГРАФИЧЕСК ВЫХОД	ПОВЫШ. ЯРКОСТЬ	С МЕРЦАН.	CAT
10	-	DELETE	DELETE	DELETE	ФОН ЧЕРНЫЙ	ЗАКРАШ. ЧЕРНЫЙ	FORMAT

INK И ДРУГИЕ ОПЕРАТОРЫ НЕ ДЕЙСТВУЮТ В НИЖНЕЙ ЧАСТИ ЭКРАНА, ПРЕД-
НАЗНАЧЕННОЙ ДЛЯ ВВОДА КОМАНД И INPUT-ДАННЫХ, ДЛЯ ИЗМЕНЕНИЯ ЦВЕ-
ТОВ В ЭТОЙ ЧАСТИ ЭКРАНА СЛУЖИТ ОПЕРАТОР: BORDER "ЦВЕТ".

КОДИРОВКА ЦВЕТОВ ПРЕЖНЯЯ, ВОЗМОЖНЫ МЕРЦАНИЯ И ПОВЫШЕННАЯ ЯРКОСТЬ ДЛЯ ЭТОГО ИСПОЛЬЗУЙТЕ СООТВЕТСТВУЮЩИЕ ПАРАМЕТРЫ В INPUT (НАПОДОБИЕ PRINT), ЭТИ ПАРАМЕТРЫ ДЕЙСТВУЮТ ДО КОНЦА ОПЕРАТОРА ИЛИ ДО ТЕХ ПОР ПОКА ЗАПРАШИВАЕМЫЕ ДАННЫЕ НЕ БУДУТ ВВЕДЕНЫ. ВЫПОЛНИМ INPUT FLASH 1 INK 1 "WHAT IS YOUR NUMBER ?" И

ВОЗМОЖНО ИЗМЕНЕНИЕ ЦВЕТОВ И С ПОМОЩЬЮ УПРАВЛЯЮЩИХ СИМВОЛОВ, ПОДОБНЫХ УПРАВЛЯЮЩИМ СИМВОЛАМ ДЛЯ AT И TAB В ГЛАВЕ 15.

CHR*16 СООТВЕТСТВУЕТ INK
CHR*17 ----//---- PAPER
CHR*18 ----//---- FLASH
CHR*19 ----//---- BRIGHT
CHR*20 ----//---- INVERSE
CHR*21 ----//---- OVER

ТАК PRINT CHR*16+CHR*9 ТО ЖЕ, ЧТО И PRINT INK 9 МОЖНО ПОЛЬЗОВАТЬСЯ ЛИБО УПРАВЛЯЮЩИМИ СИМВОЛАМИ, ЛИБО ОПЕРАТОРАМИ ИХ МОЖНО СТАВИТЬ КАК ПОСЛЕ НОМЕРА СТРОКИ, ТАК И В КОНЦЕ СТРОКИ. ДЛЯ УДОБСТВА МОЖНО ПОЛЬЗОВАТЬСЯ В РАСШИРЕННОМ РЕЖИМЕ (E) ЦИФРАМИ ЦИФРЫ ОТ 0 ДО 7 УСТАНОВЛИВАЮТ ЦВЕТ INK, ЕСЛИ CAPS SHIFT НАЖАТА, И ЦВЕТ PAPER, ЕСЛИ НЕ НАЖАТА. ЕСЛИ НАЖАТЬ ЦИФРУ В E-РЕЖИМЕ, ТО БУДУТ ВЫВЕДЕНЫ CHR*17 И CHR* КОД ЦВЕТА, ЕСЛИ В ЭТО ВРЕМЯ БЫЛА НАЖАТА CAPS SHIFT, ТО БУДУТ ВЫВЕДЕНЫ: CHR*16 И CHR* КОД ЦВЕТА. ЕСЛИ ВЫ ЗАХОТИТЕ УНИЧТОЖИТЬ ВВОДИМОЕ, НАЖМИТЕ DELETE ДВА РАЗА, ПОСЛЕ ПЕРВОГО РАЗА НА ЭКРАНЕ ВЫСВЕТИТСЯ ЗНАК ВОПРОСА, ИЛИ ЧТО-НИБУДЬ ЕЩЕ, НЕ ПУГАЙТЕСЬ И НАЖМИТЕ DELETE ЕЩЕ РАЗ. УПРАВЛЕНИЕ КУРСОРОМ ТОЖЕ НЕ РАБОТАЕТ ОБЫЧНЫМ СПОСОБОМ ДО ТЕХ ПОР ПОКА КУРСОР НЕ ВЫИДЕТ К ПРЕДЫДУЩЕМУ УПРАВЛЯЮЩЕМУ СИМВОЛУ.

ДЕЙСТВИЕ В РАСШИРЕННОМ РЕЖИМЕ (E)

8 ДАЕТ CHR*19 И CHR*0 -НОРМАЛЬНАЯ ЯРКОСТЬ

9 ДАЕТ CHR*19 И CHR*1 -ПОВЫШЕННАЯ ЯРКОСТЬ

CAPS SHIFT С 8 ДАЕТ CHR*18 И CHR*0 -НЕ МЕРЦАЮЩЕЕ

CAPS SHIFT С 9 ДАЕТ CHR*18 И CHR*1 -МЕРЦАЮЩЕЕ

В (L)- РЕЖИМЕ;

CAPS SHIFT С 3 ДАЕТ CHR*20 И CHR*0- ОБЫЧНЫЙ ВЫВОД

CAPS SHIFT С 4 ДАЕТ CHR*20 И CHR*1- ИНВЕРСНЫЙ (НЕГАТИВНЫЙ) ВЫВОД

ФУНКЦИЯ ATTR ИМЕЕТ СЛЕДУЮЩИЙ ФОРМАТ:

ATTR ("СТРОКА", " СТОЛБЕЦ")

ЗНАЧЕНИЯ ДВУХ ПАРАМЕТРОВ ФУНКЦИИ ПОДОБНО ЗНАЧЕНИЮ ПАРАМЕТРОВ В AT, В РЕЗУЛЬТАТЕ ВЫПОЛНЕНИЯ БУДУТ ВЫВЕДЕНЫ ЗНАЧЕНИЯ АТТРИБУТОВ ДЛЯ СООТВЕТСТВУЮЩЕЙ ПОЗИЦИИ ЭКРАНА. ВЫВОДИМЫЙ РЕЗУЛЬТАТ ЭТО ЧИСЛО, ПРЕДСТАВЛЯЮЩЕЕ СУММУ ЧЕТЫРЕХ ЧИСЕЛ:

1) 128-ЕСЛИ ЗНАКОМЕСТО МЕРЦАЮЩЕЕ, 0-ЕСЛИ ОБЫЧНОЕ

2) 64-ЕСЛИ ПОВЫШЕННАЯ ЯРКОСТЬ, 0-ЕСЛИ ОБЫЧНАЯ

3) 8* КОД ЦВЕТА ФОНА

4) КОД ЦВЕТА ТОНА

ПРИМЕР: ЗНАКОМЕСТО МЕРЦАЮЩЕЕ, ОБЫЧНОЙ ЯРКОСТИ, ЖЕЛТЫЙ ФОН, СИНИЙ ТОН $128+0+(8*6)+1=177$

ПРОВЕРИМ ЭТО, ВЫПОЛНИВ:

PRINT AT 0,0 FLASH 1 PAPER 6 INK 1 " " ATTR(0,0)

УПРАЖНЕНИЯ:

1) PRINT "В" CHR*8 OVER 1 "/"

ЗДЕСЬ "/" ПЕРЕЧЕРКИВАЕТ "В". ЭТИМ СПОСОБОМ МОЖНО ВЫВОДИТЬ СЛОВА КОМБИНИРОВАННЫХ ЗНАКОВ НА ZX СПЕКТРУМ: ДВА ФОНА ИЛИ ДВА ТОНА ДАЮТ ФОН, ОДИН ИЗ НИХ ДАЕТ ТОН, ЭТО ИНТЕРЕСНОЕ СВОЙСТВО. ЕСЛИ ВЫ ПОВТОРИТЕ ВЫВОД ОДНОГО СИМВОЛА ДВАЖДЫ, ТО ОН НЕ БУДЕТ ОТОБРАЖЕН. ТАК ЕСЛИ ДАТЬ: PRINT CHR*8 OVER 1 "/"

ДОПОЛНИТЕЛЬНО К ОПИСАННОМУ ВЫШЕ УТВЕРЖДЕНИЮ, ТО МЫ В РЕЗУЛЬТАТЕ УВИДИМ "В" НЕ ПЕРЕЧЕРКНУТОЕ "/", ТАК ЛИ ЭТО?

2) ВЫПОЛНИМ: PAPER 0 INK 0

ДЕЙСТВУЮТ ЛИ ЭТИ ОПЕРАТОРЫ В НИЖНЕЙ ЧАСТИ ЭКРАНА?

ЧТО МЫ УВИДИМ, ЕСЛИ ДОБАВИТЬ BORDER 07

3) ВЫПОЛНИМ ПРОГРАММУ:

10 POKE 22527+RND*704, RND*127

20 GO TO 10

РЕЗУЛЬТАТОМ ПРОГРАММЫ ЯВИТСЯ СМЕНА ЦВЕТОВ ЗНАКОМЕСТ, РАСПРЕДЕЛЕННЫХ ПО ЭКРАНУ СЛУЧАЙНЫМ ОБРАЗОМ. ВОЗМОЖНО ВЫ УВИДИТЕ СМЕНУ ЦВЕТОВ НА ДИАГОНАЛЬНЫХ СТУПЕНЯХ. ЭТО ЯВЛЯЕТСЯ СЛЕДСТВИЕМ ТОГО, ЧТО МЫ ПОЛЬЗУЕМСЯ КВАЗИСЛУЧАЙНЫМ РАСПРЕДЕЛЕНИЕМ, КОТОРОЕ ЛИШЬ ПРИБЛИЖЕННО ВОСПРОИЗВОДИТ СЛУЧАЙНОЕ РАСПРЕДЕЛЕНИЕ.

4) ВВЕД М ВРУЧНУЮ, ИЛИ С ПОМОЩЬЮ ОПЕРАТОРА LOAD ПРОГРАММУ ДЛЯ ВВОДА ШАХМАТНЫХ ФИГУР ИЗ ГЛАВЫ 14, И ВВЕДЕМ ПРОГРАММУ РАЗБИВКИ ЭКРАНА ПОД ШАХМАТНУЮ ДОСКУ:

```

5 REM DRAW BLANKBOARD
10 LET BB=1 LET BW=2 REM RED AND BLUE FOR BOARD
15 PAPER BW:INK BB:CLS
20 PLOT 79, 128: REM BORDER
30 DRAW 65,0 DRAW 0,-65
40 DRAW-65,0: DRAW 0,65
50 PAPER BB
60 REM BOARD
70 FOR N=0 TO 3: FOR M=0 TO 3
80 PRINT AT 6+2*N,11+2*M " "
90 PRINT AT 7+2*N,10+2*M " "
100 NEXT M: NEXT N
110 PAPER 8
120 LET PW=6:LET PB=5:REM COLOURS OF WHITE AND BLACK
    PIECES
200 DIM B*(8,8):REM POSITIONS OF PIECES
205 REM SET UP INITIAL POSITION
210 LET B*(1)="RNBQKBNR":REM LITTLE LINE 240
220 LET B*(2)="PPPPPPPP":REM LITTLE LINE 230
230 LET B*(7)="PPPPPPPP":REM BIG LINE 220
240 LET B*(8)="RNBQKBNR":REMBIG LINE 210
300 REM DISPLAY BOARD
310 FOR N=1 TO 8:FOR M=1 TO 8
320 LET BC CODE " " THEN GO TO 350:REM SPACE
330 IF BC>CODE "Z" THEN INK PB:LET BC-32:REM LO
    CASE FOR BLACK
340 LET BC=BC+79:REM CONVERT TO GRAPHICS
350 PRINT AT 5+N,9+M CHR*BC
360 NEXT M:NEXT N
400 PAPER 7: INK 0
    
```

ГЛАВА 17

ГРАФИКА

=====

КРАТКОЕ СОДЕРЖАНИЕ: PLOT, DRAW, CIRCLE, POINT

ЭТА ГЛАВА ОПИСЫВАЕТ ВОЗМОЖНОСТИ КОМПЬЮТЕРА ZX SPECTRUM ПО ОТОБРАЖЕНИЮ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ. ЭКРАН КОМПЬЮТЕРА СОДЕРЖИТ 22 СТРОКИ ПО 32 СИМВОЛА В КАЖДОЙ, ЧТО СОСТАВЛЯЕТ 22X32=704 СИМВОЛЬНЫЕ ПОЗИЦИИ, КАК ВЫ УЖЕ ПОНЯЛИ ИЗ 16 ГЛАВЫ, КАЖДАЯ СИМВОЛЬНАЯ ПОЗИЦИЯ ПРЕДСТАВЛЯЕТСЯ КВАДРАТОМ 8X8 ТОЧЕК, НАЗЫВАЕМЫХ ПИКСЕЛЯМИ.

ПИКСЕЛЬ ЗАДАЕТСЯ ДВУМ ЧИСЛАМИ, ЕГО КООРДИНАТАМИ, ПЕРВОЕ ЗАДАЕТ КООРДИНАТУ X, Т.Е. УДАЛЕНИЕ (В ПИКСЕЛЯХ) ДО ЛЕВОЙ ГРАНИЦЫ ЭКРАНА), ВТОРОЕ ЗАДАЕТ КООРДИНАТУ Y, УДАЛЕНИЕ ОТ НИЖНЕЙ ГРАНИЦЫ ЭКРАНА. КООРДИНАТЫ ЗАПИСЫВАЮТСЯ В СКОБКАХ, ТАК (0,0), (255,0), (0,175) И (255,175) ЗАДАЕТ СООТВЕТСТВЕННО НИЖНИЙ ЛЕВЫЙ, НИЖНИЙ ПРАВЫЙ, ВЕРХНИЙ ПРАВЫЙ И ВЕРХНИЙ ЛЕВЫЙ УГЛЫ ЭКРАНА.

ОПЕРАТОР PLOT X,Y ВЫЗЫВАЕТ ВЫСВЕЧИВАНИЕ ЗАКРАШИВАЮЩИМ ЦВЕТОМ (INK) ПИКСЕЛЯ С УКАЗАННЫМИ КООРДИНАТАМИ.

НАПРИМЕР ПРОГРАММА: 10 PLOT INT(RND*256),INT(RND*176)
20 INPUT A*
30 GOTO 10

БУДЕТ ВЫСВЕЧИВАТЬ НЕКОТОРЫЙ СЛУЧАЙНЫЙ ПИКсель ПРИ КАЖДОМ НАЖАТИИ ENTER.

ЕСТЬ И БОЛЕЕ ИНТЕРЕСНЫЕ ПРОГРАММЫ, НАПРИМЕР СЛЕДУЮЩАЯ ПРОГРАММА ВЫЧЕРЧИВАЕТ ГРАФИК ФУНКЦИИ SIN X, ДЛЯ X В ИНТЕРВАЛЕ ОТ 0 ДО 2π : 10 FOR N=0 TO 255

20 PLOT N,86+60*SIN(N/128*PI)
30 NEXT N

ИЛИ ПРОГРАММА:

10 FOR N=0 TO 255
20 PLOT N,80*SQR(N/64)
30 NEXT N

КОТОРАЯ ЧЕРТИТ ГРАФИК SQRX (ЧАСТЬ ПАРАБОЛЫ) В ИНТЕРВАЛЕ ОТ 0 ДО 4. ПОМНИТЕ, ЧТО КООРДИНАТЫ ПИКСЕЛЯ ОТЛИЧАЮТСЯ ОТ АДРЕСАЦИИ СТРОК И ПОЗИЦИИ В ПОДКОМАНДЕ AT.

ПОЛЬЗУЙТЕСЬ ДИАГРАММОЙ ИЗ ГЛАВЫ 15.

ПОМОЩЬ ПРИ ПОСТРОЕНИИ ИЗОБРАЖЕНИЙ ВАМ МОГУТ ОКАЗАТЬ ОПЕРАТОРЫ DRAW И CIRCLE.

ОПЕРАТОР DRAW ЧЕРТИТ ЛИНИЮ, ЗАДАННУЮ В ФОРМЕ: DRAW X,Y НАЧАЛОМ ЛИНИИ ЯВЛЯЕТСЯ ПИКсель, НА КОТОРОМ ЗАВЕРШИЛСЯ ОДИН ИЗ ПРЕДЫДУЩИХ ОПЕРАТОРОВ PLOT, DRAW ИЛИ CIRCLE (ЭТОТ ПИКсель НАЗЫВАЕТСЯ ТЕКУЩЕЙ PLOT- ПОЗИЦИЕЙ. ОПЕРАТОРЫ RUN, CLEAR, CLS, И NEW УСТАНОВЛИВАЕТ ЕЕ В ЛЕВЫЙ НИЖНИЙ УГОЛ ЭКРАНА). ТАКИМ ОБРАЗОМ ОПЕРАТОР DRAW ЗАДАЕТ ДЛИНУ И НАПРАВЛЕНИЕ ВЫЧЕРЧИВАНИЯ ЛИНИИ, НО НЕ ЕЕ НАЧАЛЬНУЮ ТОЧКУ.

ПОЭКСПЕРИМЕНТИРУЙТЕ С ТАКИМИ КОМАНДАМИ:

PLOT 0,100: DRAW 80,-35
PLOT 90,150: DRAW 80,-35

ЧЕРТИТЬ МОЖНО ТАКЖЕ В ЦВЕТЕ, НО ПРИ ЭТОМ НАДО ИМЕТЬ ВВИДУ, ЧТО ЦВЕТ УСТАНОВЛИВАЕТСЯ ДЛЯ ЦЕЛОЙ СИМВОЛЬНОЙ ПОЗИЦИИ И НЕ МОЖЕТ БЫТЬ ЗАДАН ДЛЯ ОТДЕЛЬНОГО ПИКСЕЛЯ. СЛЕДУЮЩАЯ ПРОГРАММА ДЕМОНИСТРИРУЕТ ЭТО: 10 BORDER 0; PAPER 0; INK 7; CLS; REM BLACK OUT SCRKKK

20 LET X1=0: LET Y1=0: REM START OF LINE
30 LET C=1: REM FOR INK COLOUR, STARTING BLUE
40 LET X2=INT(RND*256): LET Y2=INT(RND*176): REM RANDOM
FINISH OF LINE
50 DRAW INK C: X2-X1, Y2-Y1
60 LET X1=X2: LET Y1=Y2: REM NEXT LINE STARTS WHERE LINE
ONE FINISHED
70 LET C=C+1: IF C=8 THEN LET C=1: REM NEW COLOUR
80 GOTO 40

ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ В ОПЕРАТОРАХ PLOT И DRAW УПРАВЛЯЮЩИЕ СИМВОЛЫ PAPER, INK, FLASH, BRIGHT, INVERSE И OVER ТАК ЖЕ, КАК И В ОПЕРАТОРАХ PRINT И INPUT. УПРАВЛЯЮЩИЕ СИМВОЛЫ ЗАПИСЫВАЮТСЯ МЕЖДУ КЛЮЧЕВЫМ СЛОВОМ И КООРДИНАТАМИ И ОКОНЧИВАЮТСЯ ЗАПЯТОЙ ИЛИ ТОЧКОЙ С ЗАПЯТОЙ (СМОТРИ СТРОКУ 50).

ПРИ ПОМОЩИ DRAW МОЖНО ТАКЖЕ ВЫЧЕРТИТЬ ОТРЕЗОК ДУГИ, ИСПОЛЬЗУЯ ДЛЯ ЭТОГО ДОПОЛНИТЕЛЬНОЕ ЧИСЛО (В РАДИАНАХ) ЭТОЙ ДУГИ: DRAW X,Y,A

ЕСЛИ 'A' ПОЛОЖИТЕЛЬНО, ТО ДУГА ВЫЧЕРЧИВАЕТСЯ ВЛЕВО, А ЕСЛИ ОТРИЦАТЕЛЬНО, ТО ВПРАВО. ПРИ 'A' РАВНОМ 2π ВЫЧЕРЧИВАЕТСЯ ПОЛНАЯ ОКРУЖНОСТЬ, НАПРИМЕР:

10 PLOT 100,100: DRAW 50,50,PI

ВЫЧЕРТИТ ПОЛУОКРУЖНОСТЬ С НАЧАЛЬНОЙ ТОЧКОЙ (100,100) И КОНЕЧНОЙ ТОЧКОЙ (150,150). ВЫЧЕРЧИВАНИЕ НАЧНЕТСЯ В НАПРАВЛЕНИИ ЮГО-ВОСТОК

А ЗАКОНЧИТСЯ В НАПРАВЛЕНИИ СЕВЕРО-ЗАПАД.

ОПЕРАТОР CIRCLE ВЫЧЕРЧИВАЕТ ПОЛНЫЙ КРУГ ЗАДАВАЕМЫЙ КООРДИНАТАМИ ЕГО ЦЕНТРА И РАДИУСОМ:

CIRCLE X,Y, РАДИУС

КАК И В ОПЕРАТОРАХ PLOT И DRAW ВЫ МОЖЕТЕ УКАЗАТЬ В ЭТОМ ОПЕРАТОРЕ РАЗЛИЧНЫЕ ЦВЕТА.

ФУНКЦИЯ POINT ВОЗВРАЩАЕТ ХАРАКТЕРИСТИКИ ЦВЕТА ЗАДАННОГО ПИКСЕЛЯ, НАПРИМЕР СТРОКА ПРОГРАММЫ

CLS:PRINT POINT(0,0):PLOT (0,0): PRINT POINT(0,0/
ВЫВЕДЕТ: PAPER 7:INK0

ДОПУСКАЕТСЯ ТАКЖЕ ЗАДАВАТЬ УПРАВЛЯЮЩИЕ СИМВОЛЫ INVERSE И В ОПЕРАТОРЕ PLOT, ПО УМОЛЧАНИЮ ПРЕДПОЛАГАЮТСЯ РАВНЫМИ 0 (ОТКЛЮЧЕНО),

НО ВЫ МОЖЕТЕ ЗАДАТЬ И 1, ПРИ ЭТОМ:

PLOT INVERSE 1 - УСТАНАВЛИВАЕТ ДЛЯ ЗАДАННОГО ПИКСЕЛЯ ЦВЕТ ФОНА
PLOT OVER 1 - ИЗМЕНЯЕТ ЦВЕТ ПИКСЕЛЯ НА ПРОТИВОПОЛОЖНЫЙ, ЕСЛИ

БЫЛ ЦВЕТ ТОНА, ТО СТАНОВИТСЯ ЦВЕТ ФОНА И НАОБОРОТ.

PLOT INVERSE 1

OVER 1 - СОХРАНЯЕТ ЦВЕТ ПИКСЕЛЯ БЕЗ ИЗМЕНЕНИИ, НО МЕНЯЕТ ТЕКУЩУЮ PLOT ПОЗИЦИЮ.

ДРУГОЙ ПРИМЕР ИСПОЛЬЗОВАНИЯ OVER С ЗАПИСЬЮ ЧЕРНЫМ ПО БЕЛОМУ PLOT 0,0: DRAW OVER 1 255,175 ВЫЧЕРЧИВАЕТ ЛИНИЮ ПО ДИАГОНАЛИ. ТЕПЕРЬ ПОПРОБУЙТЕ:

PLOT 0,0 DRAW INVERSE 1 256,175
И ПЕРЕЧЕРТИТЕ ЕЕ КОМАНДОЙ

DRAW OVER 1 -250,-175

ЭТО НЕ ИЗМЕНИТ КАРТИНКУ, ТАК КАК ПРИ ЧЕРЧЕНИИ КАК ВПЕРЕД ТАК И НАЗАД ИСПОЛЬЗУЮТСЯ ОДНИ И ТЕ ЖЕ ПИКСЕЛИ.

ИМЕЕТСЯ СПОСОБ ПОЛУЧЕНИЯ НЕОБЫЧНЫХ ЦВЕТОВ В ОДНОМ КВАДРАТЕ С ИСПОЛЬЗОВАНИЕМ ОПРЕДЕЛЯЕМЫХ ПОЛЬЗОВАТЕЛЕМ СИМВОЛОВ. ВЫПОЛНИТЕ ЭТУ ПРОГРАММУ:

```
1000 FOR N=0 TO 6 STEP 2
1010 POKE USR "'A'+N, BIN 01010101:
POKE SR "'A'+N+1, BIN 10101010
1020 NEXT N
```

ОНА ЗАДАЕТ ОПРЕДЕЛЯЕМЫЙ ПОЛЬЗОВАТЕЛЕМ СИМВОЛ ДЛЯ ШАХМАТНОЙ ДОСКИ КОТОРЫЙ ЗАКРЕПЛЯЕТСЯ ЗА КЛАВИШЕЙ 'A', ДЛЯ СИМВОЛА ИСПОЛЬЗУЕТСЯ КРАСНЫЙ ЗАКРАШИВАЮЩИЙ ЦВЕТ И ЖЕЛТЫЙ ЦВЕТ ФОНА, НО НА ЭКРАНЕ ЭТОТ СИМВОЛ БУДЕТ КАЗАТЬСЯ ОРАНЖЕВЫМ.

ЕЩЕ ОДИН ПРИМЕР, ПРОГРАММА, КОТОРАЯ СТРОИТ ГРАФИК НЕКОТОРОЙ ФУНКЦИИ. НА ПЕРВЫЙ ЕЕ ЗАПРОС ВЫ ОТВЕЧАЕТЕ ЧИСЛОМ 'N' ЗАДАЮЩИМ ОБЛАСТЬ ЗНАЧЕНИЯ АРГУМЕНТА (Т.Е. ГРАФИК БУДЕТ СТРОИТЬСЯ ДЛЯ ЗНАЧЕНИИ АРГУМЕНТА В ДИАПАЗОНЕ ОТ -N ДО +N). ВТОРОЙ ОТВЕТ - ЭТО ВЫРАЖЕНИЕ В ВИДЕ СИМВОЛЬНОЙ СТРОКИ, ЗАДАЮЩЕЙ ФУНКЦИЮ, ИСПОЛЬЗУЕМУЮ 'X' В КАЧЕСТВЕ АРГУМЕНТА:

```
10 PLOT 0,87:DRAW 255,0
20 PLOT 127,0:DRAW 0,175
30 INPUT S,BX
40 FOR F=2 TO 255
50 LET X=(F-128)*S/123:LET Y=VAL EX
60 IF ABS Y>87 THEN LET T=0: GOTO 100
70 IF NOT T THEN PLOT F,Y+88:LET T=1:GOTO 100
80 DRAW 1,Y-OLDY
100 LET OLDY=INT(Y+5)
110 NEXT F
```

ВЫПОЛНИТЕ, ВВЕДЯ 10 ДЛЯ ЧИСЛА 'N' И '10*TAN X' ДЛЯ ФУНКЦИИ. БУДЕТ ВЫЧЕРЧЕН ГРАФИК ФУНКЦИИ TG X ПРИ X ИЗМЕНЯЮЩЕМСЯ ОТ -10 ДО +10.

=====

КРАТКОЕ СОДЕРЖАНИЕ: PAUSE, INKEY*, PEEK

ЕСЛИ ВЫ РЕШИЛИ ЗАДЕРЖАТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ НА НЕКОТОРОЕ ВРЕМЯ, ТО ВАМ СЛЕДУЕТ ИСПОЛЬЗОВАТЬ ОПЕРАТОР PAUSE N, КОТОРЫЙ ОСТАНАВЛИВАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ И ОТОБРАЖАЕТ КАРТИНУ В ТЕЧЕНИИ 'N' ТЕЛЕВИЗИОННЫХ КАДРОВ (50 КАДРОВ В ЕВРОПЕ ИЛИ 60 КАДРОВ В АМЕРИКЕ). 'N' МОЖЕТ БЫТЬ ВПЛОТЬ ДО 65535, ЧТО СОСТАВЛЯЕТ 22 МИНУТЫ, ЕСЛИ N=0, ТО ЭТО ОЗНАЧАЕТ, ЧТО ОПЕРАТОР PAUSE НЕ ИМЕЕТ ОГРАНИЧЕНИЯ ПО ВРЕМЕНИ.

ВЫПОЛНЕНИЕ ПРОГРАММЫ ВСЕГДА МОЖЕТ БЫТЬ ВОЗОБНОВЛЕНО ДО ОКОНЧАНИЯ ВРЕМЕНИ, ОПРЕДЕЛЕННОГО В ОПЕРАТОРЕ PAUSE, НАЖАТИЕМ ЛЮБОЙ КЛАВИШИ (НАДО ПОМНИТЬ, ЧТО CAPS SHIFT БУДЕТ ВЫЗЫВАТЬ ПРЕРВАНИЕ).

ПРИМЕР ПРОГРАММЫ МОДЕЛИРОВАНИЯ СЕКУНДНОЙ СТРЕЛКИ ЧАСОВ:

```
10 REM FIRST WE DRAW THE CLOCK FACE
20 FOR N = 1 TO 12
30 PRINT AT 10-10*COS(N/6*PI),16+10*SIN(N/6*PI) N
40 NEXT N
50 REM NOW WE START THE CLOCK
60 FOR T=0 TO 200000:REM THIS THE TIME IN SECONDS
70 LET A=T/30*PI:REM A IS THE ANGLE OF THE SECOND
  HAND IN RADIANS
80 LET SX=80*SIN A:LET SY=80*COS A
200 PLOT 128,88:DRAW OVER 1;SX,SY:REM DRAW SECOND HAND
210 PAUSE 42
220 PLOT 128,88:DRAW OVER 1 SX,SY:REM ERASE SECOND HAND
400 NEXT T
```

ЭТИ ЧАСЫ ОСТАНОВЯТСЯ, ПРОРАБОТАВ ПРИБЛИЗИТЕЛЬНО 55,5 ЧАСОВ, ЧТО ЗАДАЕТСЯ В ОПЕРАТОРЕ С НОМЕРОМ 60. ОПЕРАТОР 210 ПРОИЗВОДИТ ОТСЧЕТ ВРЕМЕНИ. КАЗАЛОСЬ БЫ ЗДЕСЬ ДОЛЖЕН БЫТЬ ОПЕРАТОР PAUSE 50 (ЕВРОПА), ДЛЯ ТОЧНОГО ОТСЧЕТА ОДНОЙ СЕКУНДЫ, НО ТОГДА БЫ МЫ НЕ УЧЛИ ВРЕМЯ, ЗАТРАЧЕННОЕ НА ВЫПОЛНЕНИЕ ОСТАЛЬНЫХ ОПЕРАТОРОВ ПРОГРАММЫ. РАССМАТРИВАЕМЫЙ ВАРИАНТ ЧАСОВ ОБЕСПЕЧИВАЕТ 2%-Ю ТОЧНОСТЬ ИЛИ ИНЫМИ СЛОВАМИ УХОД НА ПОЛЧАСА В ДЕНЬ.

ВОЗМОЖНЫ И БОЛЕЕ ТОЧНЫЕ СПОСОБЫ ИЗМЕРЕНИЯ ВРЕМЕНИ. ДЛЯ ЭТОГО МОЖНО ИСПОЛЬЗОВАТЬ СОДЕРЖИМОЕ СПЕЦИАЛЬНЫХ ОБЛАСТЕЙ ПАМЯТИ, В ЭТОМ СЛУЧАЕ ДАННЫЕ ИЗ ПАМЯТИ МОГУТ БЫТЬ ВЫЗВАНЫ С ПОМОЩЬЮ ФУНКЦИИ PEEK. ПОДРОБНО ЭТО РАССМОТРЕНО В ГЛАВЕ 25, ЗДЕСЬ ЖЕ В КАЧЕСТВЕ ПРИМЕРА РАССМОТРИМ ВЫРАЖЕНИЕ:

$(65536 * \text{PEEK } 23674 + 256 * \text{PEEK } 23673 + \text{PEEK } 23672) / 50$

ОНО ДАЕТ КОЛИЧЕСТВО СЕКУНД, ПРОШЕЛШИХ С ТЕХ ПОР, КАК КОМПЬЮТЕР БЫЛ ВКЛЮЧЕН (ВПЛОТЬ ДО 3-Х СУТОК И 21-ГО ЧАСА). НИЖЕ ПРИВОДИТСЯ МОДИФИЦИРОВАННАЯ ПРОГРАММА МОДЕЛИРОВАНИЯ ЧАСОВ:

```
10 REM FIRST WE DRAW THE CLOCK FACE
20 FOR X=1 TO 12
30 PRINT AT 10-10*COS(N/6*PI),16+10*SIN(N/6*PI) N
40 NEXT N
50 DEF FN T()=INT((65536*PEEK 23674+256*PEEK 23673+PEEK
  23672)/50):REM NUMBER OF SEKOND SINCE START
100 REM NOW WE START THE CLOCK
110 LET T1=FN T()
120 LET A=T1/30*PI:REM A IS THE ANGLE OF THE SECOND HAND
  IN RADIANS
130 LET SX=72*SIN A:LET SY=72*COS A
140 PLOT 131,91:DRAW OVER 1; SX,SY:REM DRAW HAND
200 LET T=FN T()
210 IF T<=T1 THEN GOTO 200:REM WAIT UNTIL TIME FOR NEXT
  HAND
```

```
220 PLOT 131,91: DRAW OVER 1: SX,SY: REM RUB OUT OLD HAND
230 LET T1=T: GOTO 120
```

ЭТИ ЧАСЫ ОБЕСПЕЧИВАЮТ ТОЧНОСТЬ 0,001% ИЛИ УХОД НА 10 СЕКУНД В ДЕНЬ, ОДНАКО, ЭТО ВОЗМОЖНО ПРИ УСЛОВИИ, ЧТО ВЫ НЕ ИСПОЛЬЗОВАЛИ ОПЕРАТОР ВЕЕР, ВВОД/ВЫВОД НА МАГНИТОФОН И ПРИНТЕР. ВСЕ ЭТИ ОПЕРАЦИИ УВЕЛИЧИВАЮТ ПОГРЕШНОСТЬ.

ЧИСЛА РЕЕК 23674, РЕЕК 23673 И РЕЕК 23672 ВЫДЕЛЯЮТ АДРЕСА ЯЧЕЕК ПАМЯТИ КОМПЬЮТЕРА И ИСПОЛЗУЕМЫХ ДЛЯ ПОДСЧЕТА 1/50 ДОЛЕЙ СЕКУНДЫ, В КАЖДОЙ ИЗ ЯЧЕЕК ПОДСЧИТЫВАЕТСЯ СУММА ОТ 0 ДО 255. ПОСЛЕ ДОСТИЖЕНИЯ ВЕЛИЧИНЫ 255 В ЛЮБОЙ ИЗ ЯЧЕЕК, ОНА СБРАСЫВАЕТСЯ В 0. ПЕРВОЙ НАЧИНАЕТ ОТСЧИТЫВАТЬ ЯЧЕЙКА РЕЕК 23672. КАЖДЫЕ 1/50 СЕК ЕЕ СОДЕРЖИМОЕ УВЕЛИЧИВАЕТСЯ НА 1. КОГДА В ЯЧЕЙКЕ НАКОПИТСЯ ВЕЛИЧИНА РАВНАЯ 255, ТО ОНА СБРАСЫВАЕТСЯ В 0, А ЗНАЧЕНИЕ ЯЧЕЙКИ РЕЕК 23673 УВЕЛИЧИВАЕТСЯ НА 1. ЧЕРЕЗ КАЖДЫЕ 256/50 СЕК СОДЕРЖИМОЕ ЭТОЙ ЯЧЕЙКИ ПЕРЕХОДИТ ИЗ СОСТОЯНИЯ 255 В 0, А СОДЕРЖИМОЕ ЯЧЕЙКИ РЕЕК 23674 УВЕЛИЧИВАЕТСЯ НА 1.

ПРИ ЗНАЧЕНИЯХ 0 ДЛЯ ЯЧЕЙКИ РЕЕК 23674 И 255 ДЛЯ ЯЧЕЕК РЕЕК 23673 И РЕЕК 23672 (ЭТОТ МОМЕНТ НАСТУПИТ ЧЕРЕЗ 21 МИНУТУ) НАШЕ ВЫРАЖЕНИЕ ПРИМЕТ ЗНАЧЕНИЕ $(65536+256*255+255)/50=1310,7$, НО ЗДЕСЬ ИМЕЕТСЯ СКРЫТАЯ СПАСНОСТЬ, ЧЕРЕЗ СЛЕДУЮЩУЮ 1/50 СЕК ЯЧЕЙКИ БУДУТ СОДЕРЖАТЬ СЛЕДУЮЩИЕ ЗНАЧЕНИЯ: 1, 0, 0.

ПОКА ПРОИЗВОДИТСЯ ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЯ, КОМПЬЮТЕР МОЖЕТ ОЦЕНИТЬ ЗНАЧЕНИЕ ЯЧЕЙКИ РЕЕК 23674 КАК 0 ДО ЗАВЕРШЕНИЯ ЦИКЛИЧЕСКОГО ПЕРЕНОСА, В РЕЗУЛЬТАТЕ ПОЛУЧИМ: $(65536*0+256*0+0)/50=0$, ЧТО БЕЗНАДЕЖНО НЕВЕРНО.

ПРОСТОЕ ПРАВИЛО ПОЗВОЛЯЕТ РЕШИТЬ ЭТУ ПРОБЛЕМУ:

"СЛЕДУЕТ ВЫЧИСЛЯТЬ ВЫРАЖЕНИЕ ДВАЖДЫ В НЕКОТОРОЙ ПОСЛЕДОВАТЕЛЬНОСТИ И ИСПОЛЬЗОВАТЬ СОХРАНЕННЫЙ ОТВЕТ"

ПРИМЕР: 10 DEF FN M(X,Y)=(X+Y+ABS(X-Y))/2: REM THE LANDER OF AND
20 DEF FN U()=(65536*РЕЕК 23674+256*РЕЕК 23673+РЕЕК 23672)/50: REM TIME, MAY BU WRONG

30 DEF FN T()=FN M(FN U(),FN U()): REM TIME RIGHT

ВЫ МОЖЕТЕ ИЗМЕНЯТЬ ЗНАЧЕНИЯ ЧИСЛОВЫХ СЧЕТЧИКОВ ТАК, ЧТОБЫ ПОЛУЧАТЬ РЕАЛЬНОЕ ВРЕМЯ ТОГО МОМЕНТА, КОГДА КОМПЬЮТЕР БЫЛ ВКЛЮЧЕН. НАПРИМЕР, НАДО УСТАНОВИТЬ 10 ЧАСОВ ВЕЧЕРА, ВЫ ПОСЧИТАЛИ ЧТО $12*60*60=180000$ 50-Х ДОЛЕЙ СЕКУНДЫ И ЗНАЧИТ

$180000 = 65536*27+256*119+64$

ДЛЯ ПРИСВОЕНИЯ ТРЕМ ЯЧЕЙКАМ ЗНАЧЕНИЯ 27, 119 И 64 НЕОБХОДИМО ВЫПОЛНИТЬ: РОКЕ 23674,27: РОКЕ 23673,119: РОКЕ 23672,64.

ФУНКЦИЯ INKEY* (БЕЗ АРГУМЕНТОВ) СЧИТЫВАЕТ С КЛАВИАТУРЫ, ЕСЛИ ВЫ НАЖАЛИ НЕКОТОРУЮ КЛАВИШУ (ИЛИ SHIFT И КАКУЮ НИБУДЬ КЛАВИШУ). РЕЗУЛЬТАТОМ БУДЕТ СИМВОЛ, КОТОРЫЙ ИМЕЕТ ЭТА КЛАВИША В РЕЖИМЕ МАРКЕРА <L>, ИЛИ ПУСТАЯ СТРОКА.

ВЫПОЛНИТЕ ПРОГРАММУ КОТОРАЯ ИСПОЛЬЗУЕТ ЭТУ ФУНКЦИЮ:

```
10 IF INKEY* <> " " THEN GOTO 10
20 IF INKEY*=" " THEN GOTO 20
30 PRINT INKEY*
40 GOTO 10
```

ПОМНИТЕ, ЧТО ФУНКЦИЯ INKEY* НЕ БУДЕТ ПОДОБНО INPUT ЖДАТЬ ВАС, ЕСЛИ ВЫ НЕ ВЫПОЛНИТЕ ВВОД ТО СЧИТАЙТЕ, ЧТО ВАШ ШАНС УПУЩЕН.

ГЛАВА 19

ПРОГРАММИРОВАНИЕ ЗВУКОВ

=====

КРАТКОЕ СОДЕРЖАНИЕ: ВЕЕР

ZX SPECTRUM МОЖЕТ ВОСПРОИЗВОДИТЬ ЗВУКИ ПРИ ПОМОЩИ ОПЕРАТОРА ВЕЕР: ВЕЕР: ПРОДОЛЖИТЕЛЬНОСТЬ, ВЫСОТА ЗВУКА
ГДЕ "ПРОДОЛЖИТЕЛЬНОСТЬ" И "ВЫСОТА ЗВУКА" НЕКОТОРЫЕ ЧИСЛОВЫЕ ВЫРАЖЕНИЯ. ПРОДОЛЖИТЕЛЬНОСТЬ ЗАДАЕТСЯ В СЕКУНДАХ, А ВЫСОТА В ПОЛУТОНАХ ОТ ОСНОВНОГО ТОНА "ДО": ПРИ ПОЛОЖИТЕЛЬНЫХ ЧИСЛАХ - ВЫШЕ

НОТЫ 'ДО', А ПРИ ОТРИЦАТЕЛЬНЫХ - НИЖЕ НОТЫ 'ДО'. НА ДИАГРАММЕ ПРИВЕДЕНЫ ВСЕ ЗНАЧЕНИЯ НОТ ОДНОЙ ОКТАВЫ;

	C#	D#		F#	C#	A#						
	DB	EB		CB	AB	BB						
-2	1	3		5	6	10		13	15			
ЛЯ	СИ	ДО	РЕ	МИ	ФА	СОЛЬ	ЛЯ	СИ	ДО	РЕ	МИ	
-3	-1	0	2	4	5	7	9	11	12	14	16	
	C	D	E	F	G	A	B	C				

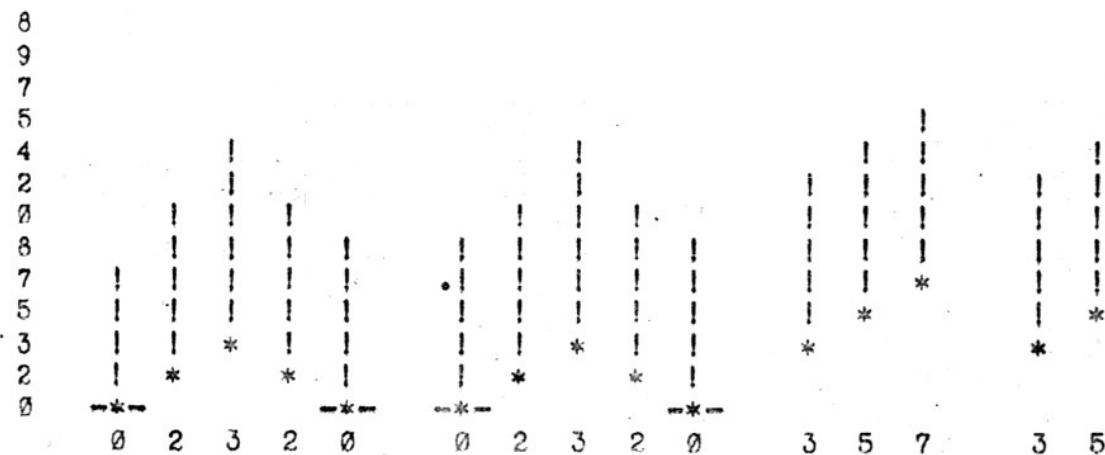
ДЛЯ ПОЛУЧЕНИЯ БОЛЕЕ ВЫСОКИХ ИЛИ БОЛЕЕ НИЗКИХ НОТ, ВЫ ДОЛЖНЫ ПРИБАВИТЬ ИЛИ ОТНЯТЬ 12 ДЛЯ КАЖДОЙ ОКТАВЫ ВВЕРХ ИЛИ ВНИЗ. НАПРИМЕР:

```

10 PRINT "FRERE CUSTAV"
20 ВЕЕР 1,0:ВЕЕР 1,2:ВЕЕР .5,3:ВЕЕР .5,2:ВЕЕР 1,0
30 ВЕЕР 1,0:ВЕЕР 1,2:ВЕЕР .5,3:ВЕЕР .5,2:ВЕЕР 1,0
40 ВЕЕР 1,3:ВЕЕР 1,5:ВЕЕР 2,7
50 ВЕЕР 1,3:ВЕЕР 1,5:ВЕЕР 2,7
60 ВЕЕР .75,7:ВЕЕР .25,8:ВЕЕР .5,7:ВЕЕР .5,5:ВЕЕР .5,3:
  ВЕЕР .6,2:ВЕЕР 1,0
70 ВЕЕР .75,7:ВЕЕР .25,8:ВЕЕР .5,7:ВЕЕР .5,5:ВЕЕР .5,3
  ВЕЕР .5,2:ВЕЕР 1,0
80 ВЕЕР 1,0:ВЕЕР 1,-5:ВЕЕР 2,0
90 ВЕЕР 1,0:ВЕЕР 1,-5:ВЕЕР 2,0

```

КОГДА ВЫ ЗАПУСТИТЕ ЭТУ ПРОГРАММУ, ВЫ УСЛЫШИТЕ ПОХОРОННЫЙ МАРШ ИЗ ПЕРВОЙ СИМФОНИИ МОЛЬЕРА, ТУ ЧАСТЬ, КОГДА ГОБЛИНЫ ХОРОНЯТ РЫЦАРЯ. ЗАПИСЬ НАЧАЛА ЭТОЙ МЕЛОДИИ В КЛЮЧЕ ДО-МИНОР С УКАЗАНИЕМ ЗНАЧЕНИЙ НОТ ПРИВЕДЕНА НА РИСУНКЕ:



ЕСЛИ ВЫ ЖЕЛАЕТЕ ИСПОЛНИТЬ МЕЛОДИЮ В ДРУГОМ КЛЮЧЕ, ВЫ ДОЛЖНЫ ВСТАВИТЬ В ВЫРАЖЕНИЕ НЕКОТОРУЮ ПЕРЕМЕННУЮ 'KEY', НАПРИМЕР ДЛЯ ВТОРОЙ СТРОКИ ПРОГРАММЫ:

```

20 ВЕЕР 1,KEY+0:ВЕЕР 1,KEY+2:ВЕЕР .5,KEY+3:ВЕЕР .5,KEY+2:
  ВЕЕР 1,KEY+0

```

ТЕПЕРЬ, ПРИ ВЫПОЛНЕНИИ ЭТОЙ ПРОГРАММЫ ВЫ МОЖЕТЕ ПРИСВОИТЬ ПЕРЕМЕННОЙ 'KEY' ЗНАЧЕНИЯ: 0 - ДЛЯ ДО-МИНОР, 2 - ДЛЯ РЕ-МИНОР, 12 - ДЛЯ ДО-МИНОР ВЕРХНЕЙ ОКТАВЫ И Т.Д. ПЕРЕМЕННАЯ 'KEY' МОЖЕТ ТАКЖЕ ПРИНИМАТЬ ЗНАЧЕНИЯ КРАТНЫЕ 1/2, 1/4, И Т.Д.

ТАКИМ ЖЕ ОБРАЗОМ МОЖНО ИЗМЕНЯТЬ И ДЛИТЕЛЬНОСТЬ ЗВУЧАНИЯ. НО ПОМНИТЕ, ЧТО КОМПЬЮТЕР МОЖЕТ ОДНОВРЕМЕННО ИСПОЛНЯТЬ ТОЛЬКО

ОДНУ НОТУ, ЧТО НЕ ПОЗВОЛЯЕТ ВОСПРОИЗВОДИТЬ СЛОЖНЫЕ МЕЛОДИИ.

ПОПРОБУЙТЕ ЗАПРОГРАММИРОВАТЬ СОБСТВЕННУЮ МЕЛОДИЮ. НАЧНИТЕ С САМОЙ ПРОСТОЙ. ЕСЛИ ВЫ НЕ ЗНАЕТЕ НОТНОЙ ГРАМОТЫ МОЖЕТЕ ИЗУЧАТЬ ЕЕ ПРЯМО НА КОМПЬЮТЕРЕ, НАПРИМЕР ФРАГМЕНТ ПРОГРАММЫ:

```
FOR N=0 TO 1000:VEER .5,N:NEXT N
```

БУДЕТ ИСПОЛНЯТЬ ПОСЛЕДОВАТЕЛЬНО НОТЫ ДО ПРЕДЕЛЬНО ВЫСОКИХ И ЗАВЕРШИТСЯ СООБЩЕНИЕМ ОБ ОШИБКЕ 'В'. ВЫ МОЖЕТЕ ТАКЖЕ ПАРАЛЛЕЛЬНО ВЫВОДИТЬ ЗНАЧЕНИЯ 'N', ЧТОБЫ ЗНАТЬ ЗНАЧЕНИЕ ИСПОЛНЯЕМОЙ НОТЫ. ФРАГМЕНТ ПРОГРАММЫ:

```
10 VEER.5,0:VEER.5,2:VEER.5,4:VEER .5,5:VEER .5,7:
   VEER.5,11:VEER .5,12:STOP
```

ИСПОЛНЯЕТ ГАММУ ДО-МИНОР, В КОТОРОЙ ИСПОЛЬЗУЮТСЯ ЧИСТЫЕ НОТЫ ОТ СРЕДНЕГО ДО ДО ВЕРХНЕГО ДО. ОДНАКО В ЭТОЙ ГАММЕ НЕЕСТЕСТВЕННЫ ИНТЕРВАЛЫ. СКРИПАЧ БЫ ИСПОЛНИЛ ЕЕ ТАК:

```
10 VEER .5,0:VEER .5,2.039:VEER .5,3.86:VEER .5,4.9
   VEER.5,7.02:VEER .5,8.84:VEER .5,10.88:VEER .5,11:
   VEER .5,12:STOP
```

ЭТИ ЖЕ ИНТЕРВАЛЫ БУДУТ ЕСТЕСТВЕННЫМИ ДЛЯ ГАММЫ ИСПОЛНЯЕМОЙ В ЛЮБОМ КЛЮЧЕ, ОТЛИЧНОМ ОТ ДО.

НЕКОТОРАЯ МУЗЫКА, НАПРИМЕР ИНДИЙСКАЯ, ИСПОЛЬЗУЕТ ИНТЕРВАЛЫ МЕНЬШЕ ЧЕМ ПОЛУТОН. ВЫ МОЖЕТЕ БЕЗ ОСОБОГО ТРУДА ЗАПРОГРАММИРОВАТЬ ЭТО В ОПЕРАТОРЕ ВЕЕР. НАПРИМЕР ДЛЯ ЗВУКА НА ЧЕТВЕРТЬ ТОНА ВЫШЕ СРЕДНЕГО ДО НАДО УКАЗАТЬ ЗНАЧЕНИЕ ЗВУКА РАВНОЕ,5.

ВЫ МОЖЕТЕ СДЕЛАТЬ КЛАВИАТУРУ КОМПЬЮТЕРА КЛАВИШАМИ МУЗЫКАЛЬНОГО ИНСТРУМЕНТА, ВЫПОЛНИВ ПЕРЕКЛЮЧЕНИЕ:

```
ROKE 23609,255
```

ВТОРОЕ ЧИСЛО ЗДЕСЬ ОПРЕДЕЛЯЕТ ПРОДОЛЖИТЕЛЬНОСТЬ НАХОЖДЕНИЯ В ЭТОМ СОСТОЯНИИ (ПОПРОБУЙТЕ ИЗМЕНЯТЬ ЕГО ОТ 0 ДО 255).

МОЖНО ТАКЖЕ ВЫВЕСТИ МУЗЫКУ НА ВНЕШНИЕ УСТРОЙСТВА, ПОДКЛЮЧАЯ ИХ К ВЫХОДНЫМ РАЗЪЕМАМ 'MIG' И 'EAR'.

ГЛАВА 20

ВНЕШНЯЯ ПАМЯТЬ НА МАГНИТНОЙ ЛЕНТЕ

=====

КРАТКОЕ СОДЕРЖАНИЕ: LOAD, SAVE, VERIFY, MERGE

ОСНОВНЫЕ КОМАНДЫ РАБОТЫ С МАГНИТОФОНОМ SAVE, LOAD И VERIFY УЖЕ РАССМАТРИВАЛИСЬ ВО ВВОДНОМ ОПИСАНИИ. ВЫ МОГЛИ ВИДЕТЬ, ЧТО LOAD ЗАТИРАЕТ СТАРУЮ ПРОГРАММУ В ПАМЯТИ КОМПЬЮТЕРА ПРИ ЗАГРУЗКЕ НОВОЙ ПРОГРАММЫ С ЛЕНТЫ. ЕСТЬ ДРУГАЯ КОМАНДА MERGE, НЕ ДЕЛАЮЩАЯ ЭТОГО. ЭТА КОМАНДА СТИРАЕТ ЛИШЬ ТЕ СТРОКИ ПРОГРАММЫ ИЛИ ПЕРЕМЕННЫЕ, КОТОРЫЕ СОВПАДАЮТ С НОМЕРАМИ СТРОК НОВОЙ ПРОГРАММЫ ИЛИ ИМЕНАМИ НОВЫХ ПЕРЕМЕННЫХ. ПРОГРАММУ 'DICE' (ИГРАЛЬНАЯ КОСТЬ) ИЗ ГЛАВЫ 11 ЗАПИШЕМ НА ЛЕНТУ ПОД ИМЕНЕМ 'DICE', А ТЕПЕРЬ ВВЕДЕМ И ВЫПОЛНИМ СЛЕДУЮЩУЮ ПРОГРАММУ:

```
1 PRINT 1
2 PRINT 2
10 PRINT 10
20 LET X = 20
```

А ЗАТЕМ ОСУЩЕСТВИМ ЕЕ ПРОВЕРКУ, ЗАМЕНИВ КОМАНДУ VERIFY 'DICE' НА КОМАНДУ MERGE 'DICE'. ВЫ УВИДИТЕ, ЧТО СТРОКИ 1 И 2 СОХРАНЯЮТСЯ, А СТРОКИ 10 И 20 ЗАМЕНЯЮТСЯ НА СТРОКИ С ЭТИМИ НОМЕРАМИ ИХ ПРОГРАММЫ 'DICE', ПЕРЕМЕННАЯ X ТОЖЕ СОХРАНЯЕТСЯ (ПРОВЕРЬТЕ PRINT X).

ТЕПЕРЬ ВЫ ЗНАЕТЕ ЧЕТЫРЕ ОПЕРАТОРА ДЛЯ РАБОТЫ С КАССЕТНЫМ МАГНИТОФОНОМ:

SAVE - ЗАПИСЫВАЕТ ПРОГРАММУ И ПЕРЕМЕННЫЕ НА МАГНИТОФОН
 VERIFY - ПРОВЕРЯЕТ ПРОГРАММУ И ПЕРЕМЕННЫЕ В ПАМЯТИ КОМПЬЮТЕРА ПО ИХ КОПИИ НА ЛЕНТЕ

LOAD - ОЧИЩАЕТ ПАМЯТЬ КОМПЬЮТЕРА ОТ ВСЕХ ПРОГРАММ И ЗАГРУЖАЕТ В НЕЕ НОВЫЕ, СЧИТАННЫЕ С МАГНИТОФОНА

MERGE - ПОДОБНА LOAD, ТОЛЬКО НЕ ОЧИЩАЕТ ВСЮ ПАМЯТЬ, А ЛИШЬ ЗАМЕНЯЕТ ТЕ СТРОКИ ПРОГРАММЫ ИЛИ ПЕРЕМЕННЫЕ, У КОТОРЫХ СОВПАДАЮТ НОМЕРА ИЛИ ИМЕНА С ТАКИМИ ЖЕ НА МАГНИТНОЙ ЛЕНТЕ.

ЗА КАЖДОЙ ИЗ ЭТИХ КОМАНД СЛЕДУЕТ КЛЮЧЕВОЕ СЛОВО - ИМЯ ПРОГРАММЫ, ОПРЕДЕЛЕННОЕ ПЕРВОНАЧАЛЬНО КОМАНДЕ SAVE. ПОКА КОМПЬЮТЕР ИМЕЕТ УКАЗАННУЮ ПРОГРАММУ, ОН ВЫВОДИТ ИМЕНА ВСЕХ ПРОГРАММ, УЖЕ ПРОЧИТАННЫЕ С ЛЕНТЫ. ИМЕЕТСЯ ДВЕ ВОЗМОЖНОСТИ ДЛЯ СНЯТИЯ СПРАВКИ С ЛЕНТЫ.

ВАРИАНТ 1, В ОПЕРАТОРАХ VERIFY, LOAD И MERGE ВМЕСТО ИМЕНИ МОЖНО УКАЗАТЬ ПУСТУЮ СТРОКУ, ТОГДА БУДЕТ ВЗЯТ ПЕРВЫЙ ВСТРЕЧЕННЫЙ ФАЙЛ.

ВАРИАНТ 2, С ИСПОЛЬЗОВАНИЕМ ОПЕРАТОРА SAVE:

SAVE STRING LINE NUMBER

ПРОГРАММА ЗАПИСЫВАЕТСЯ НА ЛЕНТУ ТАК, ЧТО КОГДА ОНА БУДЕТ ВНОВЬ СЧИТАНА ПО КОМАНДЕ LOAD (НО НЕ MERGE), ОНА АВТОМАТИЧЕСКИ УСТАНОВИТСЯ НА СТРОКУ С УКАЗАННЫМ НОМЕРОМ И САМА ИНИЦИИРУЕТ СВОЕ ВЫПОЛНЕНИЕ. КРОМЕ ТЕКСТОВ ПРОГРАММ НА ЛЕНТУ МОЖНО ЗАПИСЫВАТЬ ТАКЖЕ МАССИВЫ ИЛИ ДАННЫЕ.

ЗАПИСАТЬ НА ЛЕНТУ МАССИВ ВЫ МОЖЕТЕ, ИСПОЛЬЗУЯ КОМАНДУ SAVE С DATA ТАКИМ ОБРАЗОМ:

SAVE STRING DATA ARRAY NAME()

ЗДЕСЬ "STRING" - ИМЯ, ПРИСВАИВАЕМОЕ ФАЙЛУ ДАННЫХ, КОТОРОЕ МОЖЕТ СОСТОЯТЬ ИЗ БУКВ ИЛИ БУКВ И СИМВОЛА "X" (ПЕРЕЧЕРКНУТАЯ БУКВА "S"). ДЛЯ СТРОКОВЫХ ДАННЫХ ЭТО ТРЕБОВАНИЕ ЗДЕСЬ НЕ ВАЖНО. ЗАГРУЖАЮТСЯ ТАКИЕ ДАННЫЕ ПО КОМАНДЕ:

LOAD STRING DATA ARRAY NAME()

НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ ОПЕРАТОР MERGE.

ЕСЛИ ЗАГРУЖАЕТСЯ СТРОКОВЫЙ МАССИВ, ТО ПОСЛЕ ОБНАРУЖЕНИЯ ЕГО НА ЛЕНТЕ, КОМПЬЮТЕР ВЫДАЕТ: "CHARACTER ARRAY:" И ДАЛЕЕ ИМЯ ВТОРОГО МАССИВА. СУЩЕСТВУЕТ ВОЗМОЖНОСТЬ ЗАПИСИ НА МАГНИТНУЮ ЛЕНТУ И ОТДЕЛЬНЫХ БАЙТОВ ИНФОРМАЦИИ. ТАК НАПРИМЕР, ЭТО МОЖЕТ БЫТЬ ТЕЛЕВИЗИОННАЯ КАРТИНКА ИЛИ ОПРЕДЕЛЕННЫЕ ПОЛЬЗОВАТЕЛЕМ ГРАФИЧЕСКИЕ СИМВОЛЫ И Т.Д., ДЛЯ ЭТОГО ИСПОЛЬЗУЕТСЯ КЛЮЧЕВОЕ СЛОВО CODE, НАПРИМЕР:

SAVE "PICTURE" CODE 16384,6912

ЗДЕСЬ ПЕРВОЕ ЧИСЛО - АДРЕС ПЕРВОГО БАЙТА В ОБЛАСТИ ПАМЯТИ ГДЕ РАСПОЛОЖЕНЫ ДАННЫЕ, А ВТОРОЕ ЧИСЛО - КОЛИЧЕСТВО БАЙТОВ, КОТОРЫЕ НУЖНО ЗАПИСАТЬ НА ЛЕНТУ (6912 - ОБЪЕМ В БАЙТАХ ОДНОГО ЭКРАНА, 16384 - АДРЕС ЭКРАНА В ПАМЯТИ). ЗАГРУЖАЮТСЯ ЭТИ ДАННЫЕ ПО КОМАНДЕ:

LOAD "PICTURE" CODE

ПОСЛЕ CODE МОЖНО УКАЗАТЬ ЧИСЛА:

LOAD "PICTURE" CODE START, LENGTH

LENGTH (ДЛИНА) - ОПРЕДЕЛЯЕТ СКОЛЬКО ДАННЫХ (БАЙТАХ) НАДО ЗАГРУЗИТЬ С ЛЕНТЫ. ЕСЛИ ДЛИНА БОЛЬШЕ, ЧЕМ ЗАПИСАНО НА ЛЕНТУ, ТО ВЫДАЕТСЯ СООБЩЕНИЕ "R TARE LOADING ERROR" (ОШИБКА ЗАГРУЗКИ С ЛЕНТЫ). ЭТОТ ПАРАМЕТР МОЖНО ОПУСТИТЬ И ТОГДА КОМПЬЮТЕР СЧИТАЕТ ВСЕ ДАННЫЕ, КОТОРЫЕ ЗАПИСАНЫ НА ЛЕНТЕ.

START (НАЧАЛО) - УКАЗЫВАЕТ АДРЕС, С КОТОРОГО ДОЛЖНЫ ЗАГРУЖАТЬСЯ ДАННЫЕ И МОЖЕТ БЫТЬ ОТЛИЧНЫМ ОТ АДРЕСА, УКАЗАННОГО В SAVE, ВЫ МОЖЕТЕ ОПУСТИТЬ ЭТОТ ПАРАМЕТР В КОМАНДЕ LOAD.

ВЫРАЖЕНИЕ CODE 16384,6912 МОЖНО ЗАМЕНИТЬ НА SCREEN*;

SAVE "PICTURE" SCREEN* И ЗАТЕМ

LOAD "PICTURE" SCREEN*

ЭТО ТОТ СЛУЧАЙ КОГДА VERIFY НЕ РАБОТАЕТ, В ОСТАЛЬНЫХ СЛУЧАЯХ VERIFY МОЖНО ИСПОЛЬЗОВАТЬ ВЕЗДЕ ГДЕ ИСПОЛЬЗУЕТСЯ SAVE.

В ЗАКЛЮЧЕНИЕ, ВЕЗДЕ, ГДЕ УКАЗЫВАЕТСЯ ИМЯ ФАЙЛА НА ЛЕНТЕ,

ИСПОЛЗУЮТСЯ ТОЛЬКО ПЕРВЫЕ 10 СИМВОЛОВ, СУЩЕСТВУЕТ ЧЕТЫРЕ ТИПА ИНФОРМАЦИИ, КОТОРЫЕ МОГУТ БЫТЬ ЗАПИСАНЫ НА ЛЕНТУ:

- ПРОГРАММЫ И ПЕРЕМЕННЫЕ (СОВМЕСТНО)
- ЧИСЛОВЫЕ МАССИВЫ
- СТРОКОВЫЕ МАССИВЫ
- НЕПОСРЕДСТВЕННО БАЙТЫ

КОГДА КОМАНДЫ VERIFY, LOAD, И MERGE ОСУЩЕСТВЛЯЮТ ПОИСК ДАННЫХ НА ЛЕНТЕ, ОНИ ВЫВОДЯТ НА ЭКРАН ВСЕ СЧИТАННЫЕ ИМИ С ЛЕНТЫ ИМЕНА С УКАЗАНИЕМ ТИПА В ВИДЕ: "PROGRAM:", "NUMBER ARRAY:", "CHARACTER ARRAY:", "BYTES:"

ЕСЛИ ИМЯ - ПУСТАЯ СТРОКА, ЭТИ КОМАНДЫ БЕРУТ ПЕРВЫЙ ВСТРЕТИВШИЙСЯ ФАЙЛ С УКАЗАННЫМ ТИПОМ.

КОМАНДА SAVE СЛУЖИТ ДЛЯ ЗАПИСИ ИНФОРМАЦИИ НА ЛЕНТУ ПОД ЗАДАНЫМ ИМЕНЕМ. СООБЩЕНИЕ ОБ ОШИБКЕ F ВЫДАЕТСЯ, ЕСЛИ ВМЕСТО ИМЕНИ УКАЗАНА ПУСТАЯ СТРОКА ИЛИ ЧИСЛО СИМВОЛОВ В ИМЕНИ 11 И БОЛЕЕ. SAVE ВСЕГДА ВЫДАЕТ СООБЩЕНИЕ:

"START TAPE, THEN PRESS ANY KEY" ("ЗАПУСТИ МАГНИТОФОН И НАЖМИ ЛЮБУЮ КЛАВИШУ"), И ЖДЕТ НАЖАТИЯ. ПОСЛЕ ВСЕГО ЗАПИСЫВАЕТ ДАННЫЕ НА ЛЕНТУ.

1. ПРОГРАММА И ПЕРЕМЕННЫЕ,

SAVE NAME LINE LINE NUMBER

ЗАПИСЫВАЕТ ПРОГРАММУ НА ЛЕНТУ ТАКИМ ОБРАЗОМ, ЧТО ПОСЛЕДУЮЩАЯ КОМАНДА LOAD АВТОМАТИЧЕСКИ ВСТАВЛЯЕТ В ПРОГРАММУ

GO TO LINE NUMBER

И НАЧИНАЕТ ЕЕ ВЫПОЛНЯТЬ

2. БАЙТЫ,

SAVE NAME CODE START, LENGHT

ЗАПИСЫВАЕТ НА ЛЕНТУ "LENGHT" БАЙТ, НАЧИНАЯ С АДРЕСА START

SAVE NAME SCREEN*

ЭКВИВАЛЕНТНО SAVE NAME CODE 16384,6912

И ЗАПИСЫВАЕТ ОДИН ТЕЛЕВИЗИОННЫЙ ЭКРАН.

3. МАССИВЫ

SAVE NAME DATA LETTER () ИЛИ

SAVE NAME DATA LETTER* ()

ЗАПИСЫВАЮТ ЧИСЛОВОЙ ИЛИ СТРОКОВЫЙ МАССИВ (ТРЕБОВАНИЕ * НЕ ОТНОСИТСЯ К "NAME").

КОМАНДА VERIFY ПРОПЕРЯЕТ (СРАВНИВАЕТ) ИНФОРМАЦИЮ В ПАМЯТИ И НА ЛЕНТЕ, МОЖЕТ ВЫДАВАТЬ СООБЩЕНИЕ:

"R TAPE LOADING ERROR",

1. ПРОГРАММА И ПЕРЕМЕННЫЕ,

VERIFY NAME

2. БАЙТЫ,

VERIFY NAME CODE START LENGTH

ЕСЛИ ДАННЫХ В ФАЙЛЕ "NAME" БОЛЕЕ, ЧЕМ УКАЗАНО В "LENGHT" ТО ВЫДАЕТСЯ СООБЩЕНИЕ ОБ ОШИБКЕ "R".

VERIFY NAME CODE START

ЗДЕСЬ ОСУЩЕСТВЛЯЕТСЯ СРАВНЕНИЕ ДАННЫХ НА ЛЕНТЕ С ДАННЫМИ В ПАМЯТИ, НАЧИНАЯ С АДРЕСА, С КОТОРОГО ЗАПИСЫВАЕТСЯ НА ЛЕНТУ ПЕРВЫЙ БАЙТ ДАННЫХ.

VERIFY NAME SCREEN*

ИЛИ ЭКВИВАЛЕНТНО

VERIFY NAME CODE 16384,6912

ОДНАКО ЭТО БУДЕТ ПРОВЕРКА УЖЕ ПРОВЕРЕННОГО ФАЙЛА.

3. МАССИВЫ,

VERRIFY NAME DATA LETTER()

VERIFY NAME DATA LETTER*()

КОМАНДА LOAD ЗАГРУЖАЕТ НОВЫЕ ДАННЫЕ С ЛЕНТЫ, СТИРАЯ СТАРЫЕ ДАННЫЕ В ПАМЯТИ.

1. ПРОГРАММА И ПЕРЕМЕННЫЕ.

LOAD NAME

МОЖЕТ ВЫДАВАТЬ СООБЩЕНИЕ "4 OUT OF MEMORY", ЕСЛИ НЕТ МЕСТА ДЛЯ НОВОЙ ПРОГРАММЫ. В ЭТОМ СЛУЧАЕ СТАРАЯ ПРОГРАММА НЕ УНИЧТОЖАЕТСЯ.

2. БАЙТЫ.

LOAD NAME CODE START, LENGTH

ЕСЛИ ДАННЫХ В ФАЙЛЕ "NAME" БОЛЬШЕ, ЧЕМ УКАЗАНО В "LENGTH", ТО ВЫДАЕТСЯ СООБЩЕНИЕ R.

LOAD NAME CODE START

ПРОИЗВОДИТ ЗАГРУЗКУ ДАННЫХ ИЗ "NAME" В ПАМЯТЬ, НАЧИНАЯ С АДРЕСА "START"

LOAD NAME CODE

ЗАГРУЖАЕТ ПРОГРАММУ ПО АДРЕСУ, С КОТОРОГО ЗАПИСЫВАЛИСЬ ДАННЫЕ НА ЛЕНТУ В ФАЙЛ "NAME".

3. МАССИВЫ

LOAD NAME DATE LETTER()

ИЛИ

LOAD NAME DATE LETTERx()

УНИЧТОЖАЕТ В ПАМЯТИ МАССИВ С ИМЕНЕМ "LETTER" ИЛИ "LETTERx", ФОРМИРУЕТ НОВЫЙ МАССИВ И ПЕРЕПИСЫВАЕТ ТУДА ДАННЫЕ ИЗ ФАЙЛА "NAME", МОЖЕТ ВЫДАТЬ СООБЩЕНИЕ "4 OUT OF MEMORY" ПРИ НЕХВАТКЕ ПАМЯТИ ПОД МАССИВ. В ЭТОМ СЛУЧАЕ СТАРЫЙ МАССИВ НЕ УНИЧТОЖАЕТСЯ.

КОМАНДА MERGE ЗАГРУЖАЕТ НОВЫЕ ДАННЫЕ С ЛЕНТЫ, НЕ УНИЧТОЖАЯ СТАРЫЕ.

1. ПРОГРАММА И ПЕРЕМЕННЫЕ.

MERGE NAME

ДОПИСЫВАЕТ ПРОГРАММУ "NAME" К НЕКОТОРОЙ ПРОГРАММЕ, НАХОДЯЩЕЙСЯ В ПАМЯТИ, МОЖЕТ ВЫДАТЬ СООБЩЕНИЕ

"4 OUT OF MEMORY"

2. БАЙТЫ.

НЕ ПОДДЕРЖИВАЕТСЯ.

3. МАССИВЫ.

НЕ ПОДДЕРЖИВАЕТСЯ.

ПРИМЕР: ЗАПИСАТЬ НА ЛЕНТУ ИНФОРМАЦИЮ О 21-М ОПРЕДЕЛЕННОМ ПОЛЬЗОВАТЕЛЕМ СИМВОЛЕ.

SAVE "CHESS" CODE USR "A", 21*8.

ОБРАТНАЯ ЗАГРУЗКА

LOAD "CHESS" CODE

LOAD "CHESS" CODE USR "A"

ГЛАВА 21

УСТРОЙСТВО ПЕЧАТИ

=====

КРАТКОЕ СОДЕРЖАНИЕ: LPRINT, LLIST, COPY.

ЭТА ГЛАВА ОПИСЫВАЕТ ОПЕРАТОРЫ БЕЙСИКА, НЕОБХОДИМЫЕ ДЛЯ РАБОТЫ С ПРИНТЕРОМ ZX SPECTRUM.

ДВА ОПЕРАТОРА LPRINT И LLIST ПОДОБНЫ ОПЕРАТОРАМ PRINT И LIST, НО С ТОЙ РАЗНИЦЕЙ, ЧТО ОНИ РАБОТАЮТ НЕ С ТЕЛЕВИЗОРОМ, А С ПРИНТЕРОМ. ПОПРОБУЙТЕ ДЛЯ ПРИМЕРА ВЫПОЛНИТЬ СЛЕДУЮЩУЮ ПРОГРАММУ:

10 LPRINT " THIS PROGRAM."

20 LLIST

30 LPRINT " PRINTS OUT THE CHARACTER SET."

40 FOR N=32 TO 255

50 LPRINT CHR N

60 NEXT N

ОПЕРАТОР COPY ПОЗВОЛЯЕТ РАСПЕЧАТАТЬ ЭКРАН ТЕЛЕВИЗОРА, НАПРИМЕР, ПО LIST ТЕКСТ ПРОГРАММЫ БУДЕТ ВЫВЕДЕН НА ЭКРАН, А ЗАТЕМ ПО COPY ЕГО МОЖНО РАСПЕЧАТАТЬ НА ПРИНТЕРЕ.

ВЫ ВСЕГДА МОЖЕТЕ ПРЕКРАТИТЬ ВЫВОД НА ПЕЧАТЬ ВЫДАВ · BREAK (CAPS SHIFT И SPACE).

ЕСЛИ ВЫ ЗАДАЛИ ОПЕРАТОРЫ УПРАВЛЕНИЯ ПРИНТЕРОМ БЕЗ ПОДКЛЮЧЕНИЯ РЕАЛЬНОГО УСТРОЙСТВА, ТО ВЫВОДА ПРОСТО НЕ БУДЕТ И ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРОДОЛЖИТСЯ ДО СЛЕДУЮЩЕГО ОПЕРАТОРА.

ТЕПЕРЬ ПОПРОБУЙТЕ ВЫПОЛНИТЬ СЛЕДУЮЩУЮ ПРОГРАММУ:

```
10 FOR N=31 TO 0 STEP -1
20 PRINT AT 31-N,N CHR$(CODE"0"+N)
30 NEXT N
```

ВЫ ПОЛУЧИТЕ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ РАСПОЛОЖЕННЫХ ПО ДИАГОНАЛИ ЭКРАНА, НАЧИНАЯ С ПРАВОГО ВЕРХНЕГО УГЛА. ТЕПЕРЬ ЗАМЕНИМ В СТРОКЕ 20 " AT 31-N,N" НА "ТАВ N" ПРОГРАММА БУДЕТ РАБОТАТЬ ТАКЖЕ КАК И ПРЕЖДЕ. ТЕПЕРЬ ЗАМЕНИМ В СТРОКЕ 20 PRINT НА LPRINT. ЗАМЕТИМ, ЧТО РАЗВЕРТКИ ПО ДИАГОНАЛИ НЕ ПОЛУЧИТСЯ, А ЗАМЕНИВ ТЕПЕРЬ "ТАВ N" НА "AT 31-N,N" И СОХРАНИВ LPRINT ПОЛУЧИМ ПО ОДНОМУ СИМВОЛУ НА СТРОКУ, ЧТО И ТРЕБОВАЛОСЬ ПОЛУЧИТЬ.

ВООБЩЕ ПРИ ПЕЧАТИ ПЕРЕВОД СТРОКИ ОСУЩЕСТВЛЯЕТСЯ В СЛЕДУЮЩИХ СЛУЧАЯХ:

- А) ПРИ ЗАПОЛНЕНИИ БУФЕРА СТРОКИ
- Б) ПОСЛЕ LPRINT, ЕСЛИ ЭТО НЕ КОНЕЦ ОПЕРАТОРА И В НЕМ ВСТРЕТИЛАСЬ ЗАПЯТАЯ ИЛИ ТОЧКА С ЗАПЯТОЙ
- В) ЕСЛИ ЗАПЯТАЯ, АПОСТРОФ ИЛИ ТАВ ТРЕБУЮТ НОВОЙ СТРОКИ
- Г) ПРИ ОКОНЧАНИИ ПРОГРАММЫ, ЕСЛИ ОСТАЛИСЬ НЕВЫВЕДЕННЫЕ ДАННЫЕ.

ГЛАВА 22

ДРУГОЕ ПЕРИФЕРИЙНОЕ ОБОРУДОВАНИЕ

=====

ИМЕЕТСЯ РЯД ДРУГИХ УСТРОЙСТВ, КОТОРЫЕ МОГУТ БЫТЬ ПОДКЛЮЧЕНЫ К КОМПЬЮТЕРУ ZX SPECTRUM.

ZX MICRODRIVE - ВЫСОКОСКОРОСТНОЕ УСТРОЙСТВО ПАМЯТИ, МОЖЕТ БЫТЬ ИСПОЛЬЗОВАНО ВМЕСТО КАССЕТНОГО МАГНИТОФОНА, ОДНАКО ОНО НЕ МОЖЕТ УПРАВЛЯТЬСЯ КОМАНДАМИ SAVE, VERIFY, LOAD И MERGE, А ЛИШЬ КОМАНДАМИ PRINT, LIST, INPUT И INKEY.

ПРИ ПОМОЩИ ЭТОГО УСТРОЙСТВА МОЖНО ОРГАНИЗОВАТЬ СЕТЬ ИЗ НЕСКОЛЬКИХ КОМПЬЮТЕРОВ ZX SPECTRUM.

СТАНДАРТНЫМ ИНТЕРФЕЙСОМ ДЛЯ ZX SPECTRUM ЯВЛЯЕТСЯ RS-232, ПОСРЕДСТВОМ КОТОРОГО ПОДКЛЮЧАЮТСЯ: КЛАВИАТУРА, ПРИНТЕР И ЛЮБЫЕ ДРУГИЕ УСТРОЙСТВА ОТВЕЧАЮЩИЕ СТАНДАРТАМ ЭТОГО ИНТЕРФЕЙСА. ПРИ РАБОТЕ С ТАКИМИ УСТРОЙСТВАМИ МОГУТ ИСПОЛЬЗОВАТЬСЯ ИМЕЮЩИЕСЯ НА КЛАВИАТУРЕ ДОПОЛНИТЕЛЬНЫЕ КЛЮЧЕВЫЕ СЛОВА: OPEN#, CLOSE#, MOVE, ERASE, CAT И FORMAT.

ГЛАВА 23

ВВОД И ВЫВОД

=====

КРАТКОЕ СОДЕРЖАНИЕ: OUT, IN.

КОМПЬЮТЕР МОЖЕТ СЧИТЫВАТЬ НЕКОТОРУЮ ИНФОРМАЦИЮ И ЗАПИСЫВАТЬ ЕЕ В СВОЮ ОПЕРАТИВНУЮ ПАМЯТЬ ПО КОМАНДАМ REEK И ROKE. ВСЯ ПАМЯТЬ КОМПЬЮТЕРА, И ПЗУ И ОЗУ, ПРЕДСТАВЛЯЕТСЯ СОВОКУПНОСТЬЮ АДРЕСОВ ОТ 0 ДО 65536, КАЖДЫЙ ИЗ КОТОРЫХ АДРЕСУЕТ ОДИН БАЙТ.

ТАКИМ ЖЕ ОБРАЗОМ МОЖНО АДРЕСОВАТЬ И ЕЩЕ 65536 АДРЕСОВ, НАЗЫВАЕМЫХ ПОРТАМИ ВВОДА-ВЫВОДА. ОНИ ИСПОЛЬЗУЮТСЯ ПРОЦЕССОРОМ ДЛЯ СВЯЗИ С КЛАВИАТУРОЙ И ПРИНТЕРОМ И МОГУТ УПРАВЛЯТЬСЯ ОПЕРАТОРАМИ БЕЙСИКА IN И OUT.

IN АНАЛОГИЧЕН ОПЕРАТОРУ REEK: IN ADDRESS
ОН ИСПОЛЬЗУЕТ ОДИН АРГУМЕНТ- АДРЕС ПОРТА, И ПОЗВОЛЯЕТ СЧИТАТЬ ОДИН БАЙТ ИЗ УКАЗАННОГО ПОРТА.

OUT ПОДОБЕН ОПЕРАТОРУ ROKE:

OUT ADDRESS, VALUE

И ЗАПИСЫВАЕТ УКАЗАННЫЕ ДАННЫЕ В ЗАДАННЫЙ ПОРТ ВЫВОДА.

ZX SPECTRUM ОБОЗНАЧАЕТ ШЕСТНАДЦАТИРАЗРЯДНЫМИ АДРЕСАМИ, КОТОРЫЕ МЫ БУДЕМ ОБОЗНАЧАТЬ БУКВОЙ А:

A15, A14, A13, ..., A1, A0.

БИТЫ АДРЕСА A0, A1, A2, A3 И A4 ОЧЕНЬ ВАЖНЫ. КАК ПРАВИЛО, ОНИ В 1, НО ЕСЛИ ХОТЯ БЫ ОДИН ИЗ НИХ В 0, ЭТО ПРЕДПИСЫВАЕТ КОМПЬЮТЕРУ НЕКОТОРЫЕ ДЕЙСТВИЯ, НЕ БОЛЕЕ ЧЕМ ОДИН ИЗ ЭТИХ ПЯТИ БИТОВ МОЖЕТ БЫТЬ В 0.

БИТЫ A6 И A7 ИГНОРИРУЮТСЯ, ТАК ЧТО ЕСЛИ ВЫ ЗНАКОМЫ С ЭЛЕКТРОНИКОЙ, ТО МОЖЕТЕ ИСПОЛЬЗОВАТЬ ИХ ПО СВОЕМУ УСМОТРЕНИЮ.

БИТЫ A8, A9 ИСПОЛЬЗУЮТСЯ ИНОГДА ДЛЯ ПОЛУЧЕНИЯ ДОПОЛНИТЕЛЬНОЙ ИНФОРМАЦИИ.

ИНФОРМАЦИОННЫЙ БАЙТ МЫ БУДЕМ ОБОЗНАЧАТЬ БУКВОЙ D:

D7, D6, D5, ..., D1, D0.

ТЕПЕРЬ ПРЕДОСТАВИМ СПИСОК АДРЕСОВ ПОРТОВ. ИМЕЕТСЯ ЦЕЛЫЙ РЯД ВХОДНЫХ АДРЕСОВ ДЛЯ ЧТЕНИЯ С КЛАВИАТУРЫ, А ТАКЖЕ ВХОДНОГО РАЗЪЕМА 'EAR', САМА КЛАВИАТУРА РАЗБИТА НА 8 ПОЛУРАЗРЯДОВ ПО 5 КЛАВИШ В РЯДУ:

IN 65278 СЧИТЫВАЕТ РЯД ОТ CAPS SHIFT ДО V,

IN 65022 СЧИТЫВАЕТ РЯД ОТ A ДО G,

IN 64510 СЧИТЫВАЕТ РЯД ОТ Q ДО T,

IN 63486 СЧИТЫВАЕТ РЯД ОТ 1 ДО 5,

IN 61438 СЧИТЫВАЕТ РЯД ОТ 0 ДО 6,

IN 57342 СЧИТЫВАЕТ РЯД ОТ P ДО 7,

IN 49150 СЧИТЫВАЕТ РЯД ОТ ENTER ДО H,

IN 32766 СЧИТЫВАЕТ РЯД ОТ SPACE ДО B,

ЭТИ АДРЕСА МОГУТ БЫТЬ ВЫЧИСЛЕНЫ ИЗ ВЫРАЖЕНИЯ:

$254 + 256 * (255 - 2 * N)$ ПРИ N ПРОБЕГАЮЩИМ ОТ 0 ДО 7.

В БАЙТЕ, СЧИТАННОМ С КЛАВИАТУРЫ, БИТЫ ОТ D0 ДО D4 СЛУЖАТ ДЛЯ ОБОЗНАЧЕНИЯ ПЯТИ КЛАВИШ В ДАННОМ ПОЛУРЯДУ, D0 ДЛЯ КРАЙНЕЙ КЛАВИШИ, А D4 ДЛЯ ТОЙ, ЧТО БЛИЖЕ К ЦЕНТРУ. СОСТОЯНИЕ ОДНОГО ИЗ ЭТИХ БИТОВ 0 УКАЗЫВАЕТ, ЧТО СООТВЕТСТВУЮЩАЯ ЕМУ КЛАВИША НАЖАТА, D6 ПРИНИМАЕТ СВОЕ ЗНАЧЕНИЕ ПРИ ЧТЕНИИ С РАЗЪЕМА 'EAR'.

ВЫХОДНОЙ ПОРТ 254 ОБЕСПЕЧИВАЕТ ГРОМКОГОВОРИТЕЛЬ (D4) И РАЗЪЕМ 'MIC' (D3), А ТАКЖЕ УСТАНОВКУ ЦВЕТА (D2, D1, D0).

ПОРТ 251 ОБЕСПЕЧИВАЕТ СВЯЗЬ С ПРИНТЕРОМ, КАК ЧТЕНИЕ ТАК И ЗАПИСЬ, ЧТЕНИЕ ДЛЯ ПРОВЕРКИ ГОТОВНОСТИ ПРИНТЕРА К РАБОТЕ.

ПОРТЫ 254, 247 И 239 ИСПОЛЬЗУЮТСЯ ДЛЯ СВЯЗИ С ДОПОЛНИТЕЛЬНЫМИ УСТРОЙСТВАМИ, ОПИСАННЫМИ В ГЛАВЕ 22.

ЗАПУСТИТЕ СЛЕДУЮЩУЮ ПРОГРАММУ:

10 FOR N=0 TO 7:REM N=ROM NUMBER(НОМЕР ПОЛУРЯДА)

20 LET A=254+256*(255-2*N)

30 PRINT AT=0,0 IN A: GO TO 30

И ПОНАЖИМАЙТЕ ПО ОДНОЙ КЛАВИШЕ В КАЖДОМ ПОЛУРЯДУ. ПОСЛЕ НАЖАТИЯ ОЧЕРЕДНОЙ КЛАВИШИ ВВЕДИТЕ BREAK, А ЗАТЕМ NEXT N.

НИЖЕ, НА РИСУНКЕ ПОКАЗАНО РАСПРЕДЕЛЕНИЕ КОНТАКТОВ РАЗЪЕМА;

A14-----		-----A15
A12-----		-----A13
+5V-----		-----D7
+9V-----		-----
0V-----		-----D0
0V-----		-----D1
СК-----		-----D2
A0-----		-----D6
A1-----		-----D5
A2-----		-----D3
A3-----		-----D4

IORQGE	-----	-----	INT
0V	-----	-----	NMI
VIDEO	-----	-----	HALT
Y	-----	-----	MREQ
V	-----	-----	IOREQ
U	-----	-----	RD
BUSRQ	-----	-----	WR
BRESET	-----	-----	5V
A7	-----	-----	WAIT
A6	-----	-----	+12V
A5	-----	-----	12V
A4	-----	-----	MI
ROMOS	-----	-----	RFSH
BUSACK	-----	-----	A8
A9	-----	-----	A10
A11	-----	-----	

КНИГИ "ИСКУССТВО СХЕМОТЕХНИКИ" П.ХОРОВИЦ... У.ХИЛЛ
 МИР, 1986Г. ТОМ 2. СТР. 579-580
 "ТЕХНИЧЕСКИЕ СРЕДСТВА МИКРОПРОЦЕССОРНЫХ СИСТЕМ"
 ДЖ. КОФРОН. МИР 1983Г. СТР. 334-340

С И С Т Е М О Я	HE M1	<	27	МИКРОПРОЦЕССОР	30	>	A0	А Д Р И С Н А Я
	HE MREQ	<	19		31	>	A1	
	HE IONG	<	20		---			
	HE RD	<	21		---			
	HE WR	<	22		40	>	A10	
	HE RFSH	<	28		1	>	A11	
	HE HALT	<	18		2	>	A12	
	HE WAIT	<	24		---			
	HE INT	<	16		---			
	HE WNI	<	17		---			
Ц П	HE RESET	<	26	МИКРОПРОЦЕССОР	5	>	A15	Х
	HE BUSRQ	<	25		14	>	D0	
	HE BUSAK	<	23		15	>	D1	
	FTAKT	<	6		12	>	D2	
	+5V	<	11		8	>	D3	
Ц П	КОРПУС	<	29	МИКРОПРОЦЕССОР	7	>	D4	Н
					9	>	D5	
					10	>	D6	Е
					13	>	D7	

РИС.1 НАЗНАЧЕНИЕ ВЫВОДОВ МИКРОПРОЦЕССОРА Z80, Z80A

- A0 - A15** (АДРЕСНАЯ ШИНА). ВЫХОДЫ С ТРЕМЯ УСТОЙЧИВЫМИ СОСТОЯНИЯМИ АКТИВНЫЙ УРОВЕНЬ СИГНАЛОВ-ВЫСОКИЙ. АДРЕСУЕТ ОЗУ ИЛИ УВВ (ДО 64 К ДЛЯ ОЗУ).
- D0 - D7** (ШИНА ДАННЫХ). ВХОДЫ-ВЫХОДЫ С ТРЕМЯ УСТОЙЧИВЫМИ СОСТОЯНИЯМИ. АКТИВНЫЙ УРОВЕНЬ ВЫСОКИЙ.
- NE N** (МАШИННЫЙ ЦИКЛ). ВЫХОД. АКТИВНЫЙ СИГНАЛ НИЗКИЙ. УКАЗЫВАЕТ, ЧТО В ТЕКУЩЕМ ЦИКЛЕ ОСУЩЕСТВЛЯЕТСЯ ВЫБОРКА КОП.
- NE MREQ** (ЗАПРОС ПАМЯТИ). ВЫХОД С ТРЕМЯ УСТОЙЧИВЫМИ СОСТОЯНИЯМИ. АКТИВНЫЙ УРОВЕНЬ - НИЗКИЙ, СИГНАЛ УКАЗЫВАЕТ, ЧТО НА АДРЕСНОЙ ШИНЕ УСТАНОВЛЕН АДРЕС ДЛЯ ОПЕРАЦИИ ЧТЕНИЯ ИЛИ ЗАПИСИ В ПАМЯТЬ.
- NE IORD** (ЗАПРОС ВВОДА - ВЫВОДА). ВЫХОД С ТРЕМЯ УСТОЙЧИВЫМИ СОСТОЯНИЯМИ. АКТИВНЫЙ УРОВЕНЬ СИГНАЛА-НИЗКИЙ. СИГНАЛ УКАЗЫВАЕТ, ЧТО МЛАДШИЙ БАЙТ ШИНЫ АДРЕСА СОДЕРЖИТ АДРЕС УВВ, КРОМЕ ТОГО, ЭТОТ СИГНАЛ ГЕНЕРИРУЕТСЯ ПОСЛЕ ВЫДАЧИ ПОДТВЕРЖДЕНИЯ ПРЕРЫВАНИЯ, ТЕМ САМЫМ УКАЗЫВАЯ, ЧТО ВЕКТОР ПРЕРЫВАНИЯ МОЖЕТ БЫТЬ ПОМЕЩЕН НА ШИНУ ДАННЫХ.
- NE RD** (ЧТЕНИЕ ИЗ ПАМЯТИ). ВЫХОД С ТРЕМЯ УСТОЙЧИВЫМИ СОСТОЯНИЯМИ. АКТИВНЫЙ УРОВЕНЬ СИГНАЛА - НИЗКИЙ. СИГНАЛ УКАЗЫВАЕТ, ЧТО ЦП ГОТОВ К ЧТЕНИЮ ДАННЫХ ИЗ ПАМЯТИ ИЛИ ИЗ УСТРОЙСТВА ВВ. АДРЕСОВАННОЕ УВВ ИЛИ ПАМЯТЬ ИСПОЛЬЗУЕТ ЭТОТ СИГНАЛ ДЛЯ СТРОБИРОВАНИЯ ПРИ ПОДАЧЕ ДАННЫХ НА ШИНЫ ДАННЫХ ЦП.
- NE WR** (ЗАПИСЬ В ПАМЯТЬ). ВЫХОД С ТРЕМЯ УСТОЙЧИВЫМИ СОСТОЯНИЯМИ. АКТИВНЫЙ УРОВЕНЬ СИГНАЛА-НИЗКИЙ, СИГНАЛ УКАЗЫВАЕТ, ЧТО НА ШИНЕ ДАННЫХ СОДЕРЖАТСЯ ДАННЫЕ, ПРЕДНАЗНАЧЕННЫЕ ДЛЯ ЗАПИСИ В ПАМЯТЬ ИЛИ ВЫВОДА НА УСТРОЙСТВО ВЫВОДА.
- NE RFSH** (ВОССТАНОВЛЕНИЕ). ВЫХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ, СИГНАЛ УКАЗЫВАЕТ, ЧТО МЛАДШИЕ СЕМЬ РАЗРЯДОВ ШИНЫ АДРЕСА СОДЕРЖАТ АДРЕС ВОССТАНОВЛЕНИЯ ДЛЯ ОЗУ И ТЕКУЩИЙ СИГНАЛ НЕ MREQ, КОТОРЫЙ ДОЛЖЕН ИСПОЛЬЗОВАТЬСЯ ДЛЯ ВОССТАНОВЛЕНИЯ ДИНАМИЧЕСКОГО ОЗУ.
- NE HALT** (ОСТАНОВ). ВЫХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ. СИГНАЛ УКАЗЫВАЕТ, ЧТО ЦП ВЫПОЛНИЛ КОМАНДУ HALT И ОЖИДАЕТ ПОЯВЛЕНИЕ ЛИБО НЕМАСКИРУЕМОГО ПРЕРЫВАНИЯ, ЛИБО МАСКИРУЕМОГО ПРЕРЫВАНИЯ, ПОСЛЕ КОТОРОГО ОН ПРОДОЛЖИТ РАБОТУ. ПЕРЕД ВЫПОЛНЕНИЕМ HALT ЗАНОСИТ В ОЗУ ИНФОРМАЦИЮ, КОТОРАЯ НУЖНА ДЛЯ ВОССТАНОВЛЕНИЯ.
- NE WAIT** (ОЖИДАНИЕ). ВХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ. СИГНАЛ УКАЗЫВАЕТ МИКРОПРОЦЕССОРУ, ЧТО АДРЕСУЕМАЯ ПАМЯТЬ ИЛИ УСТРОЙСТВО ВВОДА-ВЫВОДА ГОТОВЫ К ПЕРЕДАЧЕ ДАННЫХ, ЦП ЖДЕТ ПОКА АКТИВЕН ЭТОТ УРОВЕНЬ.
- NE INT** (ЗАПРОС НА МАСКИРУЕМОЕ ПРЕРЫВАНИЕ). ВХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ, ЗАПРОС БУДЕТ ВОСПРИНЯТ ЦП В КОНЦЕ ВЫПОЛНЕНИЯ ТЕКУЩЕЙ КОМАНДЫ, ЕСЛИ ТРИГГЕР РАЗРЕШЕНИЯ ПРЕРЫВАНИЯ IFF, НЕ УПРАВЛЯЕМЫЙ ВНУТРЕННИМИ ПРОГРАММНЫ-

МИ СРЕДСТВАМИ УСТАНОВЛЕН В ОПРЕДЕЛЕННОЕ СОСТОЯНИЕ.

- НЕ NMI (ЗАПРОС НА НЕМАСКИРУЕМОЕ ПРЕРЫВАНИЕ). ВХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ. ЭТО ПРЕРЫВАНИЕ ИМЕЕТ БОЛЕЕ ВЫСОКИЙ ПРИОРИТЕТ ЧЕМ INT. РАСПОЗНАЕТСЯ В КОНЦЕ ТЕКУЩЕЙ КОМАНДЫ. СИГНАЛ АВТОМАТИЧЕСКИ ПЕРЕВОДИТ ЦП К ВЫПОЛНЕНИЮ ПРОГРАММЫ С АДРЕСА 66(HEX).
- НЕ RESET (СБРОС). ВХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ. ПРИ ПОСТУПЛЕНИИ СИГНАЛА ВЫПОЛНЯЮТСЯ СЛЕДУЮЩИЕ ДЕЙСТВИЯ:
 А) СБРОС ТРИГГЕРА РАЗРЕШЕНИЯ ПРЕРЫВАНИЯ IFF.
 Б) ОЧИСТКА СЧЕТЧИКА КОМАНД И РЕГИСТРОВ I И R.
 В) ШИНЫ АДРЕСНАЯ И ДАННЫХ В СОСТОЯНИИ ВЫСОКОГО СОПРОТИВЛЕНИЯ.
 Г) ДЛЯ ВСЕХ УПРАВЛЯЮЩИХ ВЫХОДНЫХ СИГНАЛОВ УСТАНОВЛИВАЕТСЯ НЕАКТИВНЫЙ УРОВЕНЬ.
- НЕ BUSRQ (ЗАПРОС ШИН). ВХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ. СИГНАЛ ИМЕЕТ БОЛЕЕ ВЫСОКИЙ ПРИОРИТЕТ, ЧЕМ NMI И ВСЕГДА РАСПОЗНАЕТСЯ В КОНЦЕ ТЕКУЩЕГО МАШИННОГО ЦИКЛА. ОН ИСПОЛЬЗУЕТСЯ ДЛЯ ОРГАНИЗАЦИИ ПРЯМОГО ДОСТУПА К ПАМЯТИ (ПДП) И ПЕРЕВОДИТ В СОСТОЯНИЕ ВЫСОКОГО СОПРОТИВЛЕНИЯ ВСЕ ШИНЫ И ТРИСТАВИЛЬНЫЕ ВЫХОДЫ СИГНАЛОВ УПРАВЛЕНИЯ. ПОСЛЕ ЧЕГО ЭТИМИ ШИНАМИ МОГУТ УПРАВЛЯТЬ ДРУГИЕ ВНЕШНИЕ УСТРОЙСТВА
- НЕ BUSAK (ПОДТВЕРЖДЕНИЕ ПЕРЕВОДА ШИН В СОСТ. ВЫСОКОГО СОПРОТИВЛЕНИЯ). ВЫХОД, АКТИВНЫЙ УРОВЕНЬ НИЗКИЙ. СИГНАЛ ПОДАЕТСЯ НА ЗАПРАШИВАЕМОЕ ВНЕШНЕЕ УСТРОЙСТВО.

ГЛАВА 24

ПАМЯТЬ

=====

КРАТКОЕ СОДЕРЖАНИЕ: CLEAR

ВСЯ ПАМЯТЬ КОМПЬЮТЕРА РАЗБИТА НА БАЙТЫ, КАЖДЫЙ ИЗ КОТОРЫХ ПРЕДСТАВИМ ЧИСЛОМ ОТ 0 ДО 255. КАЖДЫЙ БАЙТ МОЖЕТ БЫТЬ ЗАПИСАН В ПАМЯТИ ПО ОПРЕДЕЛЕННОМУ АДРЕСУ ОТ 0 ДО FFFFH (H - ЗДЕСЬ И ДАЛЕЕ ОЗНАЧАЕТ ШЕСТНАДЦАТИРИЧНОЕ ПРЕДСТАВЛЕНИЕ ЧИСЛА). САМ АДРЕС МОЖЕТ БЫТЬ ЗАПИСАН В ПАМЯТЬ КАК ДВА БАЙТА. НА ДИАГРАММЕ ПОКАЗАНО РАСПРЕДЕЛЕНИЕ ПАМЯТИ КОМПЬЮТЕРА ZX SPECTRUM:

0	3FFFH	4000H	7FFFH	8000H	FFFFH
ROM (ПЗУ)	RAM (ОЗУ) (СИСТЕМНАЯ ПАМЯТЬ)		ДОСТУПНО ДЛЯ ПРОГРАММ		
0	16383	16384	32767	32768	65536

ДЛЯ ПОЛУЧЕНИЯ СОДЕРЖИМОГО ЛЮБОЙ ОБЛАСТИ ПАМЯТИ ИСПОЛЬЗУЕТСЯ ФУНКЦИЯ PEEK С АДРЕСОМ В КАЧЕСТВЕ АРГУМЕНТА, ФУНКЦИЯ ВОЗВРАЩАЕТ ЗНАЧЕНИЕ БАЙТА ПО ЭТОМУ АДРЕСУ.

РАССМАТРИВАЕМАЯ НИЖЕ ПРОГРАММА ВЫВОДИТ СОДЕРЖИМОЕ ПЕРВЫХ 21 БАЙТОВ И ROM С ИХ АДРЕСАМИ:

```
10 PRINT " ADDRESS" TAB 8 " BYTE"
20 FOR A=0 TO 20
30 PRINT A TAB 8 PEEK A
40 NEXT A
```

ДЛЯ ИЗМЕНЕНИЯ СОДЕРЖИМОГО ПАМЯТИ (ТОЛЬКО ДЛЯ ОЗУ), ИСПОЛЬЗУЕТСЯ ОПЕРАТОР POKE В ФОРМЕ:

```
POKE ADDRESS, NEW CONTENTS,
```

ГДЕ " ADDRESS" И " NEW CONTENTS" - ЧИСЛОВЫЕ ВЫРАЖЕНИЯ, НАПРИМЕР: POKE 31000,57
 "NEW ADDRESS" МОЖЕТ ПРИНИМАТЬ ЗНАЧЕНИЯ ОТ -255 ДО +255.
 ВСЯ ПАМЯТЬ ПОДРАЗДЕЛЯЕТСЯ НА ОБЛАСТИ, ПРЕДНАЗНАЧЕННЫЕ ДЛЯ ХРАНЕНИЯ ИНФОРМАЦИИ РАЗЛИЧНОГО НАЗНАЧЕНИЯ, ЧТО ПОКАЗАНО НА ДИАГРАММЕ:

ОБЛАСТЬ ЭКРАНА TV	АТТРИБУТЫ	БУФЕР ПРИНТЕРА	СИСТЕМНЫЕ ПЕРЕМЕННЫЕ	ПЛАН ДОПОЛНИТЕЛЬНОЙ ПАМЯТИ	
16384	22528	23296	23562	23734	CHAN

КАНАЛЬНАЯ ИНФОРМАЦИЯ	80Н	ПРОГРАММА НА БЕЙСИКЕ	ПЕРЕМЕННЫЕ ПРОГРАММЫ	80Н	РЕДАКТИРУЕМЫЕ СТРОКИ ПРОГРАММЫ
CHANS	PROG	VARs	E	LINE	

СЧИТАННЫЕ NL ДАННЫЕ	РАБОЧ. ОБЛАСТЬ	СТЕК КАЛЬКУЛЯ- ТОРА	РЕЗЕРВ	АППАРАТ- НЫЙ СТЕК	СТЕК ПЕРЕХОДОВ К ПОДПРОГР.
WORKSP	STKBOT	STKEND	SP		

7 ЗЕН ОПРЕДЕЛЯЕМЫЕ
ПОЛЬЗОВАТЕЛЕМ
СИМВОЛЫ

RAMTOP UDG P RAMT

ОБЛАСТЬ ТЕЛЕВИЗИОННОГО ЭКРАНА СОДЕРЖИТ ОБРАЗ ТЕКУЩЕГО КАДРА, ОНА ДОСТУПНА ДЛЯ ОПЕРАТОРОВ РЕЕК И РОКЕ. КАЖДАЯ ПОЗИЦИЯ ЭКРАНА ПРЕДСТАВИМА МАТРИЦЕЙ 8Х8 ТОЧЕК (ОДИН БАЙТ НА КАЖДЫЙ РЯД ИЗ 8 ТОЧЕК) ОДНАКО ЭТИ ВОСЕМЬ БАЙТ ХРАНЯТСЯ В ПАМЯТИ НЕ ВМЕСТЕ.

ПОЛНЫЙ ЭКРАН ПРЕДСТАВЛЯЕТ СОБОЙ 24 СТРОКИ ПО 32 СИМВОЛА. КАЖДАЯ СТРОКА ЭКРАНА ПРОПИСЫВАЕТСЯ 8-МЬЮ СТРОКАМИ РАЗВЕРТКИ ТЕЛЕВИЗИОННОГО ЭКРАНА. ИТОГО ДЛЯ ЗАПИСИ ОДНОГО ЭКРАНА ВЫПОЛНЯЕТСЯ 172 СКАНИРОВАНИЯ И В ПАМЯТИ РЯДОМ ХРАНЯТСЯ БАЙТЫ ОДНОИМЕННЫХ РЯДОВ МАТРИЦ СОСЕДНИХ ПОЗИЦИИ ЭКРАНА.

ОБЛАСТЬ АТТРИБУТОВ СОДЕРЖИТ ДАННЫЕ О ЦВЕТЕ И ДРУГИХ ПАРАМЕТРАХ КАЖДОЙ ПОЗИЦИИ ЭКРАНА. ИСПОЛЬЗУЕТСЯ В ФОРМАТЕ ATTR.

БУФЕР ПРИНТЕРА СОДЕРЖИТ СИМВОЛЫ, ПЕРЕДАВАЕМЫЕ НА ПЕЧАТЬ.

ОБЛАСТЬ СИСТЕМНЫХ ПЕРЕМЕННЫХ СОДЕРЖИТ ДАННЫЕ УПРАВЛЕНИЯ ВЫЧИСЛЕНИЯМИ. ОНИ ПОЛНОСТЬЮ ОПИСАНЫ В СЛЕДУЮЩЕЙ ГЛАВЕ. НЕКОТОРЫЕ ИЗ НИХ (CHANS, PROG, VARs, E LINE И ТД.) СОДЕРЖАТ АДРЕСА ГРАНИЦ МЕЖДУ СИСТЕМНЫМИ ОБЛАСТЯМИ ПАМЯТИ, НО ЭТО НЕ ПЕРЕМЕННЫЕ БЕЙСИКА, И ИХ ИМЕНА НЕ РАСПОЗНАЮТСЯ КОМПЬЮТЕРОМ.

ОБЛАСТЬ ПЛАНОВ ДОПОЛНИТЕЛЬНОЙ ПАМЯТИ ИСПОЛЬЗУЕТСЯ ТОЛЬКО С MICRODRIVE.

В ОБЛАСТИ КАНАЛЬНОЙ ИНФОРМАЦИИ СОДЕРЖАТСЯ ДАННЫЕ ОБ УСТРОЙСТВАХ ВВОДА-ВЫВОДА, А ИМЕННО: КЛАВИАТУРЕ (С НИЖНЕЙ ПОЛОВИНОЙ ЭКРАНА), ВЕРХНЕЙ ПОЛОВИНЕ ЭКРАНА В ПРИНТЕРЕ.

КАЖДАЯ СТРОКА В ОБЛАСТИ БЕЙСИК ПРОГРАММ ИМЕЕТ ФОРМАТ:

СТАРШИЙ БАЙТ		МЛАДШИЙ БАЙТ		ТЕКСТ	00
2 БАЙТА		2 БАЙТА			
НОМЕР	СТРОКИ	ДЛИНА			END
		ТЕКСТА + ENTER			

ЧИСЛОВЫЕ КОНСТАНТЫ В ПРОГРАММЕ ПРЕДСТАВЛЕНЫ В ДВОИЧНОЙ ФОРМЕ ИСПОЛЬЗУЯ CHR*14 И СЛЕДУЮЩИЕ ЗА НИМ 5 БАЙТ САМОГО ЧИСЛА. ПЕРЕМЕННЫЕ ИМЕЮТ РАЗЛИЧНЫЕ ФОРМАТЫ ПРЕДСТАВЛЕНИЯ В ПАМЯТИ. НИЖЕ ПРЕДСТАВЛЕН ФОРМАТ ЗАПИСИ ЧИСЛА, ИМЯ КОТОРОГО СОСТОИТ ИЗ ОДНОЙ БУКВЫ.

60H	1 БАЙТ ПОРЯДКА	БИТ ЗНАКА	4 БАЙТА МАНТИССЫ
-----	-------------------	--------------	---------------------

БУКВУ: ФОРМАТ РАЗМЕЩЕНИЯ ЧИСЛА, ЕСЛИ ИМЯ ИМЕЕТ БОЛЕЕ ЧЕМ ОДНУ

60H	NNN	...	NNN	5 БАЙТ ЗНАЧЕНИЯ ЧИСЛА
1 БУКВА	2 БУКВА		ПОСЛЕДНЯЯ БУКВА	

ФОРМАТ РАЗМЕЩЕНИЯ ЧИСЛОВОГО МАССИВА;

NNN	2 БАЙТА	1 БАЙТЕ	2 БАЙТА	2 БАЙТА
БУКВА	ОБЩАЯ ДЛИНА ЭЛЕМЕНТОВ + 1 НА КАЖДОЕ ИЗМЕРЕНИЕ	НОМЕРА РАЗМЕР- НОСТЕЙ	ПЕРВАЯ РАЗМЕР- НОСТЬ	ПОСЛЕДНЯЯ РАЗМЕРНОСТЬ

ПО 5 БАЙТОВ НА КАЖДЫЙ
ЭЛЕМЕНТ

ПОРЯДОК ЭЛЕМЕНТОВ СЛЕДУЮЩИЙ:

1. ЭЛЕМЕНТЫ, ДЛЯ КОТОРЫХ ПЕРВАЯ РАЗМЕРНОСТЬ РАВНА 1.
2. ЭЛЕМЕНТЫ, ДЛЯ КОТОРЫХ ПЕРВАЯ РАЗМЕРНОСТЬ РАВНА 2.
3. ЭЛЕМЕНТЫ, ДЛЯ КОТОРЫХ ПЕРВАЯ РАЗМЕРНОСТЬ РАВНА 3 И Т.Д.

ЗАТЕМ В ТОМ ЖЕ ПОРЯДКЕ ПО СЛЕДУЮЩЕЙ РАЗМЕРНОСТИ И Т. Д.
НАПРИМЕР, ЭЛЕМЕНТЫ МАССИВА С РАЗМЕРНОСТЬЮ (3,6) РАСПОЛОЖАТСЯ
ПАМЯТИ СТРОБИРОВАНИЯ В СЛЕДУЮЩЕМ ПРЯДКЕ:

B(1,1), B(1,2), B(1,3), B(1,4), B(1,5), B(1,6), B(2,1)...
B(3,6)

ФОРМАТ РАЗМЕЩЕНИЯ УПРАВЛЯЮЩИХ ПЕРЕМЕННЫХ ДЛЯ FOR-NEXT ОПЕРАТОРОВ :

5 БАЙТ 5 БАЙТ 5 БАЙТ 2 БАЙТА 1 БАЙТ

БУКВА ЗНАЧЕНИЕ ОГРАНИ- ПРИРАМЕ- СТРОКА ЦИКЛА НОМЕР
 ЧЕНИЕ НИЕ ОПЕРАТОРА

ФОРМАТ РАЗМЕЩЕНИЯ СТРОКИ СИМВОЛОВ:

2 БАЙТА ТЕКСТ(МОЖЕТ БЫТЬ ПУСТАЯ СТРОКА)

БУКВА КОЛИЧЕСТВО СИМВОЛОВ

ФОРМАТ РАЗМЕЩЕНИЯ СТРОКОВОГО МАССИВА:

2 БАЙТА 1 БАЙТ 2 БАЙТА

БУКВА ОБЩЕЕ ЧИСЛО ЭЛЕМЕНТОВ НОМЕРА РАЗ- 1-Я РАЗМЕРНОСТЬ
 ПЛЮС 1 НА КАЖДУЮ ВАЗМ. МЕРНОСТИ

2 БАЙТА ПО ОДНОМУ БАЙТУ НА КАЖДЫЙ
 ЭЛЕМЕНТ

ПОСЛЕДНЯЯ ЭЛЕМЕНТЫ
РАЗМЕРНОСТЬ

ПРОГРАММИРУЕМЫЙ СТЕК ЕСТЬ ЧАСТЬ ИНТЕРПРЕТАТОРА БЕЙСИКА. АППАРАТНЫЙ СТЕК ИСПОЛЬЗУЕТСЯ МИКРОПРОЦЕССОРОМ Z80 ДЛЯ ЗАПОМИНАНИЯ АДРЕСОВ ВОЗВРАТА.

РЕЗЕРВ В ДАННОЙ ВЕРСИИ НЕ ИСПОЛЬЗУЕТСЯ.

НАЗНАЧЕНИЕ СТЕКА ПЕРЕХОДОВ К ПОДПРОГРАММАМ ОПИСАНО В ГЛАВЕ 5 БАЙТ, АДРЕСУЕМЫЙ ПО РАМТОР СОДЕРЖИТ ВЕРХНИЙ АДРЕС, ИСПОЛЬЗУЕМЫЙ БЕЙСИКОМ.

ДАЖЕ ОПЕРАТОР NEW, КОТОРЫЙ ОЧИЩАЕТ ОЗУ, НЕ ИЗМЕНЯЕТ СОДЕРЖИМОГО ОБЛАСТИ, ОПРЕДЕЛЯЕМЫХ ПОЛЬЗОВАТЕЛЕМ СИМВОЛОВ.

ВЫ МОЖЕТЕ ИЗМЕНИТЬ АДРЕС РАМТОР В ОПЕРАТОРЕ CLEAR:

CLEAR NEW РАМТОР ,

ПО КОТОРОМУ:

А) ОЧИЩАЮТСЯ ВСЕ ОБЛАСТИ ПЕРЕМЕННЫХ

Б) ОЧИЩАЕТСЯ ОБЛАСТЬ ЭКРАНА (ПОДОБНО CLS)

В) ПЕРЕУСТАНАВЛИВАЕТСЯ ПОЗИЦИЯ PLOT В ЛЕВЫЙ НИЖНИЙ УГОЛ ЭКРАНА

Г) ВЫПОЛНЯЕТСЯ ФУНКЦИЯ RESTORE

Д) ОЧИЩАЕТСЯ СТЕК ПЕРЕХОДОВ И УСТАНАВЛИВАЕТСЯ НОВОЕ ЗНАЧЕНИЕ РАМТОР.

ФУНКЦИЯ RUN ТАКЖЕ ВЫПОЛНЯЕТ ДЕЙСТВИЯ CLEAR, ХОТЯ И НЕ ИЗМЕНЯЕТ ЗНАЧЕНИЕ РАМТОР.

ИСПОЛЬЗУЯ ФУНКЦИЮ CLEAR, ВЫ МОЖЕТЕ СМЕЩАТЬ РАМТОР, УВЕЛИЧИВАЯ ОБЛАСТЬ ДЛЯ БЕЙСИКА, УМЕНЬШАЯ ТЕМ САМЫМ ОБЛАСТЬ ОПРЕДЕЛЯЕМЫЙ

ПОЛЬЗОВАТЕЛЕМ СИМВОЛОВ. МОЖНО НЕСКОЛЬКО УВЕЛИЧИТЬ ДОСТУПНУЮ ЧАСТЬ РАМ (ОЗУ), ИСПОЛЬЗУЯ ФУНКЦИЮ NEW, НАПРИМЕР, ВЫПОЛНЕНИЕ NEW ЗАТЕМ CLEAR 23800 ПОМОГАЕТ КОМПЬЮТЕРУ ПРИ ПЕРЕПОЛНЕНИИ ОЗУ.

ВСЕ УКАЗАННЫЕ ДЕЙСТВИЯ МОГУТ ПРИВОДИТЬ К ДВУМ СООБЩЕНИЯМ ОБ ОШИБКЕ И ВЫДАЧЕ ЗВУКОВОГО СИГНАЛА:

" 4 MEMORU FULL " (ПЕРЕПОЛНЕНИЕ ПАМЯТИ)

" G NO ROOM FOR LINE " (НЕТ МЕСТА ДЛЯ СТРОКИ ПРОГРАММЫ)

МОЖНО ИЗМЕНИТЬ ДЛИТЕЛЬНОСТЬ ПОДАЧИ ЗВУКОВОГО СИГНАЛА, ИЗМЕНЯЯ ЧИСЛО ПО АДРЕСУ 23608. ПО УМОЛЧАНИЮ ПРЕДПОЛАГАЕТСЯ 64.

ЧИСЛА (ЗА ИСКЛЮЧЕНИЕМ 0) МОГУТ ЗАПИСЫВАТЬСЯ В ПОКАЗАТЕЛЬНОЙ ФОРМЕ КАК: $+M*2^E$

ГДЕ M — МАНТИСА В ИНТЕРВАЛЕ 0.5, ..., 1 (НЕ МОЖЕТ БЫТЬ 1)

E — ЭКСПОНЕНТА, ПОЛОЖИТЕЛЬНОЕ ИЛИ ОТРИЦАТЕЛЬНОЕ ЧИСЛО.

ДОПУСТИМ, ВЫ ЗАПИСАЛИ "N" В ДВОИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ.

"M" — ДРОБНОЕ И ИМЕЕТ ДВОИЧНУЮ ТОЧКУ (ПОДОБНО ДЕСЯТИЧНОЙ ТОЧКЕ), ТО ТОГДА БУДЕТ:

1/2	-->	.1
1/4	-->	.01
3/4	-->	.11 И Т.Д.

НАШЕ ЧИСЛО "N" МЕНЬШЕ, ЧЕМ 1, ЗНАЧИТ У НЕГО НЕТ БИТОВ ПЕРЕД ДВОИЧНОЙ ТОЧКОЙ, А ПОСКОЛЬКУ ОНО БОЛЬШЕ 0.5, ТО ЛЕВЫЙ БИТ, СЛЕДУЮЩИЙ ЗА ТОЧКОЙ — ЭТО 1.

ДЛЯ ЗАПИСИ ЧИСЛА В ПАМЯТЬ МЫ ИСПОЛЬЗУЕМ 5 БАЙТОВ В СЛЕДУЮЩЕМ ПОРЯДКЕ: . .

А) ЗАПИСЫВАЕМ ПЕРВЫЕ 8 БИТОВ МАНТИСЫ ВО ВТОРОЙ БАЙТ (МЫ ПОМНИМ, ЧТО ПЕРВЫЙ БИТ ЭТО 1), ВТОРЫЕ 8 БИТОВ В ТРЕТИЙ БАЙТ И ТАК ДО ПЯТОГО БАЙТА.

Б) ЗАМЕНЯЕМ ПЕРВЫЙ БИТ ВТОРОГО БАЙТА, В КОТОРОМ ЗАПИСАНА 1 НА ЗНАКОВЫЙ БИТ (0 ДЛЯ +, И 1 ДЛЯ -)

В) ЗАПИСЫВАЕМ В ПЕРВЫЙ БАЙТ ЭКСПОНЕНТУ + 128, НАПРИМЕР, МЫ ХОТИМ ЗАПИСАТЬ ЧИСЛО 1/10.

$$1/10 = 4/5 * 2^{*-3}$$

МАНТИСА БУДЕТ: .11001100110011001100110011001100

(ПОСКОЛЬКУ 33-Й БИТ РАВЕН 1, МЫ ДОЛЖНЫ ОКРУГЛИТЬ 32-Й БИТ, ЗАПИСАВ 1 ВМЕСТО 0), А ЭКСПОНЕНТА РАВНА -3.

ТЕПЕРЬ, ПРИМЕНИВ НАШИ ТРИ ПРАВИЛА, ЗАПИШЕМ 5 БАЙТ:

АДДД ЗНАК ЧИСЛА
У

0111 11 01 0 — 100 1100 1100 1100 1100 1100 1100 1100

-3+128 МАНТИСА 4/5, ИСКЛЮЧАЯ ЛЕВЫЙ ЗНАКОВЫЙ БИТ

ИМЕЕТСЯ АЛЬТЕРНАТИВНЫЙ СПОСОБ ЗАПИСИ ЦЕЛОГО ЧИСЛА В ИНТЕРВАЛЕ -65535...+65535:

А) ПЕРВЫЙ БАЙТ РАВЕН 0

Б) ВТОРОЙ БАЙТ РАВЕН 0 ДЛЯ ПОЛОЖИТЕЛЬНОГО ЧИСЛА И FFH ДЛЯ ОТРИЦАТЕЛЬНОГО

В) 3, 4 БАЙТЫ СОДЕРЖАТ МЛАДШИЕ И СТАРШИЕ ЗНАЧАНИЕ БИТЫ ЧИСЛА (ИЛИ ЧИСЛО +131072, ЕСЛИ ОНО ОТРИЦАТЕЛЬНОЕ)

Г) 5 БАЙТ РАВЕН 0.

ГЛАВА 25
СИСТЕМНЫЕ ПЕРЕМЕННЫЕ
=====

БАЙТЫ ПАМЯТИ С 23532 ДО 23733 ПРЕДНАЗНАЧЕНЫ ДЛЯ СПЕЦИАЛЬНОГО ИСПОЛЬЗОВАНИЯ. В НИХ РАЗМЕЩАЮТСЯ, ТАК НАЗЫВАЕМЫЕ СИСТЕМНЫЕ ПЕРЕМЕННЫЕ. НЕ НАДО ПУТАТЬ ИХ ИМЕНА С ИМЕНАМИ ПЕРЕМЕННЫХ В ПРОГРАММЕ. КОМПЬЮТЕР НЕ РАСПОЗНАЕТ ССЫЛКИ К ЭТИМ ПЕРЕМЕННЫМ ИЗ БЕГСИК-ПРОГРАММЫ ПО ИХ ИМЕНАМ. ИМЕНА ИСПОЛЬЗУЮТСЯ ТОЛЬКО ДЛЯ МНЕМОНИЧЕСКОГО ОБОЗНАЧЕНИЯ ЭТИХ ПЕРЕМЕННЫХ В ЭТОМ ОПИСАНИИ.

ИНФОРМАЦИЯ, ЗАПИСАННАЯ В ПЕРВОМ СТОЛБЦЕ ТАБЛИЦЫ, ИМЕЕТ СЛЕДУЮЩЕЕ ЗНАЧЕНИЕ:

X - ПЕРЕМЕННАЯ,

N - ИЗМЕНЕНИЕ ПЕРЕМЕННОЙ НЕ ПРИВОДИТ К ДЛИТЕЛЬНОМУ ЭФФЕКТУ.

ЧИСЛО - ЧИСЛО БАЙТОВ В ПЕРЕМЕННОЙ (ДЛЯ ДВУХБАЙТОВЫХ ПЕРЕМЕННЫХ МЛАДШИЙ БАЙТ ПЕРВЫЙ).

НАПРИМЕР, НЕОБХОДИМО ИЗМЕНИТЬ ЗНАЧЕНИЕ НА V В ДВУХБАЙТОВОЙ ПЕРЕМЕННОЙ ПО АДРЕСУ N:

10 POKE N,V - 256*INT(V/256)

20 POKE N+1,INT(V/256)

ДЛЯ ПРОСМОТРА НОВОГО ЗНАЧЕНИЯ МОЖНО ИСПОЛЬЗОВАТЬ ОПЕРАТОР;

PEEK N+256*PEEK(N+1)

ЗН	АДРЕС	ИМЯ	СОДЕРЖАНИЕ
N8	23552	KSTATE	ИСПОЛЬЗУЕТСЯ ПРИ ЧТЕНИИ С КЛАВИАТУРЫ
N1	23560	LAST K	ЗАПОМИНАЕТСЯ ВНОВЬ НАЖАТАЯ КЛАВИША
1	23561	REPERL	ВРЕМЯ, В 50-Х ДОЛЯХ СЕКУНДЫ, В ТЕЧЕНИЕ КОТОРОГО КЛАВИША ДОЛЖНА БЫТЬ ЗАФИКСИРОВАНА В НАЖАТОМ СОСТОЯНИИ, НАЧАЛЬНОЕ ЗНАЧЕНИЕ 35, НО МОЖЕТ БЫТЬ ИЗМЕНЕНО.
1	23562	REPPER	ЗАДЕРЖКА В 50-Х ДОЛЯХ СЕКУНДЫ, МЕЖДУ ПОСЛЕДОВАТЕЛЬНЫМИ ОПРОСАМИ КЛАВИШ, НАЧАЛЬНОЕ ЗНАЧЕНИЕ 5
N2	23563	DEFADD	АДРЕС АРГУМЕНТОВ ФУНКЦИИ ПОЛЬЗОВАТЕЛЯ ЕСЛИ ОНИ ИСПОЛЬЗУЮТСЯ, ИНАЧЕ 0.
N1	23565	K DATA	ВТОРОЙ БАЙТ УПРАВЛЕНИЯ ЦВЕТОМ С КЛАВИАТУРЫ.
N2	23566	TVDATA	БАЙТЫ ЦВЕТА, АТ, ТАВ УПРАВЛЕНИЯ ТЕЛЕВИ-
ЗОРМ, X38	23568	STRMS	АДРЕСА ПОДКЛЮЧЕННЫХ КАНАЛОВ.
2	23606	CHARS	АДРЕС СИМВОЛЬНОГО НАБОРА -256, ОБЫЧНО ЭТОТ НАБОР НАХОДИТСЯ В ПЗУ, НО МОЖЕТ БЫТЬ РАЗМЕЩЕН И В ОЗУ С УКАЗАНИЕМ CHARS АДРЕСА РАЗМЕЩЕНИЯ:
1	23608	RASP	ПРОДОЛЖИТЕЛЬНОСТЬ ЗВУКОВОГО СИГНАЛА
1	23609	PIR	ДЛИТЕЛЬНОСТЬ ЗАДЕРЖКИ УСТРАНЯЮЩЕЙ ДРЕБЕЗГ КЛАВИАТУРЫ.

1	23610	ERR NR	КОД СООБЩЕНИЯ -1, НАЧАЛЬНОЕ ЗНАЧЕНИЕ 255 (ДЛЯ '-1'), Т.Е. РЕЕК 23610=255
X1	23611	FLAGS	УПРАВЛЯЮЩИЕ ФЛАЖКИ БЕЙСИКА
X1	23612	TV FLAG	ФЛАЖОК ТЕЛЕВИЗОРА
X2	23613	ERR SP	АДРЕС В АППАРАТНОМ СТЕКЕ, ИСПОЛЬЗУЕМЫЙ КАК АДРЕС ВОЗВРАТА ПРИ ОШИБКЕ.
N2	23615	LIST SF	АДРЕС ВОЗВРАТА ИЗ АВТОМАТИЧЕСКОГО ЛИСТИНГА.
N1	23617	MODE	РЕЖИМ, СПЕЦИФИКАЦИЯ (K), (L), (C), (E) ИЛИ (G) КУРСОРА.
2	23618	NEW PPC	НОМЕР СТРОКИ, НА КОТОРУЮ ДОЛЖЕН БЫТЬ СДЕЛАН ПЕРЕХОД.
2	23621	PPC	НОМЕР СТРОКИ, ОПЕРАТОР В КОТОРОЙ ВЫПОЛНЯЕТСЯ.
1	23623	SUB PPC	ПОРЯДКОВЫЙ НОМЕР ВЫПОЛНЯЮЩЕГОСЯ ОПЕРАТОРА В СТРОКЕ.
1	23624	DORDCR	ЦВЕТ РАМКИ ЭКРАНА, СОДЕРЖИТ АТТРИБУТЫ
2	23625	KEPPC	КОЛИЧЕСТВО ТЕКУЩИХ СТРОК КУРСОРОВ
X2	23627	VAR\$	! АДРЕСА ПЕРЕМЕННЫХ.
N2	23629	DEST	! АДРЕС ПЕРЕМЕННОЙ В ЗАДАНИИ
X2	23631	CHANS	АДРЕС КАНАЛА ДАННЫХ
X2	23633	CURGRL	АДРЕС ДАННЫХ ДЛЯ ВВОДА-ВЫВОДА
X2	23635	PROG	АДРЕС БЕЙСИК ПРОГРАММЫ.
X2	23637	NXTLIN	АДРЕС СЛЕДУЮЩЕЙ СТРОКИ В ПРОГРАММЕ
X2	23639	DATADD	АДРЕС ТЕРМИНАТОРА ПОСЛЕДНЕГО СИМВОЛА В DATA.
X2	23641	E LINE	АДРЕС ВЫВЕДЕННОЙ КОМАНДЫ.
2	23643	K CUR	АДРЕС КУРСОРА.
X2	23645	CH ADD	АДРЕС СЛЕДУЮЩЕГО ИНТЕРПРЕТИРУЕМОГО СИМВОЛА: СИМВОЛ АРГУМЕНТА В РЕЕК, NEWLINE ИЛИ РОКЕ ОПЕРАТОРАХ.
2	23647	X PRI	АДРЕС СИМВОЛА СЛЕДУЮЩЕГО ЗА МАРКЕРОМ (?)
X2	23649	WORR SP	АДРЕС ВРЕМЕННОЙ РАБОЧЕЙ ОБЛАСТИ

X2	23651	STK SOT	АДРЕС "ДНА" ПРОГРАМИРУЕМОГО СТЕКА
X2	23653	STK END	АДРЕС НАЧАЛА РЕЗЕРВНОЙ ОБЛАСТИ ПАМЯТИ
N1	23655	BREG	В -РЕГИСТР КАЛЬКУЛЯТОРА
N2	23656	MEM	АДРЕС ОБЛАСТИ, ИСПОЛЬЗУЕМОЙ КАК ПА- МЯТЬ КАЛЬКУЛЯТОРА (ОБЫЧНО МЕМВОТ, НО НЕ ВСЕГДА).
1	23658	FLAG S2	СТАРШИЕ ФЛАЖКИ
M1	23659	DF SZ	ЧИСЛО СТРОК (ВКЛЮЧАЯ И ОДНУ ЧИСТУЮ) В НИЖНЕЙ ЧАСТИ ЭКРАНА
2	23660	S TOP	КОЛИЧЕСТВО ВЕРХНИХ СТРОК В ПРОГРАММЕ В АВТОМАТИЧЕСКОМ ЛИСТИНГЕ.
2	23662	OLDPPC	НОМЕР СТРОКИ НА КОТОРУЮ УКАЗЫВАЕТ CONTINUE.
1	23664	OSPPC	НОМЕР ОПЕРАТОРА В СТРОКЕ НА КОТОРУЮ УКАЗЫВАЕТ CONTINUE.
N1	23665	FLAGX	ПЕРЕМЕННЫЕ ФЛАЖКИ
N2	23666	STR LEN	РАЗМЕР РАССТОЯНИЯ МЕЖДУ СТРОКАМИ
N2	23668	T ADDR	АДРЕС СЛЕДУЮЩЕГО СИМВОЛА В СИНТАКСИ- ЧЕСКОЙ ТАБЛИЦЕ.
2	23670	SEED	НАЧАЛЬНОЕ ЗНАЧЕНИЕ ДЛЯ RND, ИЗМЕНЯЕТСЯ ФУНКЦИЕЙ RANDOMIZE
3	23672	FRAMES	СЧЕТЧИК КАДРОВ - ПРИРАЩЕНИЕ ЧЕРЕЗ КАЖДЫЕ 20 MS (СМ. ГЛАВУ 18).
2	23675	UDG	АДРЕС ПЕРВОГО ОПРЕДЕЛЯЕМОГО ПОЛЬЗОВАТЕЛЕМ СИМВОЛА.
1	23677 23678	COORDS.	X -КООРДИНАТА ТОЧКИ ГРАФОПОСТРОИТЕЛЯ Y -КООРДИНАТА ТОЧКИ ГРАФОПОСТРОИТЕЛЯ
1	23679	P POSN	33-ПОЗИЦИОННОЕ ЧИСЛО ДЛЯ ПОЗИЦИОНИР- ОВАНИЯ ПРИНТЕРА.
1	23680	PR CC	МЛАДШИЙ БАЙТ АДРЕСА ПОЗИЦИИ ДЛЯ LPRINT ДЛЯ ПЕЧАТИ.
1	23681		НЕ ИСПОЛЬЗУЕТСЯ
2	23682	ECHO E	33-ПОЗИЦИОННОЕ И 24-СТРОКОВОЕ ЧИСЛО (В НИЖНЕЙ ПОЛОВИНЕ) КОНЦА ВХОДНОГО БУФЕРА
2	23684	DF CC	АДРЕС ПРИНТ-ПОЗИЦИИ В ОБЛАСТИ ЭКРАНА

2	23686	DF CCL	ПОДОБНО DF CC ДЛЯ НИЖНЕЙ ЧАСТИ ЭКРАНА
X1	23688	S POSN	33-ПОЗИЦИОННОЕ ЧИСЛО ДЛЯ PRINT ПОЗИЦИИ
X1	23689		23- СТРОКОВОЕ ЧИСЛО ДЛЯ PRINT ПОЗИЦИИ
X2	23690	S POSNL	ПОДОБНО S POSN ДЛЯ НИЖНЕЙ ЧАСТИ
1	23692	SCR CT	СЧЕТЧИК СВЕРТОК: ВСЕГДА НА 1 БОЛЬШЕ ЧИСЛА СВЕРТОК, КОТОРЫЕ ДОЛЖНЫ ПРОВЕДены ПЕРЕД ОСТАНОВОМ СО СВЕРТКОЙ ЕСЛИ ВЫ УСТАНОВИТЕ ЭТО ЧИСЛО БОЛЬШЕ ЧЕМ НА 1 (СКАЖЕМ 255), ТО ЭКРАН БУДЕТ СВОРАЧИВАТЬСЯ БЕЗ ЗАПРОСА К ВАМ.
1	23693	ATTR P	СПЛОШНЫЕ ЦВЕТА
1	23694	MASK P	ИСПОЛЬЗУЕТСЯ ДЛЯ ВЫСВЕЧИВАНИЯ ЦВЕТОВ. БИТ, УСТАНОВЛЕННЫЙ В 1, ПОКАЗЫВАЕТ-БИТЫ АТТРИБУТОВ БЕРУТСЯ НЕ ИЗ ATTR, А ИЗ ТОГО, ЧТО УКАЗАНО НА ЭКРАНЕ.
N1	23695	ATTR T	ВРЕМЕННЫЙ УКАЗАТЕЛЬ ЦВЕТОВ
N1	23696	MASK T	ВРЕМЕННЫЙ MASK P
1	23697	P FLAG	СТАРШИЕ ФЛАЖКИ
N30	23698	MEMBOT	ОБЛАСТЬ ПАМЯТИ ДЛЯ КАЛЬКУЛЯТОРА, ИСПОЛЬЗУЕТСЯ ДЛЯ ЗАПИСИ ЧИСЕЛ, КОТОРЫЕ НЕ МОГУТ БЫТЬ РАЗМЕЩЕНЫ В ПРОГРАМИРУЕМОМ СТЕКЕ КАЛЬКУЛЯТОРА
2	23728		НЕ ИСПОЛЬЗУЕТСЯ
2	23730	RAMTOP	АДРЕС ПОСЛЕДНЕГО БАЙТА ОБЛАСТИ БЕЙСИК-СИСТЕМЫ
2	23732	P-RAMT	АДРЕС ПОСЛЕДНЕГО БАЙТА ФИЗИЧЕСКОГО ОЗУ

СЛЕДУЮЩАЯ ПРОГРАММА ВЫДАСТ ВАМ ПЕРВЫЕ 22 БАЙТА ОБЛАСТИ СИСТЕМНЫХ ПЕРЕМЕННЫХ:

```
10 FOR N=0 TO 21
20 PRINT PEEK (PEEK 23627+256*PEEK 23628+N)
30 NEXT N
```

ТЕПЕРЬ ЗАМЕНИТЕ СТРОКУ 20 НА

```
20 PRINT PEEK(23765+N)
```

И ВЫ ДОПОЛНИТЕЛЬНО ПОЛУЧИТЕ DUMP САМОЙ ПРОГРАММЫ.

ГЛАВ., 26

ИСПОЛЬЗОВАНИЕ МАШИННЫХ КОДОВ

=====

КРАТКОЕ СОДЕРЖАНИЕ: USR С ЧИСЛОВЫМ АРГУМЕНТОМ

ЭТА ГЛАВА ОПИСЫВАЕТ ПРИМЕНЕНИЕ МАШИННЫХ КОМАНД МИКРОПРОЦЕССОРА Z80 (U880 (ГДР), 1810BM80 (СССР)).

ПРОГРАММЫ В МАШИННЫХ КОДАХ ПИШУТ ОБЫЧНО НА АСЕМБЛЕРЕ С ПОСЛЕДУЮЩЕЙ ТРАНСЛЯЦИЕЙ, (СПЕРЕЧЕНЬ МНЕМОКОДОВ КОМАНД МИКРОПРОЦЕССОРА Z80 ПРИВЕДЕН В ПРИЛОЖЕНИИ А). ТРАНСЛЯТОР С АСЕМБЛЕРА ВСТРОЕН В КОМПЬЮТЕР ZX SPECTRUM.

ПРИВЕДЕМ ПРИМЕР ПРОГРАММЫ:

ID BC,99

RET

КОТОРАЯ ЗАГРУЖАЕТ В "BC" РЕГИСТР ЧИСЛО 99, ЭТА ПРОГРАММА БУДЕТ ТРАНСЛИРОВАТЬСЯ В 4-Х БАЙТНЫЙ МАШИННЫЙ КОД:

° БАЙТЫ 1,99,0 ДЛЯ ID BC,99 И

201 ДЛЯ RET

СЛЕДУЮЩИМ ШАГОМ ЯВЛЯЕТСЯ ЗАГРУЗКА ПРОГРАММЫ В КОМПЬЮТЕР, ДЛЯ ЭТОГО ИСПОЛЬЗУЕТСЯ ДОПОЛНИТЕЛЬНАЯ ПАМЯТЬ, ПОЛУЧАЕМАЯ МЕЖДУ БЕЙСИК ОБЛАСТЬЮ И ОБЛАСТЬЮ ОПРЕДЕЛЯЕМЫХ ПОЛЬЗОВАТЕЛЕМ СИМВОЛОВ. ДОПУСТИМ, ВЫ ИМЕЛИ СЛЕДУЮЩЕЕ РАСПРЕДЕЛЕНИЕ ПОСЛЕДНЕЙ ЧАСТИ ОЗУ:

ОПРЕДЕЛЯЕМЫЕ
ПОЛЬЗОВАТЕЛЕМ СИМВОЛЫ

RAMTOP=32599 UDG=32600

PRAMT=32767

ЕСЛИ ВЫ ТЕПЕРЬ ВЫПОЛНИТЕ:

CLEAR 32499

ТО ПОЛУЧИТЕ 100 БАЙТОВ ПАМЯТИ, НАЧИНАЯ С АДРЕСА 32500

100
БАЙТОВ

ОПРЕДЕЛЯЕМЫЕ
ПОЛЬЗОВАТЕЛЕМ СИМВОЛЫ

32500

UDC=32600

FRAMT=32767

RAMTOP=32499

ДЛЯ ЗАГРУЗКИ ПРОГРАММЫ В МАШИННЫХ КОДАХ ВЫ МОЖЕТЕ ВЫПОЛНИТЬ СЛЕДУЮЩУЮ БЕЙСИК-ПРОГРАММУ:

10 LET A=32500

20 READ N :POKE A,N

30 LET A=A+1: GOTO 20

40 DATA 1,99,0,201

(ПРОГРАММА МОЖЕТ ЗАВЕРШИТЬСЯ СООБЩЕНИЕМ 'E OUT OF DATA' ЕСЛИ ПЕРЕПОЛНЯЮТСЯ ОТВЕДЕННЫЕ ВАМИ 4 БАЙТА).

ДЛЯ ВЫПОЛНЕНИЯ ЗАГРУЖЕННЫХ МАШИННЫХ КОДОВ ИСПОЛЬЗУЕТСЯ ФУНКЦИЯ USR, НО С ЧИСЛОВЫМ АРГУМЕНТОМ, ОПРЕДЕЛЯЮЩИМ НАЧАЛЬНЫЙ АДРЕС.

ЕСЛИ ВЫ ВЫПОЛНИТЕ: PRINT USR 32500

ТО ПОЛУЧИТЕ ОТВЕТ 99.

ВОЗВРАТ В БЕЙСИК-ПРОГРАММУ ОСУЩЕСТВЛЯЕТСЯ ОБЫЧНЫМ ОБРАЗОМ ПО КОМАНДЕ МИКРОПРОЦЕССОРА RET. В МАШИННОЙ ПРОГРАММЕ ВЫ НЕ ДОЛЖНЫ ИСПОЛЬЗОВАТЬ РЕГИСТРЫ IY И I.

ВЫ МОЖЕТЕ ЗАПИСАТЬ ВАШУ ПРОГРАММУ НА ЛЕНТУ:

SAVE "NAME" CODE 32500,4

МОЖНО ЗАПИСАТЬ ЭТУ ПРОГРАММУ И ТАК, ЧТО ОНА БУДЕТ АВТОМАТИЧЕСКИ ВЫПОЛНЯЕТСЯ ПОСЛЕ ЗАГРУЗКИ:

10 LOAD " " " " CODE 32500,4

20 PRINT USR 32500

ДЛЯ ЧЕГО НАДО СДЕЛАТЬ:

SAVE NAME LINE,

А ЗАТЕМ: SAVE "XXXX" CODE 32500,4

LOAD "NAME"

ЭТО ПРИВЕДЕТ К ТОМУ, ЧТО ВНАЧАЛЕ БУДЕТ ЗАГРУЖЕНА И АВТОМАТИЧЕСКИ ВЫПОЛНЕНА БЕЙСИК-ПРОГРАММА, КОТОРАЯ, В СВОЮ ОЧЕРЕДЬ, ЗАГРУЗИТ И ВЫПОЛНИТ ПРОГРАММУ В МАШИННЫХ КОДАХ.

КНИГА "ИСКУССТВО СХЕМОТЕХНИКИ" П. ХОРОВИЦ, У. ХИЛЛ
МИР. 1986. ТОМ 2. СТР 579-580

ДАЛЕЕ ПРИВОДЯТСЯ 78 КОМАНД МИКРОПРОЦЕССОРА 8085 СОВМЕСТИМЫХ
МИКРОПРОЦЕССОРОМ 280 (158 КОМАНД), (U880-ГДР, K1810BM-СССР)

МНЕМОНИКА	ДЕЙСТВИЕ	КОП	ЦИКЛ
1	2	3	4
ПЕРЕСЫЛКА, ЗАГРУЗКА, ЗАПИСЬ			
MOV R,R	ПЕРЕСЛАТЬ РЕГИСТР В РЕГИСТР	01RR RRRR	4[7]
MVI R,D	ПЕРЕСЛАТЬ НЕПОСР. В РЕГИСТР	00RR R110	7[10]
LXI RP,DD	ЗАГРУЗИТЬ НЕПОСР. В ДВА РЕГ.	00RP 0001	10
STAX B	ЗАПОМНИТЬ А КОСВЕННО ПО ВС	0000 0010	7
STAX D	ЗАПОМНИТЬ А КОСВЕННО ПО DE	0001 0010	7
LDAX B	ЗАГРУЗИТЬ А КОСВЕННО ПО ВС	0000 1010	7
LDAX D	ЗАГРУЗИТЬ А КОСВЕННО ПО DE	0001 1010	7
STA DD	ЗАПОМНИТЬ А ПО АДРЕСУ DD	0011 0010	13
LDA DD	ЗАГРУЗИТЬ А ПО АДРЕСУ DD	0011 1010	13
SHLD DD	ЗАПОМНИТЬ H,L ПО АДРЕСУ DD	0010 0010	16
SHLD DD	ЗАГРУЗИТЬ H,L ПО АДРЕСУ DD	0010 1010	16
XCHG	ОБМЕНЯТЬ DE И HL	1110 1011	4
ПРИРАЩЕНИЕ И УМЕНЬШЕНИЕ			
INR R	ПРИРАЩЕНИЕ РЕГИСТРА	00RR R100	4[11]
DCR R	УМЕНЬШЕНИЕ РЕГИСТРА	00RR R101	4[11]
INX RP	ПРИРАЩЕНИЕ ПАРЫ РЕГИСТРОВ	00RP 0011	6
DCX RP	УМЕНЬШЕНИЕ ПАРЫ РЕГИСТРОВ	00RP 1011	6
АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ			
ADD R	ПРИБАВИТЬ РЕГИСТР К А	1000 0RRR	4[7]
ADC R	ПРИБАВИТЬ РЕГ. К А С ПЕРЕНОС.	1000 1RRR	4[7]
SUB R	ВЫЧЕСТЬ РЕГИСТР ИЗ А	1001 0RRR	4[7]
SBB R	ВЫЧЕСТЬ С ЗАЕМОМ	1001 1RRR	4[7]
ANA R	РЕГИСТР & А	1010 0RRR	4[7]
XRA R	ИСКЛ. ИЛИ РЕГИСТРА И А	1010 1RRR	4[7]
ORA R	ИЛИ РЕГИСТРА И А	1011 0RRR	4[7]
CMA R	СРАВНИТЬ РЕГИСТР И А	1011 1RRR	4[7]
ADI D	ПРИБАВИТЬ НЕПОСР. ДАННЫЕ И А	1100 0110	7
ACI D	ПРИБАВИТЬ НЕПОСР. С ПЕРЕНОС.	1100 1110	7
SUI D	ВЫЧЕСТЬ НЕПОСР. ИЗ А	1101 0110	7
SBI D	ВЫЧЕСТЬ НЕПОСР. С ЗАЕМОМ	1101 1110	7
ANI D	НЕПОСР. & А	1110 0110	7
XRI D	ИСКЛ. НЕПОСР. И А	1110 1110	7
ORI D	НЕПОСР. ИЛИ А	1111 0110	7
CPI D	СРАВНИТЬ НЕПОСР. С А	1111 1110	7
DAD RP	ПРИБАВИТЬ ПАРУ РЕГ. К HL	00RP 1001	11

ОПЕРАЦИИ С НАКОПИТЕЛЯМИ ФЛАГАМИ

RLC	СДВИНУТЬ А ВЛЕВО	0000 0111	4
RRC	СДВИНУТЬ А ВПРАВО	0000 1111	4
RAL	СДВИНУТЬ А ВЛЕВО ЧР. РАЗР. ПЕР.	0001 0111	4
RAR	СДВ. А ВПРАВО ЧР. РАЗР. ПЕР.	0001 1111	4
DAA	ДЕСЯТИЧН. КОРРЕКЦИЯ НАКОПИТ	0010 0111	4
CMA	ДОПОЛНЕНИЕ К НАКОПИТЕЛЮ	0010 1111	4
STC	УСТАНОВИТЬ БИТ ПЕРЕНОСА	0011 0111	4
CMC	ОБРАТИТЬ БИТ ПЕРЕНОСА	0011 1111	4

I/O УПРАВЛЕНИЕ И ОПЕРАЦИИ СО СТЕКОМ 0

IN D	ВВОД ИЗ ПОРТА D	1101 1011	10
OUT D	ВЫВОД В ПОРТ D	1101 0011	11
EI	РАЗРЕШЕНИЕ ПРЕРЫВАНИЯ	1111 1011	4
DI	ЗАПРЕЩЕНИЕ ПРЕРЫВАНИЯ	1111 0011	4
NOP	НЕТ ОПЕРАЦИИ	0000 0000	4
HLT	ОСТАНОВ	0111 0110	4
PUSH RP	ЗАНЕСТИ ПАРУ РЕГИСТР. В СТЕК.	11PP 0101	11
POP RP	ВЗЯТЬ ПАРУ РЕГИСТР. ИЗ СТЕКА	11PP 0001	10
XTHL	ОБМЕНЯТЬ HL С ВЕРХОМ СТЕКА	1110 0011	19
SPHL	ПЕРЕСЛАТЬ HL В SP	1111 1001	6

ПЕРЕДАЧИ УПРАВЛЕНИЯ

JMP DD	БЕЗУСЛОВНЫЙ ПЕРЕХОД	1100 0011	10
JCC DD	ПЕРЕЙТИ ПО УСЛОВИЮ CC	1100 0010	10
CALL DD	БЕЗУСЛОВНЫЙ ВЫЗОВ	1100 1101	17
CCC DD	ВЫЗОВ ПО УСЛОВИЮ CC	1100 0100	17(10)
RET	ВОЗВРАТ ПОСЛЕ ВЫЗОВА	1100 1001	10
RCC	ВОЗВРАТ ПО УСЛОВИЮ CC	1100 0000	11(5)
RST N	ВОЗВРАЩЕНИЕ К ЯЧЕЙКЕ 8*N	11NN N111	11
PNHL	ПЕРЕСЛАТЬ HL В PC	1110 1001	4

ОБОЗНАЧЕНИЯ:

1. ПОЛЯ ДАННЫХ.

D - ОДИН БАЙТ НЕПОСРЕДСТВЕННЫХ ДАННЫХ (ДЛИНА КОМАНДЫ 2 БАЙТА)
 DD - ДВУБАЙТОВЫЙ АДРЕС (ДЛИНА КОМАНДЫ 3 БАЙТА)
 ВСЕ ОСТАЛЬНЫЕ КОМАНДЫ ИМЕЮТ ДЛИНУ 1 БАЙТ

2. ЦИКЛЫ.

N - ЧИСЛО ТАКТОВ НУЖНОЕ ДЛЯ ВЫПОЛНЕНИЯ КОМАНДЫ,
 (N)-ЧИСЛО ТАКТОВ КОГДА R=M (ДОСТУП В ПАМЯТЬ,)
 (N)-ЧИСЛО ТАКТОВ ЕСЛИ УСЛОВИЕ НЕ ВЫПОЛНЕНО.

3. ПОЛЯ РЕГИСТРОВ.

"R"	RRR		"RP"	PP
B	000		BC	00
C	001		DE	01
D	010		HL	10
E	011		PS	11
H	100		PSW	11
L	101			
M	110	("M"=(HL))		
A	111			

4. КОДЫ УСЛОВИЙ.

"CC"	CCC	УСЛОВИЕ
NZ	000	НЕ НУЛЬ
Z	001	НУЛЬ
NC	010	НЕТ ПЕРЕНОСА
C	011	ПЕРЕНОС
PO	100	НЕЧЕТНЫЙ ПАРИТЕТ
PE	101	ЧЕТНЫЙ ПАРИТЕТ
P	110	ПОЛОЖИТЕЛЬНОЕ
M	111	ОТРИЦАТЕЛЬНОЕ

ПРИЛОЖЕНИЕ А

ПОЛНЫЙ НАБОР СИМВОЛОВ

=====

ДЕС КОД	СИМВОЛ	ШН. КОД	АССЕМБЛЕР МНЕМОНИКА	СВН...	ЕДН...
1	2	3	4	5	
0	НЕ ИСПОЛЬЗ.	00	NOP	RLC B	
1	НЕ ИСПОЛЬЗ.	01	LD BC, NN	RLC C	
2	НЕ ИСПОЛЬЗ.	02	LD (BC), A	RLC D	
3	НЕ ИСПОЛЬЗ.	03	INC BC	RLC E	
4	НЕ ИСПОЛЬЗ.	04	INC B	RLC H	
5	НЕ ИСПОЛЬЗ.	05	DEC B	RLC L	
6	PRINT УПР.	06	LD B, N	RLC(HL)	
7	EDIT	07	RLCA	RLC A	
8	КУРС. ВЛЕВО	08	EX AF, AF"	RRC B	
9	КУРС. ВПРАВО	09	ADD HL, BC	RRC E	
10	КУРС ВНИЗ	0A	LD A, (BC)	RRC D	
11	КУРСОР ВВЕРХ	0B	DEC BC	RRC E	
12	DELETE	0C	INC C	RRC H	
13	ENTER	0D	DEC C	RRC L	
14	ЧИСЛО	0E	LD C, N	RRC(HL)	
15	НЕ ИСПОЛЬЗ.	0F	RRCA	RRC A	
16	INC УПР.	10	DJNZ DIS	RL B	

17	PAPER УПР.	11	LD DE, NN	RL C	
18	FLASH УПР.	12	LD (DE), A	RL D	
19	BRIGHT УПР.	13	INC DE	RL E	
20	INVERSE УПР.	14	INC D	RL H	
21	OVER УПР.	15	DEC D	RL L	
22	AT УПР.	16	LD D, M	RL (HL)	
23	TAB УПР.	17	RLA	RL A	
24	НЕ ИСПОЛЬЗ.	18	JR DIS	RR B	
25	НЕ ИСПОЛЬЗ.	19	ADD HL, DE	RR C	
26	НЕ ИСПОЛЬЗ.	1A	LD A, (DE)	RR D	
27	НЕ ИСПОЛЬЗ.	1B	DEC DE	RR E	
28	НЕ ИСПОЛЬЗ.	1C	INC E	RR H	
29	НЕ ИСПОЛЬЗ.	1D	DEC E	RR L	
30	НЕ ИСПОЛЬЗ.	1E	LD E, M	RR (HL)	
31	НЕ ИСПОЛЬЗ.	1F	RRA	RR A	
32	ПРОБЕЛ	20	JR NZ, DIS	SLA B	
33		21	LD HL, NN	SLA C	
34	"	22	LD (NN), HL	SLA C	
35	#	23	INC HL	SLA E	
36	x	24	INC H	SLA H	
37	z	25	DEC H	SLA L	
38	&	26	LD H, M	SLA (HL)	
39	'	27	DAA	SLA A	
40	(	28	JR Z, DIS	SRA B	
41	)	29	ADD HL, HL	SRA C	
42	*	2A	LD HL, NN	SRA D	
43	+	2B	DEC HL	SRA E	
44	,	2C	INC L	SRA H	
45	-	2D	DEC L	SRA E	
46	???????	2E	LD L, M	SRA (HL)	
47	/	2F	CPL	SRA A	
48	0	30	JR NC, DIS		
49	1	31	LD SP, NN		
50	2	32	LD (NN), A		
51	3	33	INC SP		
52	4	34	INC (HL)		
53	5	35	DEC (HL)		
54	6	36	LD (HL), N		
55	7	37	SCF		
56	8	38	JR C, DIS	SRL B	
57	9	39	ADD HL, SP	SRL C	
58	:	3A	LD A, (NN)	SRL D	
59		3B	DEC SP	SRL E	
60	<	3C	INC A	SZL H	
61	=	3D	DEC A	SRL L	
62	>	3E	LD A, N	SRL (HL)	
63	?	3F	CCF	SRL A	
64	@	40	LD B, B	BIT 0, B	IN B, (C)
65	A	41	LD B, C	BIT 0, C	OUT (C), B
66	B	42	LD B, D	BIT 0, D	SBC HL, BC
67	C	43	LD B, E	BIT 0, E	LD (NN), BC
68	D	44	LD B, H	BIT 0, H	NEG
69	E	45	LD B, L	BIT 0, L	RETN
70	F	46	LD B, (HL)	BIT 0, (HL)	IM 0
71	G	47	LD B, A	BIT 0, A	LD L, A
72	H	48	LD C, B	BIT 1, B	IN C, (C)
73	I	46	LD C, C	BIT 1, C	OUT C, (C)

74	J	4A	LD C,D	BIT 1,D	ADD HL,BC
75	K	4B	LD C,E	BIT 1,E	LD BC,(NN)
76	L	4C	LD C,H	BIT 1,H	
77	M	4D	LD C,L	BIT 1,L	RET I
78	N	4E	LD C,(HL)	BIT 1,(HL)	
79	O	4F	LD C,A	BIT 1,A	LD R,A
80	P	50	LD D,B	BIT 2,B	IN D,(C)
81	Q	51	LD D,C	BIT 2,C	OUT (C)
82	R	52	LD D,D	BIT 2,D	SBC HL,D
83	S	53	LD D,E	BIT 2,E	LD (NN)
84	T	54	LD D,H	BIT 2,H	
85	U	55	LD D,L	BIT 2,L	
86	V	56	LD D,(HL)	BIT 2,(HL)	IM 1
87	W	57	LD D,A	BIT 2,A	LD A,L
88	X	58	LD E,B	BIT 3,B	IN E,(C)
89	Y	59	LD E,C	BIT 3,C	OUT (C)
90	Z	5A	LD E,D	BIT 3,D	ADC RL,D
91	[	5B	LD DE,(N)	BIT 3,E	
92	/	5C	LD E,H	BIT 3,H	
93	]	5D	LD E,L	BIT 3,L	
94	СТРЕЛКА ВРХ.	5E	LD E,(HL)	BIT 3(HL)	IM 2
95	-	5F	LD E,A	BIT 3,A	LD A,R
96	ФУНТ-СТЕРЛ.	60	LD H,B	BIT 4,B	IN H,(C)
97	A(СТРОЧН)	61	LD H,C	BIT 4,C	OUT (C)
98	B(СТРОЧН)	62	LD H,D	BIT 4,D	SBC HL,N
99	C(СТРОЧН)	63	LD H,E	BIT 4,E	LD (NN)
100	D(СТРОЧН)	64	LD H,H	BIT 4,H	
101	E(СТРОЧН)	65	LD H,L	BIT 4,L	
102	F(СТРОЧН)	66	LD H,(HL)	BIT 4(HL)	
103	G(СТРОЧН)	67	LD H,A	BIT 4,A	RRD
104	H(СТРОЧН)	68	LD L,B	BIT 5,B	IN L,(C)
105	I(СТРОЧН)	69	LD L,C	BIT 5,C	OUT(C),L
106	J(СТРОЧН)	6A	LD L,D	BIT 5,D	ADC HL,R
107	K(СТРОЧН)	6B	LD L,E	BIT 5,E	LD HL,(N)
108	L(СТРОЧН)	6C	LD L,H	BIT 5,H	
109	M(СТРОЧН)	6D	LD L,L	BIT 5,L	
110	N(СТРОЧН)	6E	LD L,(HL)	BIT 5,(HL)	
111	O(СТРОЧН)	6F	LD L,A	BIT 5,A	RID
112	P(СТРОЧН)	70	LD (HL),B	BIT 6,B	IN F,(C)
113	Q(СТРОЧН)	71	LD (HL),C	BIT 6,C	
114	R(СТРОЧН)	72	LD (HL),D	BIT 6,D	SBC HL,S
115	S(СТРОЧН)	73	LD (HL),E	BIT 6,E	LD(NN),S
116	T(СТРОЧН)	74	LD (HL),H	BIT 6,H	
117	V(СТРОЧН)	75	LD (HL),L	BIT 6,L	
118	U(СТРОЧН)	76	HALT	BIT 6,(HL)	
119	W(СТРОЧН)	77	LD (HL),A	BIT 6,A	
120	X(СТРОЧН)	78	LD A,B	BIT 7,B	IN A,(C)
121	Y(СТРОЧН)	79	LD A,C	BIT 7,C	OUT (C)
122	Z(СТРОЧН)	7A	LD A,D	BIT 7,D	ADC HL,S
123	ФИГУР.СК.ЛЕВ.	7B	LD A,E	BIT 7,E	LD SP,(R)
124	ВЕРТ. ЧЕРТА	7C	LD A,H	BIT 7,H	
125	ФИГУР.СК.ПРАВ	7D	LD A,L	BIT 7,L	
126	ДЕФИС	7E	LD A,(HL)	BIT 7,H	
127	С В ОКР.	7F	LD A,A	BIT 7,A	
128	0 0	80	ADD A,B	RES 0,B	
	0 0				

129	O X	81	ADD A,C	RES 0,C	
	O O				
130	X O	82	ADD A,D	RES 0,D	
	O O				
131	X O	83	ADD A,E	RES 0,E	
	O O				
132	O O	84	ADD A,H	RES 0,H	
	O X				
133	O X	85	ADD A,L	RES 0,L	
	O X				
134	X O	86	ADD A,HI	RES 0,(HL)	
	O X				
135	X X	87	ADD A,A	RES 0,A	
	O X				
136	O O	88	ADC A,B	RES 1,B	
	X O				
137	O X	89	ADC A,C	RES 1,C	
	X O				
138	X O	8A	ADC A,D	RES 1,D	
	X O				
139	X X	8B	ADC A,E	RES 1,E	
	X O				
140	O O	8C	ADC A,H	RES 1,H	
	X X				
141	O X	8D	ADC A,L	RES 1,L	
	X X				
142	X O	8E	ADC A,(HL)	RES 1,(HL)	
	X X				
143	X X	8F	ADC A,A	RES 1,A	
	X X				
144	(A)	90	SUB B	RES 2,B	
145	(B)	91	SUB C	RES 2,C	
146	(C)	92	SUB D	RES 2,D	
147	(D)	93	SUB E	RES 2,E	
148	(E)	94	SUB H	RES 2,H	
149	(F)	95	SUB L	RES 2,L	
150	(G)	96	SUB (HL)	RES 2,(HL)	
151	(H)	97	SUB A	RES 2,A	
152	(I)	98	SBC A,B	RES 3,B	
153	(J)	99	SBC A,C	RES 3,C	
154	(K)	9A	SBC A,D	RES 3,D	
155	(L)	9B	SBC A,E	RES 3,E	
156	(M)	9C	SBC A,H	RES 3,H	
157	(N)	9D	SBC A,L	RES 3,L	
158	(O)	9E	SBC A,(HL)	RES 3,(HL)	
159	(P)	9F	SBC A,A	RES 3,A	
160	(Q)	A0	AND B	RES 4,B	LDI
161	(R)	A1	AND C	RES 4,C	CPI
162	(S)	A2	AND D	RES 4,D	INI
163	(T)	A3	AND E	RES 4,E	OUT I
164	(U)	A4	AND H	RES 4,H	
165	RND	A5	AND L	RES 4,L	
166	INKEY*	A6	AND (HL)	RES 4,(HL)	
167	PI	A7	AND A	RES 4,A	
168	FN	A8	XOR B	RES 5,B	LDD
169	POINT	A9	XOR C	RES 5,C	CPD

170	SCREEN*	AA	XOR D	RES 5,D	IND
171	ATTR	AB	XOR E	RES 5,E	OUTD
172	AT	AC	XOR H	RES 5,H	
173	TAB	AD	XOR L	RES 5,L	
174	VAL*	AE	XOR (HL)	RES 5,(HL)	
175	CODE	AF	XOR A	RES 5,A	
176	VAL	B0	OR B	RES 6,B	LDIR
177	LEN	B1	OR C	RES 6,C	CPIR
178	SIN	B2	OR D	RES 6,D	INIR
179	COS	B3	OR E	RES 6,E	OTIR
180	TAN	B4	OR H	RES 6,H	
181	ASN	B5	OR L	RES 6,L	
182	ACS	B6	OR (HL)	RES 6,(HL)	
183	ATN	B7	OR A	RES 6,A	
184	LN	B8	CP B	RES 7,B	LDDR
185	EXP	B9	CP C	RES 7,C	CPDR
186	INT	B9	CP D	RES 7,D	INDR
187	SQR	BA	CP E	RES 7,E	OTDR
188	SGN	BB	CP H	RES 7,H	
189	ABS	BD	CP L	RES 7,L	
190	PEEK	BE	CP (HL)	RES 7,(HL)	
191	IN	BF	CP A	RES 7,A	
192	USR	C0	RET NZ	SET 0,B	
193	STR*	C1	POP BC	SET 0,C	
194	CHR*	C2	JP NZ,NN	SET 0,D	
195	NOT	C3	JP NN	SET 0,E	
196	BIN	C4	CALL NZ,NN	SET 0,H	
197	OR	C5	PUSH BC	SET 0,L	
198	AND	C6	ADD A,N	SET 0,(HL)	
199	<=	C7	RST 0	SET 0,A	
200	>=	C8	RET Z	SET 1,B	
201	<>	C9	RET	SET 1,C	
202	LINE	CA	JP Z,NN	SET 1,D	
203	THEN	CB		SET 1,E	
204	TO	CC	CALL Z,NN	SET 1,H	
205	STEP	CD	CALL NN	SET 1,L	
206	DEF FN	CE	ADC A,N	SET 1,(HL)	
207	CAT	CF	RST 8	SET 1,A	
208	FORMAT	D0	RET NC	SET 2,B	
209	MOVE	D1	POP DE	SET 2,C	
210	ERASE	D2	JP NC,NN	SET 2,D	
211	OPEN#	D3	OUT (N),A	SET 2,E	
212	CLOSE#	D4	CALL NC,NN	SET 2,H	
213	MERGE	D5	PUSH DE	SET 2,L	
214	VERIFY	D6	SUB N	SET 2,(HL)	
215	BEEP	D7	RST 16	SET 2,A	
216	CIRCLE	D8	RET C	SET 3,B	
217	INK	D9	EXX	SET 3,C	
218	PAPER	DA	JP C,NN	SET 3,D	
219	FLASH	DB	IN A,(N)	SET 3,E	
220	BRIGHT	DC	CALL C,NN	SET 3,H	
221	INVERSE	DD	PREFIXES IN- STRUCTIONS USING IX	SET 3,L	
222	OVER	DE	SBC A,N	SET 3,(HL)	

223	OUT	DF	RST 24	SET 3,A
224	LPRINT	E0	RET FO	SET 4,B
225	LLIST	E1	POP HL	SET 4,C
226	STOP	E2	JP PC,NN	SET 4,D
227	READ	E3	EX (SP),HL	SET 4,E
228	DATA	E4	CALL PO,NN	SET 4,H
229	RESTORE	E5	PUSH HL	SET 4,L
230	NEW	E6	ANU N	SET 4,(HL)
231	BORDER	E7	RST 32	SET 4,A
232	CONTINUE	E8	RET PE	SET 5,B
233	DIM	E9	JP (HL)	SET 5,C
234	REM	EA	JP PE,NN	SET 5,D
235	FOR	EB	EX DE,HL	SET 5,E
236	GO TO	EC	CALL PE,NN	SET 5,H
237	GO SUB	ED		SET 5,L
238	INPUT	EE	XOR N	SET 5,(HL)
239	LOAD	EF	RST 40	SET 5,A
240	LIST	F0	RET P	SET 6,B
241	LET	F1	POP AF	SET 6,C
242	PAUSE	F2	JP P,NN	SET 6,D
243	NEXT	F3	DI	SET 6,E
244	POKE	F4	CALL P,NN	SET 6,H
245	PRINT	F5	PUSH AF	SET 6,L
246	PLOT	F6	OR N	SET 6,(HL)
247	RUN	F7	RST 48	SET 6,A
248	SAVE	F8	RET M	SET 7,B
249	RANDOMIZE	F9	LD SP,HL	SET 7,C
250	IF	FA	JP M,NN	SET 7,D
251	CLS	FB	EI	SET 7,E
252	DRAW	FC	CALL M,NN	SET 7,H
253	CLEAR	FD	PREFIXES	SET 7,L
			INSTRUCTIONS	
			USING IY	
254	RETURN	FE	CP N	SET 7,(HL)
255	COPY	FF	RST 56	SET 7,A

ПРИЛОЖЕНИЕ В

СООБЩЕНИЯ

=====

Они появляются в нижней части экрана, если компьютер остановился при выполнении некоторого оператора Бейсика, и указывают причину вызвавшую останов.

Сообщение содержит кодовый номер или букву, краткое содержание помогает найти ошибочную строку и ошибочный оператор этой строке, (команда указывается как строка 0, оператор 1 располагается в строке первым, оператор 2 следует после первого или then и т.д.)

От состояния continue зависит очень многое в сообщениях, обычно продолжение начинается с оператора, специфицированного предыдущем сообщении, но имеются исключения - сообщения 0, 9 и D, (смотри также приложение C)

КОД	ЗНАЧЕНИЕ	СИТУАЦИЯ
1	2	3
0	OK (О' КЕЙ! ПОРЯДОК)	РАЗНОЕ
	УСПЕШНОЕ ЗАВЕРШЕНИЕ ИЛИ ПЕРЕХОД НА СТРОКУ С НОМЕРОМ БОЛЬШИМ ЧЕМ ИМЕЕТСЯ ВСЕГО. ЭТО СООБЩЕНИЕ НЕ ИМЕЕТ СТРОКИ ИЛИ ОПЕРАТОРА ОПРЕДЕЛЕННОГО ДЛЯ CONTINUE	
1	NEXT WITHOUT FOR (NEXT БЕЗ FOR)	NEXT
	УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ НЕТ (НЕ БЫЛА ОПРЕДЕЛЕНА В ОПЕРАТОРЕ FOR), НО ЕСТЬ ОБЫЧНАЯ ПЕРЕМЕННАЯ С ТЕМ ЖЕ ИМЕНЕМ.	
2	VARIABLE NOT FOUND (ПЕРЕМЕННАЯ НЕ НАЙДЕНА)	РАЗНОЕ
	ДЛЯ ПРОСТОЙ ПЕРЕМЕННОЙ ВЫДАЕТСЯ, ЕСЛИ ОНА ИСПОЛЬЗУЕТСЯ БЕЗ ПРЕДВАРИТЕЛЬНОГО ОПРЕДЕЛЕНИЯ В ОПЕРАТОРАХ LET, READ ИЛИ INPUT, ИЛИ ЗАГРУЖАЕТСЯ С ЛЕНТЫ ИЛИ УСТАНАВЛИВАЕТСЯ В ОПЕРАТОРЕ FOR. ДЛЯ ИНДЕКСИРУЕМОЙ ПЕРЕМЕННОЙ, СООБЩЕНИЕ ВЫДАЕТСЯ, ЕСЛИ ОНА НЕ БЫЛА ПРЕДВАРИТЕЛЬНО ОПРЕДЕЛЕНА В ОПЕРАТОРЕ DIM ПЕРЕД ИСПОЛЬЗОВАНИЕМ ИЛИ ЗАГРУЗКОЙ С ЛЕНТЫ.	
3	SUBSCRIPT WRONG (ОШИБОЧНЫЙ ИНДЕКС)	В ИНДЕКСНОЙ ПЕРЕМЕННОЙ ИЛИ ПОДСТРОКЕ
	ИНДЕКС ПРЕВЫШАЕТ РАЗМЕРНОСТЬ МАССИВА, ЛИБО ОШИБОЧНОЕ ЧИСЛО ЗАДАЕТ ИНДЕКС, ЕСЛИ ИНДЕКС ОТРИЦАТЕЛЬНЫЙ ИЛИ БОЛЬШЕ 65536, ТО ВЫДАЕТСЯ СООБЩЕНИЕ В.	
4	OUT OF MEMORU (ВНЕ ПАМЯТИ)	LET INPUT FOR DIM GO SUB LOAD MERGE
	В ПАМЯТИ НЕДОСТАТОЧНО МЕСТА ДЛЯ ВАШИХ ДЕЙСТВИЙ, ВЫ МОЖЕТЕ ОСВОБОДИТЬ СЕБЕ ПАМЯТЬ УДАЛИВ КОМАНДНЫЕ СТРОКИ, ИСПОЛЬЗУЯ DELETE ЗАТЕМ УДАЛИТЬ ОДНУ ИЛИ ДВЕ СТРОКИ ПРОГРАММЫ (С ЦЕЛЬЮ ВОЗВРАТА ИХ В ПОСЛЕДСТВИИ), ПОЛУЧИТЬ ДОПОЛНИТЕЛЬНУЮ ПАМЯТЬ МАНЕВРИРУЯ ОПЕРАТОРОМ CLEAR.	
5	OUT OF SCREEN (ВНЕ ЭКРАНА)	INPUT PRINT
	ЕСЛИ INPUT ОПЕРАТОР ГЕНЕРИРУЕТ БОЛЬШЕ ЧЕМ 23 СТРОКИ В НИЖНЕЙ ПОЛОВИНЕ ЭКРАНА, ТАКЖЕ ВСТРЕЧАЕТСЯ С PRINT AT 22,.....	
6	NUMBER TOO BIG (ЧИСЛО БОЛЬШЕ МАКСИМАЛЬНО ДОПУСТИМОГО) В РЕЗУЛЬТАТЕ ВЫЧИСЛЕНИЯ ПОЛУЧИЛОСЬ ЧИСЛО БОЛЬШЕ 10**36.	АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ

7	RETURN WITHOUT GO SUB (RETURN БЕЗ GO SUB) ВСТРЕТИЛОСЬ БОЛЬШЕ ОПЕРАТОРОВ RETURN, ЧЕМ БЫЛО ОПЕРАТОРОВ GO SUB	RETURN
8	END OF FILE (КОНЕЦ ФАЙЛА)	ОПЕРАЦИИ С ВНЕШНЕЙ ПАМЯТЬЮ
9	STOP STATEMENT (ОПЕРАТОР STOP) ПОСЛЕ ЭТОГО СООБЩЕНИЯ CONTINUE НЕ МОЖЕТ ПОВТОРИТЬ STOP, НО МОЖЕТ ПЕРЕДАТЬ УПРАВЛЕНИЕ НА СЛЕДУЮЩИЙ ОПЕРАТОР.	STOP
A	INVALID ARGUMENT (ОШИБОЧНЫЙ АРГУМЕНТ) АРГУМЕНТ ФУНКЦИИ НЕ ДОПУСТИМ В ДАННОЙ ВЕРСИИ	SQR ON ASN ACS USR СТРОКОВЫЙ АРГУМЕНТ
B	INTEGER OUT OF RANGE (ПЕРЕПОЛНЕНИЕ ЦЕЛОГО) ВЫДАТСЯ, КОГДА АРГУМЕНТ С ПЛАВАЮЩЕЙ ТОЧКОЙ ОКРУГЛЯЕТСЯ К ЦЕЛОМУ. ДЛЯ СЛУЧАЯ МАССИВОВ СМОТРИ ТАКЖЕ СООБЩЕНИЕ 3.	RUN RANDOM POKE DIM GO TO GO SUB LIST LLIST PAUSE PLOT CHR* PEEK USR(С ЧИСЛОВЫМ АРГУМЕНТОМ)
C	NONSENSE IN BASIC (ВЫРАЖЕНИЕ НЕ БЕЙСИКА) ТЕКСТ (СТРОКА) НЕ РАСПОЗНАЕТСЯ БЕЙСИКОМ КАК ДОПУСТИМОЕ ВЫРАЖЕНИЕ.	VAL VAL*
D	BREAK-CONT REPEATS КЛАВИША BREAK НАЖАТА ВО ВРЕМЯ ДЕЙСТВИЯ ПЕРИ- ФЕРИЙНОЙ ОПЕРАЦИИ. ДЕЙСТВИЯ CONTINUE ПОСЛЕ ЭТОГО ОПЕРАТОРА ОБЫЧНЫЕ, ТЕ ЧТО УКАЗАНЫ В ОПЕРАТОРЕ, СРАВНИТЕ С СООБЩЕНИЕМ L.	LOAD SAVE VERIFY MERGE LPRINT LLIST COPY(ТОЛЬ- КО КОГДА КОМПЬЮТЕР ЗАПРОСИЛ СВЕРТКУ, А ВЫ ОТВЕТИ- ЛИ N, SPASE ИЛИ STOP)

- | | | |
|---|---|--|
| E | OUT OF DATA (ВНЕ ДАННЫХ)
ПОПЫТКА ВЫДАТЬ READ, КОГДА СПИСОК ДАННЫХ В DATA КОНЧИЛСЯ. | READ |
| F | INVALID FILE NAME (НЕВЕРНОЕ ИМЯ ФАЙЛА)
ОПЕРАТОР SAVE С ПУСТОЙ СТРОКОЙ ВМЕСТО ИМЕНИ ИЛИ С ИМЕНЕМ ДЛИННЕЕ 10 СИМВОЛОВ. | SAVE |
| G | NO ROOM FOR LINE (НЕТ МЕСТА ДЛЯ СТРОКИ)
НЕДОСТАТОЧНО МЕСТА В ПАМЯТИ ДЛЯ ЗАПИСИ ОЧЕРЕДНОЙ СТРОКИ ПРОГРАММЫ | ВВОД
СТРОКИ В
ПРОГРАММУ |
| H | STOP IN INPUT
НЕКОТОРЫЕ ВВЕДЕННЫЕ ДАННЫЕ НАЧИНАЮТСЯ С ОПЕРАТОРА STOP, ИЛИ БЫЛА НАЖАТА INPUT LINE, ДЕЙСТВИЕ CONTINUE- ОБЫЧНОЕ. | INPUT |
| I | FOR WITHOUT NEXT (FOR БЕЗ NEXT)
ЦИКЛ FOR НИ РАЗУ НЕ ВЫПОЛНЯЛСЯ, НЕ НАЙДЕН NEXT ОПЕРАТОР. | FOR |
| J | INVALID I/O DEVISCE (НЕВЕРНОЕ УСТРОЙСТВО
ВВОДА-ВЫВОДА) | В ОПЕРАЦИИ
С ВНЕШНИМ
УСТРОЙСТВОМ |
| K | INVALID COLOUR (НЕВЕРНЫЙ ЦВЕТ)
СПЕЦИФИЦИРОВАННОЕ ЧИСЛО ИМЕЕТ НЕВЕРНОЕ ЗНАЧЕНИЕ | INK
PAPER
BORDER
FLASH
BRIGHT
INVERS
OVER
А ТАКЖЕ
ПОСЛЕ ОД-
НОЙ ИЗ ПЕ-
РЕДАЧ УП-
РАВЛЯЮЩИХ
СИМВОЛОВ. |
| L | BREAK INTO PROGRAMM (ВРЕАК ВО ВРЕМЯ ВЫПОЛНЕ-
НИЯ ПРОГРАММЫ)
НАЖАТА КЛАВИША ВРЕАК ЭТО ОБНАРУЖИВАЕТСЯ МЕЖДУ ДВУМЯ ОПЕРАТОРАМИ. СТРОКА И НОМЕР ОПЕРАТОРА В СТРОКЕ УКАЗЫВАЕТ НА ОПЕРАТОР ВЫПОЛНЕННЫЙ ПЕРЕД НАЖАТИЕМ ВРЕАК,НО CONTINUE ПЕРЕХОДИТ К СЛЕДУЮЩЕМУ ОПЕРАТОРУ. | РАЗНОЕ |

M	RAMTOR NO COOD (АДРЕС RAMTOR НЕ ГОДЕН)	CLEAR ВОЗМОЖНО RUN
	ЧИСЛО УКАЗАННОЕ ДЛЯ RAMTOR СЛИШКОМ БЕЛИКО ИЛИ СЛИШКОМ МАЛО.	
N	STATEMENT LOST (ОПЕРАТОР ОТСУТСТВУЕТ)	RETURN NEXT CONTINUE
	ПЕРЕХОД К ОПЕРАТОРУ, КОТОРОГО УЖЕ НЕТ	
O	INWALID STREAM (ОШИБОЧНЫЙ ПОТОК ДАННЫХ)	В ОПЕРАЦИИ ВВОДА-ВЫВОД
P	FN WITHOUT DEF (FN БЕЗ DEF)	FN
	ОПРЕДЕЛЯЕМАЯ ПОЛЬЗОВАТЕЛЕМ ФУНКЦИЯ НЕ ОПРЕДЕ- ЛЕНА В ОПЕРАТОРЕ DEF FN.	
Q	PARAMETER ERROR (ОШИБКА В ПАРАМЕТРЕ)	FN
	ОШИБОЧНОЕ ЧИСЛО АРГУМЕНТОВ ИЛИ ОДИН ИЗ НИХ НЕ ТОГО ТИПА, КАКОЙ БЫЛ ОПИСАН.	
R	TAPE LOADING ERROR (ОШИБКА ЗАГРУЗКИ С ЛЕНТЫ)	VERIFY LOAD MERGE
	ФАЙЛ НА ЛЕНТЕ НАЙДЕН, НО НЕ СЧИТЫВАЕТСЯ.	

ПРИЛОЖЕНИЕ С (ЧАСТЬ 1)

ОПИСАНИЕ МИКРОКОМПЬЮТЕРА ZX SPECTRUM

=====

КЛАВИАТУРА.

КАЖДАЯ КЛАВИША КЛАВИАТУРЫ КОМПЬЮТЕРА ZX SPECTRUM ИМЕЕТ МНОГО ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ И ПОЗВОЛЯЕТ ВВОДИТЬ КАК ОТДЕЛЬНЫЕ СИМВОЛЫ, ТАК И ЦЕЛЫЕ СЛОВА. ДЕЙСТВИЕ, ПРОИЗВОДИМОЕ КЛАВИШЕЙ ОПРЕДЕЛЯЕТСЯ ЧАСТИЧНО ПЕРЕКЛЮЧАЮЩИМИ КЛАВИШАМИ (CAPS SHIFT И SYMBOL SHIFT), А ЧАСТИЧНО РЕЖИМОМ, В КОТОРОМ НАХОДИТСЯ КОМПЬЮТЕР.

РЕЖИМ ОТОБРАЖАЕТСЯ КУРСОРОМ, МЕРЦАЮЩЕЙ БУКВОЙ, УКАЗЫВАЮЩЕЙ ПОЗИЦИЮ В КОТОРУЮ БУДЕТ ВВОДИТСЯ ОЧЕРЕДНОЙ СИМВОЛ С КЛАВИАТУРЫ.

ВОЗМОЖНЫ СЛЕДУЮЩИЕ РЕЖИМЫ:

К- (ДЛЯ КЛЮЧЕВЫХ СЛОВ) KEYWORDS.

ЭТОТ РЕЖИМ АВТОМАТИЧЕСКИ СМЕНЯЕТ РЕЖИМ L, ЕСЛИ КОМПЬЮТЕР ПЕРЕХОДИТ В ОЖИДАНИЕ ВВОДА КОМАНДЫ ИЛИ СТРОКИ ПРОГРАММЫ. ЭТО МОЖЕТ БЫТЬ ЛИБО В НАЧАЛЕ СТРОКИ, ЛИБО ПОСЛЕ THEN, ЛИБО ПОСЛЕ ':' И ЕСЛИ НЕ БЫЛО НАЖАТИЯ ПЕРЕКЛЮЧАЮЩИХ КЛАВИШ, ТО НАЖАТИЕ ЛЮБОЙ КЛАВИШИ БУДЕТ ИНТЕРПРЕТИРОВАТЬСЯ КАК КЛЮЧЕВОЕ СЛОВО (НАПИСАННОЕ НА КЛАВИШЕ) ИЛИ ЦИФРА.

L- (ДЛЯ БУКВ) LETTER.

ОСНОВНОЙ РЕЖИМ ДЛЯ КОМПЬЮТЕРА, ЕСЛИ НЕ БЫЛО ПЕРЕКЛЮЧЕНИЯ РЕГИСТРОВ, ТО КЛАВИША ИНТЕРПРЕТИРУЕТСЯ КАК ОСНОВНОЙ СИМВОЛ, НАНЕСЕННЫЙ НА ЭТУ КЛАВИШУ.

ДЛЯ ОБОИХ РЕЖИМОВ L И К ПРИ НАЖАТИИ С КЛАВИШЕЙ ОДНОВРЕМЕННО КЛАВИШИ SYMBOL SHIFT, КЛАВИША БУДЕТ ИНТЕРПРЕТИРОВАТЬСЯ КАК ВСПОМОГАТЕЛЬНЫЙ СИМВОЛ. А ПРИ НАЖАТИИ CAPS SHIFT С ЦИФРОЙ, КЛАВИША БУДЕТ ИНТЕРПРЕТИРОВАТЬСЯ КАК УПРАВЛЯЮЩАЯ ФУНКЦИЯ, НАПИСАННАЯ НА

БЕЛОМ ПОЛЕ КЛАВИШИ. НАЖАТИЕ CAPS SHIFT С ЛЕВОЙ ИЗ КЛАВИШ НЕ ВЫЗЫВАЕТ КЛЮЧЕВОГО СЛОВА В РЕЖИМАХ К И L.

C- (ДЛЯ ЗАГЛАВНЫХ БУКВ) CAPITAL

РЕЖИМ ПРЕДСТАВЛЯЕТ СОБОЙ ВАРИАНТ РЕЖИМА L, В КОТОРОМ ИСПОЛЗУЮТСЯ ЗАГЛАВНЫЕ БУКВЫ.

CAPS LOCK ИСПОЛЗУЕТСЯ ДЛЯ ПЕРЕХОДА ИЗ РЕЖИМА L В C И ОБРАТНО.

E- (ДЛЯ РАСШИРЕНИЯ) EXTEND

ИСПОЛЗУЕТСЯ ДЛЯ ПОЛУЧЕНИЯ ДАЛЬНЕЙШИХ СИМВОЛОВ, ГЛАВНЫМ ОБРАЗОМ ЗНАКОВ. ЭТОТ РЕЖИМ ВВОДИТСЯ ОДНОВРЕМЕННЫМ НАЖАТИЕМ ДВУХ ПЕРЕКЛЮЧАЮЩИХ КЛАВИШ С УДЕРЖАНИЕМ ЗАТЕМ ТОЛЬКО ОДНОЙ КЛАВИШИ. ЭТОМ РЕЖИМЕ КЛАВИША ДАЕТ ОДИН СИМВОЛ ИЛИ ЗНАК (ИЗОБРАЖЕННЫЙ НА ЗЕЛЕНОМ ПОЛЕ КЛАВИШИ). ЕСЛИ НЕ НАЖАТА ПЕРЕКЛЮЧАЮЩАЯ КЛАВИША, ИЛИ ЗНАК, ИЗОБРАЖЕННЫЙ НА КРАСНОМ ПОЛЕ, ЕСЛИ УДЕРЖИВАЕТСЯ ПЕРЕКЛЮЧАЮЩАЯ КЛАВИША.

ЦИФРОВЫЕ КЛАВИШИ ВЫДАЮТ ЗНАК, ЕСЛИ НАЖИМАЮТСЯ ВМЕСТЕ С SYMBOL SHIFT, В ПРОТИВНОМ СЛУЧАЕ ОНИ ВЫДАЮТ ПОСЛЕДОВАТЕЛЬНОСТЬ УПРАВЛЕНИЯ ЦВЕТОМ.

G- (ДЛЯ ГРАФИКИ) GRAPHICS.

РЕЖИМ ВВОДИТСЯ ПОСЛЕ НАЖАТИЯ GRAPHICS (CAPS SHIFT И 9) СОХРАНЯЕТСЯ ДО СЛЕДУЮЩЕГО НАЖАТИЯ ЭТОЙ КЛАВИШИ.

ЦИФРОВЫЕ КЛАВИШИ БУДУТ ВЫДАВАТЬ МОЗАИЧНЫЕ ГРАФИЧЕСКИЕ СИМВОЛЫ, СОХРАНЯЯ GRAPHICS И DELETE, А КАЖДАЯ АЛФАВИТНАЯ КЛАВИША, КРОМЕ V, N, X, Y И Z, БУДЕТ ВЫДАВАТЬ ОПРЕДЕЛЕННЫЙ ПОЛЬЗОВАТЕЛЕМ ГРАФИЧЕСКИЙ СИМВОЛ.

ЕСЛИ НЕКОТОРАЯ КЛАВИША БУДЕТ УДЕРЖИВАТЬСЯ БОЛЕЕ 2-3Х СЕКУНД, ЭТО ВЫЗОВЕТ ПОВТОРЕНИЕ ПРОИЗВОДИМОГО ЕЮ ДЕЙСТВИЯ.

ВВОД С КЛАВИАТУРЫ ПРОИЗВОДИТСЯ В НИЖНЮЮ ЧАСТЬ ЭКРАНА, КАЖДЫЙ СИМВОЛ (ИЛИ СОСТАВНОЙ ЗНАК) ВСТАВЛЯЕТСЯ ПЕРЕД КУРСОРОМ. КУРСОР МОЖЕТ БЫТЬ ПЕРЕСЛАН ВЛЕВО ДЕЙСТВИЕМ CAPS SHIFT И 5, А ВПРАВО CAPS SHIFT И 8. СИМВОЛ ПЕРЕД КУРСОРОМ МОЖНО УДАЛИТЬ КОМАНДОЙ DELETE (CAPS SHIFT И 0). ЦЕЛАЯ СТРОКА МОЖЕТ БЫТЬ УДАЛЕНА ВВОДОМ EDIT (CAPS SHIFT И 1) И ПОСЛЕДУЮЩИМ ENTER. КОГДА НАЖИМАЕТСЯ ENTER ВЫПОЛНЯЕТСЯ НАБРАННАЯ СТРОКА, ЛИБО ОНА ВВОДИТСЯ В ПРОГРАММУ, ЛИБО ОНА ИСПОЛЗУЕТСЯ КАК ВХОДНЫЕ ДАННЫЕ ДЛЯ INPUT ОПЕРАТОРА, ЛИБО В СТРОКЕ ИМЕЮТСЯ СИНТАКТИЧЕСКИЕ ОШИБКИ. В ЭТОМ СЛУЧАЕ НАЖАТИЕМ [?] ПРОИСХОДИТ ПЕРЕХОД НА ОШИБКУ.

КОГДА СТРОКИ ПРОГРАММЫ ВВЕДЕНА, ЛИСТИНГ ОТОБРАЖАЕТСЯ В ВЕРХНЕЙ ЧАСТИ ЭКРАНА. БОЛЕЕ ПОДРОБНО ЭТОТ ПРОЦЕСС ОПИСАН В ГЛАВЕ 2.

ПОСЛЕДНЯЯ ВВЕДЕННАЯ СТРОКА НАЗЫВАЕТСЯ ТЕКУЩЕЙ И ОТМЕЧАЕТСЯ СИМВОЛОМ [>], И ЕЕ МОЖНО ИЗМЕНИТЬ, ИСПОЛЗУЯ КЛАВИШИ ПЕРЕМЕЩЕНИЯ КУРСОРА ВВЕРХ И ВНИЗ (CAPS SHIFT И 6; CAPS SHIFT И 7).

ЕСЛИ ВВЕДЕН EDIT (CAPS SHIFT И 1), ТО ТЕКУЩАЯ СТРОКА ПЕРЕНОСИТСЯ В НИЖНЮЮ ЧАСТЬ ЭКРАНА И СТАНОВИТСЯ ДОСТУПНОЙ ДЛЯ РЕДАКТИРОВАНИЯ.

ЕСЛИ ВЫПОЛНЯЕТСЯ КОМАНДА ИЛИ ЦЕЛАЯ ПРОГРАММА, ТО РЕЗУЛЬТАТ ОТОБРАЖАЕТСЯ В ВЕРХНЕЙ ЧАСТИ ЭКРАНА И СОХРАНЯЕТСЯ ДО ВВОДА СТРОКИ ПРОГРАММЫ, ВВОДА ПУСТОЙ СТРОКИ ИЛИ НАЖАТИЯ КЛАВИШИ УПРАВЛЕНИЯ КУРСОРОМ ВВЕРХ ИЛИ ВНИЗ. В НИЖНЕЙ ЧАСТИ ВЫДАЕТСЯ СООБЩЕНИЕ И КОДЫ ПРИВЕДЕННЫЕ В ПРИЛОЖЕНИИ В. В СООБЩЕНИИ УКАЗЫВАЕТСЯ НОМЕР ОШИБОЧНОЙ СТРОКИ (0 ДЛЯ КОМАНДЫ) И ПОЗИЦИЯ ОПЕРАТОРА В ЭТОЙ СТРОКЕ. СООБЩЕНИЕ СОХРАНЯЕТСЯ НА ЭКРАНЕ ДО НАЖАТИЯ ЛЮБОЙ КЛАВИШИ (ОТОБРАЖАЕТСЯ ПЕРЕХОДОМ В РЕЖИМ K).

В ОПРЕДЕЛЕННЫХ ОБСТОЯТЕЛЬСТВАХ CAPS SHIFT И SPACE ДЕЙСТВУЕТ КАК ВБРЕАК, ОСТАНАВЛИВАЯ КОМПЬЮТЕР С СООБЩЕНИЯМИ D ИЛИ L, И ПРИ ЭТОМ ДО ОСТАНОВА:

А) ЗАВЕРШАЕТСЯ ВЫПОЛНЯВШИЙСЯ ОПЕРАТОР ИЛИ

Б) ЗАВЕРШАЮТСЯ ДЕЙСТВИЯ, ВЫПОЛНЯЕМЫЕ КОМПЬЮТЕРОМ С МАГНИТОФОНОМ ИЛИ ПРИНТЕРОМ.

ПРИМЕР ИСПОЛЬЗОВАНИЯ КЛАВИАТУРЫ:

ПРИ КУРСОРЕ [K]:

LN	1. ПРОСТОЕ НАЖАТИЕ КЛАВИШИ	--НА ЭКРАНЕ-->COPY
	2. SYMBOL SHIFT И КЛАВИША	--НА ЭКРАНЕ--> Z
I	3. CAPS SHIFT И КЛАВИША	--КУРСОР -->[L]
	4. BREAK/SPACE И КЛАВИША	--НА ЭКРАНЕ--> ;
Z	5. SYMBOL SHIFT И CAPS SHIFT	--НА ЭКРАНЕ--> Z
COPY	ПРИ КУРСОРЕ [E]	--КУРСОР. -->[E]
BEEP	6. ПРОСТОЕ НАЖАТИЕ КЛАВИШИ	--НА ЭКРАНЕ--> LN
	7. CAPS SHIFT И КЛАВИША	--НА ЭКРАНЕ-->BEEP

ЭКРАН ТЕЛЕВИЗОРА

ЭКРАН ТЕЛЕВИЗОРА СОДЕРЖИТ 24 СТРОКИ ПО 32 ПОЗИЦИИ В КАЖДОЙ. ДЕЛИТСЯ НА ДВЕ ЧАСТИ, ВЕРХНЯЯ ЧАСТЬ В 22 СТРОКИ ОТОБРАЖАЕТ ЛИСТИНГ ИЛИ ВЫВОД ИЗ ПРОГРАММЫ. КОГДА ВЫВОД В ВЕРХНЕЙ ЧАСТИ ДОСТИГАЕТ НИЖА, НЕОБХОДИМА СВЕРТКА НА ОДНУ СТРОКУ, ПРИ ЭТОМ МОЖЕТ ЗАХВАТЫВАТЬСЯ СТРОКА, КОТОРУЮ ВАМ ХОЧЕТСЯ СОХРАНИТЬ. КОМПЬЮТЕР В ЭТОМ СЛУЧАЕ ОСТАНАВЛИВАЕТСЯ С ЗАПРОСОМ 'SCROLL?'. ЕСЛИ ТЕПЕРЬ НАЖАТЬ КЛАВИШИ N, SPACE ИЛИ STOP, ТО ПРОГРАММА ОСТАНОВИТСЯ ВЫДАЧЕЙ СООБЩЕНИЯ "D BREAK=CONT REPORTS". НАЖАТИЕ ДРУГИХ КЛАВИШ РАЗРЕШАЕТ СВЕРТКУ И ПРОДОЛЖЕНИЕ ВЫПОЛНЕНИЯ.

НИЖНЯЯ ЧАСТЬ ИСПОЛЗУЕТСЯ ДЛЯ ВВОДА КОМАНД, СТРОК ПРОГРАММЫ ВХОДНЫХ INPUT-ДАННЫХ, А ТАКЖЕ ДЛЯ ОТОБРАЖЕНИЯ СООБЩЕНИЙ. НИЖНЯЯ ЧАСТЬ ЭКРАНА СОСТОИТ ИЗ ДВУХ СТРОК (ВЕРХНЯЯ ИЗ НИХ ЧИСТАЯ -ДЛЯ РАСШИРЕНИЯ). ПРИ ПЕРЕПОЛНЕНИИ ВЕРХНЕЙ СТРОКИ ОСУЩЕСТВЛЯЕТСЯ СВЕРТКА.

КАЖДАЯ ПОЗИЦИЯ ИМЕЕТ АТТРИБУТЫ, ОПРЕДЕЛЯЮЩИЕ ЕЕ КАК ЧИСТУЮ (ЦВЕТ ФОНА), ЛИБО КАК ЗАКРАШЕННУЮ (ОСНОВНОЙ ЦВЕТ), С ПОВЫШЕННОЙ ИЛИ С ПОНИЖЕННОЙ ЯРКОСТЬЮ, МЕРЦАЮЩУЮ ИЛИ НЕТ.

ДОСТУПНЫ ЦВЕТА: ЧЕРНЫЙ, ГОЛУБОЙ, КРАСНЫЙ, ПУРПУРНЫЙ(ФИОЛЕТОВЫЙ), ЗЕЛЕНый, ЖЕЛТЫЙ И БЕЛЫЙ.

КРАЯ ЭКРАНА МОГУТ БЫТЬ УСТАНОВЛЕНЫ В ОПРЕДЕЛЕННЫЙ ЦВЕТ ИСПОЛЬЗОВАНИЕМ ОПЕРАТОРА BORDER.

КАЖДАЯ ПОЗИЦИЯ ПОДРАЗДЕЛЯЕТСЯ НА 8x8 ТОЧЕК, А ГРАФИКА СИМВОЛОВ ОБЕСПЕЧИВАЕТСЯ ИНДИВИДУАЛЬНЫМ ОПРЕДЕЛЕНИЕМ КАЖДОЙ ТОЧКИ. АТТРИБУТЫ КАЖДОЙ ПОЗИЦИИ НАСТРАИВАЮТСЯ ПРИ ЗАПИСИ СИМВОЛА ИЛИ ПРИ УСТАНОВКЕ ТОЧКИ(PIXEL). СПОСОБ НАСТРОЙКИ ОПРЕДЕЛЯЕТСЯ ПАРАМЕТРАМИ ВЫВОДА, ИМЕЮЩИМИ ДВЕ УСТАНОВКИ (ПОСТОЯННУЮ И ВРЕМЕННУЮ) ШЕСТИ ОПЕРАТОРАХ:

PAPER, INK, FLASH, BRIGHT, INVERSE И OVER. ПОСТОЯННЫЕ ПАРАМЕТРЫ ДЛЯ ВЕРХНЕЙ ЧАСТИ ЭКРАНА УСТАНОВЛИВАЮТСЯ ОПЕРАТОРАХ PAPER, INK И Т.Д. ОБЫЧНО ОНИ ИМЕЮТ ЧЕРНЫЙ ЦВЕТ ДЛЯ ЗАКРАШЕННОЙ ТОЧКИ (INK) И БЕЛЫЙ ДЛЯ ФОНОВОЙ(PAPER). НОРМАЛЬНУЮ ЯРКОСТЬ, НЕ МЕРЦАЮЩИЕ, НЕ ИНВЕРСНЫЕ. ПОСТОЯННЫЕ ПАРАМЕТРЫ ДЛЯ НИЖНЕЙ ЧАСТИ ЭКРАНА ИСПОЛЗУЮТ ЦВЕТ РАМКИ (BORDER COLOUR) КАК ЦВЕТ ФОНА (НЕЗАКРАШЕННЫЙ), С ЧЕРНЫМ ИЛИ БЕЛЫМ ЦВЕТОМ, НОРМАЛЬНУЮ ЯРКОСТЬ, НЕ МЕРЦАЮЩИЕ.

ВРЕМЕННЫЕ ПАРАМЕТРЫ УСТАНОВЛИВАЮТСЯ КОМАНДАМИ:

PAPER, INK И Т.Д. ВСТАВЛЯЕМЫМИ В ОПЕРАТОРЫ

PRINT, LPRINT, INPUT, PLOT, DRAW, CIRCLE, А ТАКЖЕ PAPER, INK И ТОМУ ПОДОБНЫМИ УПРАВЛЯЮЩИМИ СИМВОЛАМИ, КОГДА ОН ВЫВОДИТСЯ НА ТЕЛЕВИЗОР.

ВРЕМЕННЫЕ ПАРАМЕТРЫ СОХРАНЯЮТСЯ ДО КОНЦА ДЕЙСТВИЯ ОПЕРАТОРА PRINT (ИЛИ ДРУГИХ).

ПАРАМЕТРЫ PAPER И INK МОГУТ ПРИНИМАТЬ ЗНАЧЕНИЯ ОТ 0 ДО 9.

ПАРАМЕТРЫ ОТ 0 ДО 7 ОПРЕДЕЛЯЮТ ЦВЕТА ВЫВОДИМОГО СИМВОЛА:

0- ЧЕРНЫЙ	(BLACK)
1- ГОЛУБОЙ	(BLUE)
2- КРАСНЫЙ	(RED)
3- ФИОЛЕТОВЫЙ	(MAGENTA)
4- ЗЕЛЕНый	(GREEN)
5- СИНИЙ	(CYAN)
6- ЖЕЛТЫЙ	(YELLOW)
7- БЕЛЫЙ	(WHITE)

ПАРАМЕТР 8 ОПРЕДЕЛЯЕТ, ЧТО ЦВЕТ ДОЛЖЕН ОСТАТЬСЯ ПРИ ВЫВОДЕ БЕЗ ИЗМЕНЕНИЯ.

ПАРАМЕТР 9 (КОНТРАСТНОСТЬ) ОПРЕДЕЛЯЕТ, ЧТО ЦВЕТ ДОЛЖЕН СТАТЬ ЛИБО БЕЛЫМ, ЛИБО ЧЕРНЫМ ДЛЯ ВЫДЕЛЕНИЯ ЕГО ОТ ДРУГИХ ЦВЕТОВ.

ПАРАМЕТРЫ FLASH И BRIGHT МОГУТ ПРИНИМАТЬ ЗНАЧЕНИЯ 0, 1 ИЛИ 8. ПАРАМЕТР 1 УКАЗЫВАЕТ, ЧТО ВКЛЮЧАЕТСЯ ПОВЫШЕННАЯ ЯРКОСТЬ И МЕРЦАНИЕ.

ПАРАМЕТР 0 УКАЗЫВАЕТ, ЧТО ПОВЫШЕННАЯ ЯРКОСТЬ И МЕРЦАНИЕ ОТКЛЮЧАЮТСЯ.

ПАРАМЕТР 8 УКАЗЫВАЕТ, ЧТО ВСЕ ОСТАЕТСЯ БЕЗ ИЗМЕНЕНИЙ.

ПАРАМЕТРЫ OVER И INVERSE МОГУТ ПРИНИМАТЬ ЗНАЧЕНИЯ 0 И 1.

OVER 0 - НОВЫЙ СИМВОЛ ЗАТИРАЕТ СТАРЫЙ

OVER 1 - КОД СТАРОГО СИМВОЛА И НОВОГО СИМВОЛА СОЕДИНЯЮТСЯ ОПЕРАЦИЕЙ "ИСКЛЮЧАЮЩЕГО ИЛИ", ОБРАЗУЯ НОВЫЙ СИМВОЛ (OVER PRINTING).

INVERSE 0 - НОВЫЙ СИМВОЛ ПЕЧАТАЕТСЯ В НЕИНВЕРСНОМ (ПОЗИТИВНОМ) ВИДЕ.

INVERSE 1 - НОВЫЙ СИМВОЛ ПЕЧАТАЕТСЯ В ИНВЕРСНОМ (НЕГАТИВНОМ) ВИДЕ
КОГДА НА ТЕЛЕВИЗОР ПЕРЕДАЕТСЯ УПРАВЛЯЮЩИЙ СИМВОЛ ТАВ, ТО ДВА СТАРЫХ БАЙТА ИСПОЛЗУЮТСЯ ДЛЯ СПЕЦИФИКАЦИИ ТАВ STOP N (ПЕРВЫЙ БАЙТ ЯВЛЯЕТСЯ СТАРЫМ). ЭТО ОБЕСПЕЧИВАЕТСЯ ПРОГОНОМ ОТ 32 ДО "N" (УКАЗАННЫМ В ТАВ) И ЗАТЕМ ВЫВОДОМ НУЖНОГО КОЛИЧЕСТВА ПРОБЕЛОВ ДЛЯ СМЕЩЕНИЯ ТЕКУЩЕЙ ПОЗИЦИИ ВЫВОДА В КОЛОНКУ "N".

ЕСЛИ НА ВЫВОД ПЕРЕДАЕТСЯ ЗАПЯТАЯ КАК УПРАВЛЯЮЩИЙ СИМВОЛ, ТО ВЫВОДИТСЯ НУЖНОЕ КОЛИЧЕСТВО ПРОБЕЛОВ ДЛЯ ПЕРЕВОДА ТЕКУЩЕЙ ПОЗИЦИИ ВЫВОДА В ПОЗИЦИЮ 0 ИЛИ 16.

ЕСЛИ ПЕРЕДАЕТСЯ УПРАВЛЯЮЩИЙ СИМВОЛ ENTER, ТО ПОЗИЦИЯ ВЫВОДА ПЕРЕВОДИТСЯ НА СЛЕДУЮЩУЮ СТРОКУ.

П Р И Н Т Е Р

ВЫВОД НА ПРИНТЕР ОСУЩЕСТВЛЯЕТСЯ ЧЕРЕЗ БУФЕР ДЛИНОЙ В 3 СИМВОЛА, ОЧЕРЕДНАЯ СТРОКА ВЫДАЕТСЯ ИЗ БУФЕРА НА ПРИНТЕР В СЛЕДУЮЩИХ СЛУЧАЯХ:

А) КОГДА ОКОНЧЕН ВЫВОД ОДНОЙ СТРОКИ И ВЫВОД ПЕРЕХОДИТ К ДРУГОЙ СТРОКЕ,

Б) ПРИ ПЕРЕДАЧИ В БУФЕР СИМВОЛА ENTER,

В) ПРИ ЗАВЕРШЕНИИ ПРОГРАММЫ, ЕСЛИ ЕЩЕ ОСТАЛИСЬ ДРУГИЕ НЕВВЕДЕННЫЕ ДАННЫЕ,

Г) ЕСЛИ ВСТРЕТИЛИСЬ УПРАВЛЯЮЩИЕ СИМВОЛЫ ТАВ ИЛИ ЗАПЯТАЯ, ТРЕБУЮЩИЕ ПЕРЕВОДА СТРОКИ.

УПРАВЛЯЮЩИЕ СИМВОЛЫ ТАВ И ЗАПЯТАЯ ПРОИЗВОДЯТ ВЫВОД ПРОБЕЛОВ ПРИ РАБОТЕ С ТЕЛЕВИЗОРОМ.

УПРАВЛЯЮЩИЙ СИМВОЛ ТАВ ИЗМЕНЯЕТ ПОЗИЦИЮ ВЫВОДА, ИСПОЛЬЗУЯ ЧИСЛО ЗАДАЮЩЕЕ ПОЗИЦИЮ.

ПРИНТЕР ТАКЖЕ ПРАВИЛЬНО РЕАГИРУЕТ НА УПРАВЛЯЮЩИЕ СИМВОЛЫ INVERSE, OVER (И ОПЕРАТОРЫ С ТЕМ ЖЕ ИМЕНЕМ), НО НЕ ВОСПРИНИМАЕТ PAPER INK, FLASH И BRIGHT.

ПРИ ВВОДЕ BREAK ПРИНТЕР ОСТАНАВЛИВАЕТСЯ С ВЫДАЧЕЙ СООБЩЕНИЯ:

'В'. ПРИ ОТСУТСТВИИ ПРИНТЕРА ВЫВОД ПРОСТО НЕ ОСУЩЕСТВЛЯЕТСЯ.

П Р И Л О Ж Е Н И Е С (ЧАСТЬ 2)

ЯЗЫК ПРОГРАММИРОВАНИЯ БЕЙСИК

=====

СПРАВОЧНОЕ ПОСОБИЕ

ВСЕ ЧИСЛА В СИСТЕМЕ МОГУТ ИМЕТЬ ТОЧНОСТЬ 9 ИЛИ 10 ЗНАКОВ. НАИБОЛЬШЕЕ ЧИСЛО $10^{**}3$, НАИМЕНЬШЕЕ ПОЛОЖИТЕЛЬНОЕ ЧИСЛО $4*10^{**}(-39)$. ЧИСЛА ИМЕЮТ ВНУТРЕННЕЕ ПРЕДСТАВЛЕНИЕ КАК ЧИСЛА С ПЛАВАЮЩЕЙ (ДВОИЧНОЙ) ТОЧКОЙ, С ВЫДЕЛЕНИЕМ ОДНОГО БИТА НА ПОКАЗАТЕЛЬ СТЕПЕНИ "Е" (ЭКСПОНЕНТЫ) В ИНТЕРВАЛЕ ОТ 1 ДО 255, И ЧЕТЫРЕХ БАЙТОВ НА МАНТИССУ "М" В ИНТЕРВАЛЕ ОТ 2,5 ДО 1 ($M=1$), ПРЕДСТАВЛЯЕТСЯ ЧИСЛОМ $M*2^{**}E(-128)$.

ПОСКОЛЬКУ $1/2 \leq M < 1$, СТАРШИЙ ЗНАЧАЩИЙ БИТ МАНТИССЫ ВСЕГДА 1. СЛЕДОВАТЕЛЬНО, МЫ МОЖЕМ ЗАМЕНИТЬ ЕГО НА БИТ, ОБОЗНАЧАЮЩИЙ ЗНАК:

0 - ДЛЯ ПОЛОЖИТЕЛЬНОГО ЧИСЛА И 1 - ДЛЯ ОТРИЦАТЕЛЬНОГО.

НАИМЕНЬШЕЕ ЦЕЛОЕ ИМЕЕТ СПЕЦИАЛЬНОЕ ПРЕДСТАВЛЕНИЕ, В КОТОРОМ ПЕРВЫЙ БАЙТ 0, ВТОРОЙ БАЙТ ЗНАКА (0 И FFH), А ТРЕТИЙ И ЧЕТВЕРТЫЙ САМО ЧИСЛО В ДОПОЛНИТЕЛЬНОМ КОДЕ (МЛАДШИЕ ЗНАЧАЩИЕ ЦИФРЫ В ПЕРВОМ БАЙТЕ).

ЧИСЛОВЫЕ ПЕРЕМЕННЫЕ ИМЕЮТ ИМЯ ПРОИЗВОЛЬНОЙ ДЛИНЫ, НАЧИНАЮЩЕЕСЯ С БУКВЫ И ПРОДОЛЖАЮЩЕЕСЯ БУКВАМИ ИЛИ ЦИФРАМИ. ПРОБЕЛЫ И СИМВОЛЫ УПРАВЛЕНИЯ ЦВЕТОМ ИГНОРИРУЮТСЯ И ВСЕ БУКВЫ ПРЕСБРАЗУЮТСЯ К МИНИМАЛЬНО УПАКОВАННОМУ ВИДУ.

УПРАВЛЯЮЩИЕ ПЕРЕМЕННЫЕ ДЛЯ FOR-NEXT ЦИКЛОВ ИМЕЮТ ИМЕНА ДЛИНОЙ В ОДНУ БУКВУ.

ЧИСЛОВЫЕ МАССИВЫ ИМЕЮТ ИМЕНА ДЛИНОЙ В ОДНУ БУКВУ, КОТОРАЯ МОЖЕТ БЫТЬ ТАКОЙ ЖЕ, КАК ИМЯ СКАЛЯРНОЙ ПЕРЕМЕННОЙ. ЭТИ МАССИВЫ МОГУТ ИМЕТЬ ПРОИЗВОЛЬНОЕ КОЛИЧЕСТВО ИЗМЕРЕНИЙ И ПРОИЗВОЛЬНЫЙ РАЗМЕР. НАЧАЛЬНЫЙ ИНДЕКС ВСЕГДА 1.

СТРОКИ СИМВОЛОВ БОЛЕЕ ГИБКИЕ В СВОЕЙ ДЛИНЕ. ИМЯ СТРОКОВОЙ ПЕРЕМЕННОЙ В ОТЛИЧИИ ОТ ПРОСТОЙ ПЕРЕМЕННОЙ ЗАКАНЧИВАЕТСЯ ЗНАКОМ ДОЛЛАРА(\$).

СТРОКОВЫЕ МАССИВЫ ТАКЖЕ МОГУТ ИМЕТЬ ПРОИЗВОЛЬНОЕ КОЛИЧЕСТВО ИЗМЕРЕНИЙ И РАЗМЕР. ИХ ИМЕНА ПРЕДСТАВЛЯЮТ СОБОЙ ОДНУ БУКВУ И СЛЕДУЮЩИЙ ЗА НЕЙ СИМВОЛ \$, НО НЕ МОГУТ СОВПАДАТЬ С ИМЕНЕМ ПРОСТОЙ СТРОКИ СИМВОЛОВ.

ВСЕ СТРОКИ В МАССИВАХ ИМЕЮТ ФИКСИРОВАННУЮ ДЛИНУ, КОТОРАЯ ОПРЕДЕЛЯЕТСЯ ЧИСЛОМ, ЗАДАЮЩИМ ПОСЛЕДНЮЮ РАЗМЕРНОСТЬ В ОПЕРАТОРЕ DIM. НАЧАЛЬНЫЙ ИНДЕКС 1.

ПОДСТРОКА ОТ СТРОКИ МОЖЕТ БЫТЬ ПОЛУЧЕНА КАК СЕЧЕНИЕ. СЕЧЕНИЕ МОЖЕТ БЫТЬ:

А) ПУСТЫМ

Б) ЧИСЛОВЫМ ВЫРАЖЕНИЕМ

В) НЕКОТОРЫМ ЧИСЛОВЫМ ВЫРАЖЕНИЕМ "ТО", ДРУГИМ ЧИСЛОВЫМ ВЫРАЖЕНИЕМ И ИСПОЛЬЗОВАТЬСЯ В;

*) СТРОКОВЫХ ВЫРАЖЕНИЯХ (СЕЧЕНИЯХ)

**) СТРОКОВЫХ МАССИВАХ ПЕРЕМЕННЫХ

(ИНДЕКС 1, ИНДЕКС 2, ИНДЕКС N, СЕЧЕНИЕ)

ИЛИ ЧТО ТОЖЕ САМОЕ

(ИНДЕКС 1, ИНДЕКС 2, ИНДЕКС N) (СЕЧЕНИЕ).

В СЛУЧАЕ *) СТРОКА ВЫРАЖЕНИЯ ИМЕЕТ ЗНАЧЕНИЕ S\$. ЕСЛИ СЕЧЕНИЕ МАССИВА ПУСТО, ТО S\$ СЧИТАЕТСЯ ПОДСТРОКОЙ ОТ САМОЙ СЕБЯ.

ЕСЛИ СЕЧЕНИЕ ПРЕДСТАВЛЯЕТСЯ ЧИСЛОВЫМ ВЫРАЖЕНИЕМ С ЗНАЧЕНИЕМ "M", ТО РЕЗУЛЬТАТОМ БУДЕТ M-ЫЙ СИМВОЛ ОТ S\$ (ПОДСТРОКА ДЛИНОЙ 1) ЕСЛИ СЕЧЕНИЕ ПРЕДСТАВЛЕНО В ФОРМЕ В) И ПЕРВОЕ ЧИСЛОВОЕ ВЫ-

РАЖЕНИЕ ИМЕЕТ ЗНАЧЕНИЕ 'М' (УМАЛЧИВАЕМОЕ ЗНАЧЕНИЕ 1), А ВТОРОЕ 'N' (УМАЛЧИВАЕМОЕ ЗНАЧЕНИЕ $S \times$), И ЕСЛИ $1 \leq m \leq n \leq$ ЧЕМ ДЛИНА $S \times$, ТО РЕЗУЛЬТАТОМ БУДЕТ ПОДСТРОКА ОТ $S \times$ С М-ЫМ НАЧАЛЬНЫМ СИМВОЛОМ И N-ЫМ КОНЕЧНЫМ.

ЕСЛИ $0 \leq N < M$, ТО РЕЗУЛЬТАТОМ БУДЕТ ПУСТАЯ СТРОКА, В ЛЮБОМ ДРУГОМ СЛУЧАЕ ВЫДАЕТСЯ СООБЩЕНИЕ ОБ ОШИБКЕ '3'.

СЕЧЕНИЕ ВЫПОЛНЯЕТСЯ ПЕРЕД ФУНКЦИЕЙ ИЛИ ОПЕРАЦИЕЙ, КОТОРАЯ ОСУЩЕСТВЛЯЕТСЯ, ЕСЛИ СКОБКАМИ НЕ ПРЕДПИСАНО СДЕЛАТЬ ИНАЧЕ.

ПОДСТРОКА МОЖЕТ НАЗНАЧАТЬСЯ (СМОТРИ ОПЕРАТОР LET). ЕСЛИ ЧАСТЬ СТРОКИ ЗАПИСАНА В СТРОКОВЫЙ ЛИТЕРАЛ, ОНА ДОЛЖНА УДВАИВАТЬСЯ.

ФУНКЦИИ

ИМЯ ФУНКЦИИ	ТИП АРГУМЕНТА	ДЕЙСТВИЕ (ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ)
ABS	число	абсолютное значение
ACS	число	арккосинус в радианах, выдает сообщение об ошибке A, если X не лежит в интервале от -1 до 1
AND	логическая операция правый операнд всегда число.	слева может быть: - число, тогда ---- $\rightarrow A \text{ AND } B =$ $L \text{ B, если } B=0$ - строка, тогда --- $\rightarrow A^x \text{ AND } B=$ $L " " , \text{ если } B=0$
ASN	число	арксинус в радианах, выдает сообщение A, если не лежит в интервале от -1 до 1
ATN	число	арктангенс в радианах.
ATTR	два числовых аргумента X и Y, заключаемые в скобки	число, двоичный код которого, представляет собой атрибуты Y-ой позиции X-ой строки экрана, бит 7 (старший) равен: 1 - для мерцающего поля, и 0 для немерцающего, биты с 5 по 3 - цвет фона, биты с 2 по 1 - цвет закрашивания, выдает сообщение B, если $0 \leq x \leq 23$ и $0 \leq y \leq 31$.

BIN		ЭТО НЕ ОБЫЧНАЯ ФУНКЦИЯ, ЗА BIN ЗАПИСЫВАЕТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ НУЛЕЙ И ЕДИНИЦ ПРЕДСТАВЛЯЮЩАЯ СОБОЙ ДВОИЧНОЕ ПРЕДСТАВЛЕНИЕ ЧИСЛА, КОТОРОЕ ЗАПИСЫВАЕТСЯ В ПАМЯТЬ.
CHR $\times$	ЧИСЛО	СИМВОЛ, ЧЕЙ КОД ПРЕДСТАВИМ ЧИСЛОМ X, ОКРУГЛЕННЫМ К БЛИЖАЙШЕМУ ЦЕЛОМУ.
CODE	СТРОКА СИМВОЛОВ	КОД ПЕРВОГО СИМВОЛА В СТРОКЕ X (ИЛИ 0, ЕСЛИ X - ПУСТАЯ СТРОКА).
COS	ЧИСЛО В РАДИАНАХ	КОСИНУС X
EXP	ЧИСЛО	E В СТЕПЕНИ X
FN		FN С ПОСЛЕДУЮЩИМ ИМЕНЕМ, ОПРЕДЕЛЕННОЙ ПОЛЬЗОВАТЕЛЕМ ФУНКЦИИ (СМ. DEF), АРГУМЕНТЫ ДОЛЖНЫ ЗАКЛЮЧАТЬСЯ В СКОБКИ, ДАЖЕ, ЕСЛИ НЕТ АРГУМЕНТОВ, СКОБКИ ВСЕ РАВНО ДОЛЖНЫ ЗАПИСЫВАТЬСЯ.
IN	ЧИСЛО	ОСУЩЕСТВЛЯЕТСЯ ВВОД НА УРОВНЕ МИКРОПРОЦЕССОРА ИЗ ПОРТА X ($0 \leq X \leq FFH$), ЗАГРУЖАЕТСЯ ПАРА РЕГИСТРОВ BC И ВЫПОЛНЯЕТСЯ КОМАНДА АССЕМБЛЕРА IN A(C).
INKEY $\times$	НЕТ	ЧТЕНИЕ С КЛАВИАТУРЫ, ВОЗВРАЩАЕТ СИМВОЛ ВВЕДЕННЫЙ С КЛАВИАТУРЫ (В РЕЖИМЕ [L] И [C]), ЕСЛИ БЫЛО ДЕЙСТВИТЕЛЬНОЕ НАЖАТИЕ КЛАВИШИ, ИЛИ ПУСТУЮ СТРОКУ В ПРОТИВНОМ СЛУЧАЕ.
INT	ЧИСЛО	ОКРУГЛЕНИЕ К БЛИЖАЙШЕМУ МЕНЬШЕМУ ЦЕЛОМУ
LEN	СТРОКА СИМВОЛОВ	ДЛИНА СТРОКИ
LN	ЧИСЛО	НАТУРАЛЬНЫЙ ЛОГАРИФМ, ВЫДАЕТ СООБЩЕНИЕ A, ЕСЛИ $X \leq 0$.
NOT	ЧИСЛО	0, ЕСЛИ $X \neq 0$, 1, ЕСЛИ $X = 0$. ОПЕРАЦИЯ ИМЕЕТ ЧЕТВЕРТЫЙ ПРИОРИТЕТ
OR	ЛОГИЧЕСКАЯ ОПЕРАЦИЯ. ОБА ОПЕРАНДА ЧИСЛА	$A \text{ OR } B = \begin{cases} 1, & \text{ЕСЛИ } B \neq 0 \\ A, & \text{ЕСЛИ } B = 0 \end{cases}$ ОПЕРАЦИЯ ИМЕЕТ ВТОРОЙ ПРИОРИТЕТ
PEEK	ЧИСЛО	ЗНАЧЕНИЕ БАЙТА В ПАМЯТИ ПО АДРЕСУ X, ОКРУГЛЕННОМУ К БЛИЖАЙШЕМУ ЦЕЛОМУ
PI	НЕТ	ЧИСЛО ПИ (3,14159265...)

POINT	ДВА ЧИСЛОВЫХ АРГУМЕНТА X И Y, ЗАКЛЮЧЕННЫХ В СКОБКИ	1, ЕСЛИ ТОЧКА ЭКРАНА С КООРДИНАТАМИ (X, Y) ЗАКРАШЕНА. 0, ЕСЛИ ЭТА ТОЧКА ИМЕЕТ ЦВЕТ ФОНА. ВЫДАЕТ СООБЩЕНИЕ B, ЕСЛИ НЕ ВЫПОЛНЯЮТСЯ УСЛОВИЯ $0 \leq X \leq 255$ И $0 \leq Y \leq 175$
RND	НЕТ	ОЧЕРЕДНОЕ ПСЕВДОСЛУЧАЙНОЕ ЧИСЛО ИЗ ПОСЛЕДОВАТЕЛЬНОСТИ, ПОЛУЧАЕМОЙ ВОЗВЕДЕНИЕМ В 75 СТЕПЕНЬ МОДУЛЯ ЧИСЛА 65537, ВЫЧИТАНИЕМ 1 И ДЕЛЕНИЕМ НА 65536. ЧИСЛО ЛЕЖИТ В ИНТЕРВАЛЕ $0 \leq Y \leq 1$.
SCREEN X	ДВА ЧИСЛОВЫХ АРГУМЕНТА X И Y, ЗАКЛЮЧЕННЫХ В СКОБКИ	СИМВОЛ (ОБЫЧНЫЙ ИЛИ ИНВЕРСНЫЙ), КОТОРЫЙ ПОЯВЛЯЕТСЯ НА ЭКРАНЕ В СТРОКЕ X, ПОЗИЦИИ Y. ДАЕТ ПУСТУЮ СТРОКУ, ЕСЛИ СИМВОЛ НЕ ОПОЗНАН.
SGN	ЧИСЛО	-1, ЕСЛИ $X < 0$ 0, ЕСЛИ $X = 0$ 1, ЕСЛИ $X > 0$
SIN	ЧИСЛО В РАДИАНАХ	СИНУС X
SQR	ЧИСЛО	КОРЕНЬ КВАДРАТНЫЙ. -ВЫДАЕТ СООБЩЕНИЕ A, ЕСЛИ $X < 0$.
STRX	ЧИСЛО	СТРОКА СИМВОЛОВ, КОТОРАЯ ДОЛЖНА БЫТЬ ОТОБРАЖЕНА, ЕСЛИ X ВЫВОДИТСЯ.
USR	ЧИСЛО	ВЫЗЫВАЕТ ПОДПРОГРАММУ В МАШИННЫХ КОДАХ, НАЧАЛЬНЫЙ АДРЕС КОТОРОЙ X. ПРИ ВОЗВРАТЕ РЕЗУЛЬТАТОМ БУДЕТ СОДЕРЖИМОЕ РЕГИСТРОВОЙ ПАРЫ BC.
USR	СТРОКА СИМВОЛОВ	АДРЕС ГРУППЫ БАЙТОВ, ЗАДАЮЩИХ ОПРЕДЕЛЕННЫЙ ПОЛЬЗОВАТЕЛЕМ СИМВОЛ ДЛЯ ЗАКРЕПЛЕНИЯ ЕГО ЗА X.
VAL	СТРОКА СИМВОЛОВ	ВЫЧИСЛЕНИЕ X КАК ЧИСЛОВОГО ВЫРАЖЕНИЯ. ВЫДАЕТ СООБЩЕНИЕ 'C', ЕСЛИ X СОДЕРЖИТ СИНТАКТИЧЕСКИЕ ОШИБКИ ИЛИ ДАЕТ СТРОКОВОЕ (НЕ ЧИСЛОВОЕ) ЗНАЧЕНИЕ. ВОЗМОЖНЫ И ДРУГИЕ ОШИБКИ.
VALX	СТРОКА СИМВОЛОВ	ВЫЧИСЛЯЕТ X КАК СТРОКОВОЕ ВЫРАЖЕНИЕ, ВЫДАЕТ СООБЩЕНИЕ C, ЕСЛИ X СОДЕРЖИТ СИНТАКТИЧЕСКУЮ ОШИБКУ ИЛИ ДАЕТ НЕ СТРОКОВОЕ (ЧИСЛОВОЕ) ВЫРАЖЕНИЕ.

ОПЕРАЦИИ

=====

ПРЕФИКСНЫЕ:

-	ЧИСЛО	ОТРИЦАТЕЛЬНОЕ ЗНАЧЕНИЕ
ИНФИКСНЫЕ	(ДВУХОПЕРАНДОВЫЕ):	
+	СЛОЖЕНИЕ	ДЛЯ ЧИСЕЛ, КОНКАТЕНЦИЯ ДЛЯ СТРОК
-	ВЫЧИТАНИЕ	
*	УМНОЖЕНИЕ	
/	ДЕЛЕНИЕ	
**	ВОЗВЕДЕНИЕ В СТЕПЕНЬ	(СТРЕЛКА ВВЕРХ), СООБЩЕНИЕ ЕСЛИ ЛЕВЫЙ ОПЕРАНД ОТРИЦАТЕЛЬНЫЙ
=	РАВЕНСТВО	А
>	БОЛЬШЕ	Е ОБА ОПЕРАНДА ДОЛЖНЫ БЫТЬ
<	МЕНЬШЕ	[ОДНОГО ТИПА, РЕЗУЛЬТАТ РАВЕН
>=	БОЛЬШЕ ИЛИ РАВНО	Е ЕСЛИ СРАВНЕНИЕ ИСТИННО И НЕ
<=	МЕНЬШЕ ИЛИ РАВНО	Е РАВНО 0, ЕСЛИ НЕТ.
<>	НЕ РАВНО	Г

ФУНКЦИИ И ОПЕРАЦИИ ИМЕЮТ СЛЕДУЮЩИЕ ПРИОРИТЕТЫ:

ИНДЕКСАЦИЯ И СЕЧЕНИЯ	-12
ВСЕ ФУНКЦИИ ЗА ИСКЛЮЧЕНИЕМ:	
NOT И ПРЕФИКСНОГО МИНУСА	-11
ВОЗВЕДЕНИЕ В СТЕПЕНЬ	-10
ПРЕФИКСНЫЙ МИНУС	- 9
*, /	- 8
+, -(ВЫЧИТАНИЕ)	- 6
=, >, <, <=, >=, < >	- 5
NOT	- 4
AND	- 3
OR	- 2

ОПЕРАТОРЫ

=====

ПРИНЯТЫЕ ОБОЗНАЧЕНИЯ:

- A - ОДНА БУКВА
- V - ПЕРЕМЕННАЯ
- X, Y, Z - ЧИСЛОВЫЕ ВЫРАЖЕНИЯ
- M, N - ЧИСЛОВЫЕ ВЫРАЖЕНИЯ, КОТОРЫЕ ОКРУГЛЯЮТСЯ К БЛИЖАЙШЕМУ ЦЕЛОМУ
- E - НЕКОТОРОЕ ВЫРАЖЕНИЕ
- F - ВЫРАЖЕНИЕ, ИМЕЮЩЕЕ СТРОКОВОЕ ЗНАЧЕНИЕ
- S - ПОСЛЕДОВАТЕЛЬНОСТЬ ОПЕРАТОРОВ, РАЗДЕЛЕННЫХ ДВОЕТОЧИЕМ
- C - ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ УПРАВЛЕНИЯ ЦВЕТОМ, КАЖДЫЙ ЗАКАНЧИВАЕТСЯ ЗАПЯТОЙ ИЛИ ТОЧКОЙ С ЗАПЯТОЙ, ЦВЕТОВОЙ СИМВОЛ ИМЕЕТ ФОРМУ ОПЕРАНДОВ

PAPER, INK, FLASH, BRIGHT, INVERSE или OVER.
ТЕКСТ ПРОИЗВОЛЬНОГО ВЫРАЖЕНИЯ МОЖЕТ РАСПОЛАГАТЬСЯ В ЛЮБОМ МЕСТЕ СТРОКИ (ЗА ИСКЛЮЧЕНИЕМ НОМЕРА СТРОКИ, КОТОРЫЙ ДОЛЖЕН РАЗМЕЩАТЬСЯ ВНАЧАЛЕ СТРОКИ).

ВСЕ ОПЕРАТОРЫ КРОМЕ INPUT, DEF и DATA МОГУТ ИСПОЛЬЗОВАТЬСЯ КАК КОМАНДЫ И В ПРОГРАММАХ.

КОМАНДА ИЛИ СТРОКА ПРОГРАММЫ МОЖЕТ СОДЕРЖАТЬ НЕСКОЛЬКО ОПЕРАТОРОВ, РАЗДЕЛЕННЫХ ДВОЕТОЧИЕМ (':').
НЕТ ОГРАНИЧЕНИЙ НА ПОЛОЖЕНИЕ ОПЕРАТОРА В СТРОКЕ, ХОТЯ ЕСТЬ НЕКОТОРЫЕ ОГРАНИЧЕНИЯ В IF и REM.

ВСЕ ОПЕРАТОРЫ ЯЗЫКА СВЕДЕНЫ В СЛЕДУЮЩУЮ ТАБЛИЦУ:

ОПЕРАТОР	ДЕЙСТВИЕ ОПЕРАТОРА
1	2
BEEP X,Y	ВОСПРОИЗВОДИТ ЗВУК ДЛИТЕЛЬНОСТЬЮ X СЕК И ВЫСОТОЙ Y ПОЛУТОНОВ ВВЕРХ ОТ ОСНОВНОГО ТОНА ДО (ИЛИ ВНИЗ, ЕСЛИ Y ОТРИЦАТЕЛЬНОЕ).
BORDER M	УСТАНАВЛИВАЕТ ЦВЕТ РАМКИ (БОРДЮРА) ЭКРАНА, ВЫДАЕТ СООБЩЕНИЕ ОБ ОШИБКЕ K, ЕСЛИ $0 > M > Y$,
BRIGHT M	УСТАНАВЛИВАЕТ ЯРКОСТЬ ВЫВОДИМОГО СИМВОЛА: 0 - ДЛЯ ОБЫЧНОЙ ЯРКОСТИ, 1 - ДЛЯ ПОВЫШЕННОЙ ЯРКОСТИ, 8 - СОХРАНЯЕТ СУЩЕСТВУЮЩУЮ ЯРКОСТЬ
CAT	БЕЗ MIKRODRIVE НЕ РАБОТАЕТ
CIRCLE X,Y,Z	ИЗОБРАЖАЕТ ДУГУ ИЛИ ОКРУЖНОСТЬ С ЦЕНТРОМ В ТОЧКЕ С КООРДИНАТАМИ (X,Y) И РАДИУСОМ Z,
CLEAR	УНИЧТОЖАЕТ ВСЕ ПЕРЕМЕННЫЕ И ОЧИЩАЕТ ЗАНИМАЕМУЮ НИМ ПАМЯТЬ, ВЫПОЛНЯЕТ RESTORE И CLS, УСТАНАВЛИВАЕТ PLOT ПОЗИЦИЮ В НИЖНЮЮ ЛЕВУЮ ТОЧКУ ЭКРАНА И ОЧИЩАЕТ GO SUB СТЕК,
CLEAR N	ПОДОБНО CLEAR, НО ДОПОЛНИТЕЛЬНО ИЗМЕНЯЕТ СИСТЕМНУЮ ПЕРЕМЕННУЮ RAMTOP НА "N" И ЗАДАЕТ НОВЫЙ GO SUB СТЕК,
CLOSE#	БЕЗ MIKRODRIVE НЕ РАБОТАЕТ,
CLS	(CLEAR SCREEN) ОЧИЩАЕТ ФАЙЛ ЭКРАНА,
CONTINUE	ПРОДОЛЖАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ, НАЧАТОЙ РАНЕЕ И ОСТАНОВЛЕННОЙ С СООБЩЕНИЕМ, ОТЛИЧНЫМ ОТ 0, ЕСЛИ БЫЛО СООБЩЕНИЕ 9 ИЛИ L, ТО ВЫПОЛНЕНИЕ

ПРОДОЛЖАЕТСЯ СО СЛЕДУЮЩЕГО ОПЕРАТОРА, В ДРУГИХ СЛУЧАЯХ С ТОГО ОПЕРАТОРА, ГДЕ СЛУЧИЛАСЬ ОШИБКА,

ЕСЛИ СООБЩЕНИЕ ВОЗНИКЛО В КОМАНДНОЙ СТРОКЕ, ТО CONTINUE ВЫЗОВЕТ ПОПЫТКУ ПОВТОРИТЬ КОМАНДНУЮ СТРОКУ И ПЕРЕИДЕТ В ЦИКЛ, ЕСЛИ БЫЛО СООБЩЕНИЕ 0:1, ДАЕТ СООБЩЕНИЕ 0, ЕСЛИ БЫЛО 0:2, ИЛИ ДАЕТ СООБЩЕНИЕ N, ЕСЛИ БЫЛО 0:3 ИЛИ БОЛЕЕ, В КАЧЕСТВЕ CONTINUE ИСПОЛЬЗУЕТСЯ КЛЮЧЕВОЕ СЛОВО CONT НА КЛАВИАТУРЕ,

COPY

ПЕРЕСЫЛАЕТ КОПИЮ 22 СТРОК ЭКРАНА НА ПРИНТЕР, ЕСЛИ ОН ПОДКЛЮЧЕН, ПОМНИТЕ, ЧТО ПО COPY НЕЛЬЗЯ РАСПЕЧАТАТЬ НАХОДЯЩИЙСЯ НА ЭКРАНЕ АВТОМАТИЧЕСКИЙ ЛИСТИНГ, ВЫДАЕТ СООБЩЕНИЕ D, ЕСЛИ НАЖАТЬ КЛАВИШУ BREAK.

DATA E1,E2,E3,...

ЧАСТЬ СПИСКА ДАННЫХ, ДОЛЖНА РАСПОЛАГАТЬСЯ В ПРОГРАММЕ,

DEF FNA(A1,A2,...AK)=E

ОПРЕДЕЛЯЕМАЯ ПОЛЬЗОВАТЕЛЕМ ФУНКЦИЯ ДОЛЖНА РАСПОЛАГАТЬСЯ В ПРОГРАММЕ A,A1,A2 И Т.Д. ЕДИНСТВЕННЫЕ БУКВЫ ИЛИ БУКВЫ И * ДЛЯ СТРОКОВЫХ АРГУМЕНТОВ, ЗНАЧЕНИЙ, ИСПОЛЬЗУЕТСЯ ФОРМА DEF FNA(), ЕСЛИ НЕТ АРГУМЕНТА,

DELETE F

БЕЗ MICRODRIVE НЕ РАБОТАЕТ

DIM A(N1,N2,...NK)

УНИЧТОЖАЕТ МАССИВ С ИМЕНЕМ "A" И УСТАНОВЛИВАЕТ ЧИСЛОВЫЙ МАССИВ "A" С "K" ИЗМЕРЕНИЯМИ И ПРИСВАИВАЕТ ВСЕМ ЕГО ЭЛЕМЕНТАМ ЗНАЧЕНИЕ 0,

DIM A*(N1,N2,...NK)

УНИЧТОЖАЕТ МАССИВ ИЛИ СТРОКУ С ИМЕНЕМ "A*" И УСТАНОВЛИВАЕТ СИМВОЛЬНЫЙ МАССИВ С "K" ИЗМЕРЕНИЯМИ И ПРИСВАИВАЕТ ВСЕМ ЕГО ЭЛЕМЕНТАМ ЗНАЧЕНИЯ " ". МАССИВ ВОЖЕТ БЫТЬ ПРЕДСТАВЛЕН КАК МАССИВ СТРОК ФИКСИРОВАННОЙ ДЛИНЫ NK, С K-1 РАЗМЕРНОСТЬЮ, СООБЩЕНИЕ 4 ВЫДАЕТСЯ, ЕСЛИ НЕДОСТАТОЧНО МЕСТА ДЛЯ РАЗМЕЩЕНИЯ МАССИВА, МАССИВ НЕ ОПРЕДЕЛЕН ДО ОПИСАНИЯ В ОПЕРАТОРЕ DIM,

DRAW X,Y

ТО ЖЕ САМОЕ ЧТО И DRAW X,Y,0 ЧЕРТИТ ПРЯМУЮ ЛИНИЮ

DRAW X,Y,Z

ИЗОБРАЖАЕТ ЛИНИЮ ОТ ТЕКУЩЕЙ ГРАФИЧЕСКОЙ ПОЗИЦИИ В ТОЧКУ С ПРИРАЩЕНИЯМИ X,Y ПО ДУГЕ В Z РАДИАН, ВЫДАЕТ СООБЩЕНИЕ B ПРИ ВЫХОДЕ ЗА ПРЕДЕЛЫ ЭКРАНА,

ERAZE

БЕЗ MICRODRIVE НЕ РАБОТАЕТ

FLASH N

ОПРЕДЕЛЯЕТ, БУДЕТ ЛИ СИМВОЛ МЕРЦАЮЩИМ ИЛИ С ПОСТОЯННЫМ СВЕЧЕНИЕМ, N=1 ДЛЯ МЕРЦАНИЯ, N=8, ДЛЯ СОХРАНЕНИЯ ПРЕДЫДУЩЕГО СОСТОЯНИЯ, N=0 ДЛЯ ПОСТОЯННОГО СВЕЧЕНИЯ,

FOR A=X TO Y

FOR A=X TO Y STEP 1

FOR A=X TO Y STEP Z

УНИЧТОЖАЕТ СКАЛЯРНУЮ ПЕРЕМЕННУЮ "A" И УСТАНОВЛИВАЕТ УПРАВЛЯЮЩУЮ ПЕРЕМЕННУЮ "X", ПРЕДЕЛ "Y", ШАГ ПРИРАЩЕНИЯ "Z", ЗАЦИКЛИВАЕТ АДРЕС УКАЗАННЫЙ В УТВЕРЖДЕНИИ ПОСЛЕ FOR ОПЕРАТОРА. ПРОВЕРЯЕТ ЕСЛИ НАЧАЛЬНОЕ ЗНАЧЕНИЕ БОЛЬШЕ (ЕСЛИ STEP >=0) ИЛИ МЕНЬШЕ (ЕСЛИ STEP < 0), ЧЕМ ПРЕДЕЛ, ТО ПРОИСХОДИТ ПЕРЕХОД К УТВЕРЖДЕНИЮ NEXT A ИЛИ ВЫДАЧА СООБЩЕНИЯ 1, ЕСЛИ НЕТ (СМ. NEXT), СООБЩЕНИЕ 4 ВЫДАЕТСЯ, ЕСЛИ НЕДОСТАТОЧНО МЕСТА ДЛЯ РАЗМЕЩЕНИЯ УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ,

FORMAT F

БЕЗ MIKRODRIVE НЕ РАБОТАЕТ

GO SUB N

ПРОТАЛКИВАЕТ СТРОКУ С ОПЕРАТОРОМ GO SUB В СТЕК ДЛЯ ИСПОЛЬЗОВАНИЯ, ЗАТЕМ КАК GO TO N ВЫДАЕТСЯ СООБЩЕНИЕ 4, ЕСЛИ НЕ ВСЕ ПРОГРАММЫ ЗАВЕРШИЛИСЬ С RETURN.

GO TO N

ПРОДОЛЖАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ СО СТРОКИ N. ЕСЛИ N ОПУЩЕНО, ТО С ПЕРВОЙ СТРОКИ ПОСЛЕ ЭТОЙ.

IF X THEN S

ЕСЛИ 'X' ИСТИННО (НЕ РАВНО 0), ТО ВЫПОЛНЯЕТСЯ 'S'. 'S' ВКЛЮЧАЕТ ВСЕ ОПЕРАТОРЫ ДО КОНЦА СТРОКИ. ФОРМА 'IF X THEN НОМЕР СТРОКИ' НЕДОПУСТИМА

INK N

УСТАНОВЛИВАЕТ ЦВЕТ ЗАКРАШИВАНИЯ (Т.Е. ЦВЕТ, КОТОРЫМ БУДУТ ИЗОБРАЖАТЬСЯ СИМВОЛЫ НА ЦВЕТЕ ФОНА). 'N' В ИНТЕРВАЛЕ ОТ 0 ДО 7 УКАЗЫВАЕТ ЦВЕТ. N=8-ОСТАВИТЬ ЦВЕТ БЕЗ ИЗМЕНЕНИЯ, N=9-УВЕЛИЧЕНИЕ КОНТРАСТА. ВЫДАЕТ СООБЩЕНИЕ K, ЕСЛИ 'N' НЕ ЛЕЖИТ В ИНТЕРВАЛЕ ОТ 0 ДО 9.

INPUT ...

ГДЕ '...' ЕСТЬ ПОСЛЕДОВАТЕЛЬНОСТЬ ВВОДИМЫХ СИМВОЛОВ, РАЗДЕЛЯЕМЫХ КАК В ОПЕРАТОРЕ PRINT ЗАПЯТЫМИ, ТОЧКАМИ С ЗАПЯТОЙ ИЛИ АПОСТРОФАМИ ВВОДИМЫМИ СИМВОЛАМИ МОГУТ БЫТЬ:

- А) НЕКОТОРЫЙ PRINT-СИМВОЛ, НАЧИНАЮЩИЙСЯ НЕ С БУКВЫ
- Б) ИМЯ ПЕРЕМЕННОЙ

В) СТРОКА ИМЕН ПЕРЕМЕННЫХ СТРОКОВОГО ТИПА.

PRINT-СИМВОЛЫ В СЛУЧАЕ А) ПРЕДСТАВЛЯЮТСЯ ТАК ЖЕ, КАК И В ОПЕРАТОРЕ PRINT, ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО ОНИ ВСЕ ВЫВОДЯТСЯ В НИЖНЮЮ ЧАСТЬ ЭКРАНА. В СЛУЧАЕ Б) КОМПЬЮТЕР ОСТАНАВЛИВАЕТСЯ И ЖДЕТ ВВОДА НЕКОТОРОГО ВЫРАЖЕНИЯ С КЛАВИАТУРЫ, ЗНАЧЕНИЕ КОТОРОГО БУДЕТ ПРИСВОЕНО ПЕРЕМЕННОЙ, ВВОД ОСУЩЕСТВЛЯЕТСЯ ОБЫЧНЫМ ОБРАЗОМ, А СИНТАКТИЧЕСКИЕ ОШИБКИ ВЫДАЮТСЯ МЕРЦАЮЩИМ [?]. ДЛЯ СТРОКОВОГО ВЫРАЖЕНИЯ ВВОДНОЙ БУФЕР УСТАНОВЛИВАЕТСЯ ДЛЯ РАЗМЕЩЕНИЯ ДВУХ ТАКИХ СТРОК (КОТОРЫЙ ПРИ НЕОБХОДИМОСТИ МОЖЕТ БЫТЬ УВЕЛИЧЕН), ЕСЛИ ПЕРВЫЙ ВВОДИМЫЙ СИМВОЛ STOP, ТО ПРОГРАММА ОСТАНАВЛИВАЕТСЯ С СООБЩЕНИЕМ Н. СЛУЧАЙ В) ПОДОБЕН СЛУЧАЮ Б) С ТОЙ ЛИШЬ РАЗНИЦЕЙ, ЧТО ВВОДИМАЯ ИНФОРМАЦИЯ ПРЕДСТАВЛЯЕТ СОБОЙ СТРОКОВЫЙ ЛИТЕРАЛ НЕОГРАНИЧЕННОЙ ДЛИНЫ. STOP В ЭТОМ СЛУЧАЕ НЕ СРАБОТАЕТ. ДЛЯ ОСТАНОВА ВЫ ДОЛЖНЫ НАЖАТЬ КЛАВИШУ 'КУРСОР ВНИЗ'

INVERSE N

СИМВОЛ УПРАВЛЕНИЯ ИНВЕРСИЕЙ ВЫВОДИМОГО СИМВОЛА, ЕСЛИ N=0. СИМВОЛ ВВОДИТСЯ В ОБЫЧНОМ ВИДЕ С ПРОРИСОВКОЙ ЦВЕТОМ ЗАКРАШИВАНИЯ (INK) И ЦВЕТЕ ФОНА (PAPER), ЕСЛИ N=1, ЦВЕТОВОЕ РЕШЕНИЕ ИЗОБРАЖЕНИЯ СИМВОЛА МЕНЯЕТСЯ НА ОБРАТНОЕ, СМОТРИ ПРИЛОЖЕНИЕ В, ВЫДАЕТ СООБЩЕНИЕ К ЕСЛИ 'N' НЕ 0 ИЛИ 1.

LET V=E

ПРИСВАИВАЕТ ЗНАЧЕНИЕ 'E' ПЕРЕМЕННОЙ 'V'. КЛЮЧЕВОЕ СЛОВО LET НЕ МОЖЕТ БЫТЬ ОПУЩЕНО. СКАЛЯРНАЯ ПЕРЕМЕННАЯ НЕ ОПРЕДЕЛЕНА, ПОКА НЕ ВСТРЕТИТСЯ В ОПЕРАТОРАХ LET, READ ИЛИ INPUT. ЕСЛИ 'V' ИНДЕКСИРУЕМАЯ СТРОКОВАЯ ПЕРЕМЕННАЯ ИЛИ СЕЧЕНИЕ СТРОКОВОГО МАССИВА (ПОДСТРОКА), ТО ПРИСВАИВАНИЕ ОСУЩЕСТВЛЯЕТСЯ С УСЕЧЕНИЕМ СПРАВА ИЛИ ДОПОЛНЕНИЕМ ПРОБЕЛАМИ ДО ФИКСИРОВАННОЙ ДЛИНЫ.

LIST

ТО ЖЕ, ЧТО И LIST 0.

LIST N

ЗАПИСЫВАЕТ ТЕКСТ ПРОГРАММЫ В ВЕРХНЮЮ ЧАСТЬ ЭКРАНА, НАЧИНАЯ С ПЕРВОЙ СТРОКИ, МЕНЬШЕЙ, ЧЕМ 'N', И ДЕЛАЕТ 'N' ТЕКУЩЕЙ СТРОКОЙ.

LLIST

ТО ЖЕ, ЧТО И LLIST 0.

LLIST N	ПОДОБНО LIST, НО ВЫВОД ОСУЩЕСТВЛЯЕТСЯ НА ПРИНТЕР.
LOAD F	ЗАГРУЖАЕТ ПРОГРАММУ И ПЕРЕМЕННЫЕ
LOAD F DATA ()	ЗАГРУЖАЕТ ЧИСЛОВОЙ МАССИВ
LOAD F DAT*()	ЗАГРУЖАЕТ СТРОКОВЫЙ МАССИВ
LOAD F CODE M,N	ЗАГРУЖАЕТ СТАРШИЕ 'N' БАЙТОВ, НАЧИНАЯ С АДРЕСА 'M'.
LOAD F CODE M	ЗАГРУЖАЕТ БАЙТЫ, НАЧИНАЯ С АДРЕСА 'M'.
LOAD F CODE	ЗАГРУЖАЕТ БАЙТЫ ПО ТОМУ ЖЕ АДРЕСУ С КОТОРОГО ОНИ БЫЛИ РАЗГРУЖЕНЫ
LOAD F SCREEN*	АНАЛОГИЧНО LOAD F CODE 16384,6912. ОЧИЩАЕТ ФАЙЛ ЭКРАНА И ЗАГРУЖАЕТ С КАССЕТНОГО МАГНИТОФОНА. СМОТРИ ГЛАВУ 20.
LPRINT	ПОДОБНО PRINT, НО ИСПОЛЬЗУЕТ ПРИНТЕР.
MERGE F	ПОДОБНО LOAD F, НО НЕ ЗАТИРАЕТ ВСЮ СТАРУЮ ПРОГРАММУ В ПАМЯТИ, А ЗАМЕНЯЕТ ТОЛЬКО ТЕ СТРОКИ И ПЕРЕМЕННЫЕ, У КОТОРЫХ СОВПАДАЮТ НОМЕРА ИЛИ ИМЕНА С ТАКИМИ ЖЕ НА ЛЕНТЕ.
MOVE F1,F2	БЕЗ MIKRODRIVE НЕ РАБОТАЕТ.
NEW	ЗАПУСКАЕТ ПО НОВОЙ СИСТЕМУ ПРОГРАММИРОВАНИЯ БЕЙСИК, УНИЧТОЖАЯ СТАРУЮ ПРОГРАММУ И ПЕРЕМЕННЫЕ И ИСПОЛЬЗУЕМУЮ ПАМЯТЬ, ВКЛЮЧАЯ И БАЙТ АДРЕСА В СИСТЕМНОЙ ПЕРЕМЕННОЙ RAMBOT, НО СОХРАНЯЕТ СИСТЕМНЫЕ ПЕРЕМЕННЫЕ UDG, P RAMT, RASP И PIP.
NEXT A	А) НАХОДИТ УПРАВЛЯЮЩУЮ ПЕРЕМЕННУЮ 'A' Б) ПРИБАВЛЯЕТ К НЕЙ ЗНАЧЕНИЕ STEP, В) ЕСЛИ STEP>=0, А ЗНАЧЕНИЕ 'A' СТАЛО БОЛЬШЕ ЗНАЧЕНИЯ 'ПРЕДЕЛ', ИЛИ STEP<0, А ЗНАЧЕНИЕ 'A' МЕНЬШЕ ЧЕМ ЗНАЧЕНИЕ 'ПРЕДЕЛ', ТО ПРОИСХОДИТ ПЕРЕХОД К ОПЕРАТОРУ ЦИКЛА. СООБЩЕНИЕ 2 ВЫДАЕТСЯ, ЕСЛИ НЕ НАЙДЕНА ПЕРЕМЕННАЯ 'A'. СООБЩЕНИЕ 1 ВЫДАЕТСЯ, ЕСЛИ 'A' НЕ ЯВЛЯЕТСЯ УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ ЦИКЛА.
OPEN#	БЕЗ MIKRODRIVE НЕ РАБОТАЕТ

OUT M,N

ВЫВОДИТ БАЙТ 'N' В ПОРТ 'M'. ОПЕРАЦИЯ ВЫПОЛНЯЕТСЯ НА УРОВНЕ МИКРОПРОЦЕССОРА (ЗАГРУЖАЕТ В РЕГИСТРОВУЮ ПАРУ ВО АДРЕС 'M', А В РЕГИСТР A - 'N' И ВЫПОЛНЯЕТ КОМАНДУ АССЕМБЛЕРА OUT (C),A).

$0 \leq M \leq 65535$, $-255 \leq N \leq 255$, ИНАЧЕ ВЫДАЕТСЯ СООБЩЕНИЕ В.

OVER N

УПРАВЛЯЮЩИЙ СИМВОЛ НАДПЕЧАТЫВАНИЯ ПО ВЫВЕДЕННОЙ СТРОКЕ, ЕСЛИ $N=0$, ТО ВЫВОДИМЫЙ СИМВОЛ ЗАТИРАЕТ СУЩЕСТВУЮЩИЙ В ДАННОЙ ПОЗИЦИИ, ЕСЛИ $N=1$, ТО НОВЫЙ СИМВОЛ СОЕДИНЯЕТСЯ СО СТАРЫМ, ОБРАЗУЯ ЗАКРАШИВАЮЩИЙ ЦВЕТ, ПРИ УСЛОВИИ, ЧТО СТАРЫЙ СИМВОЛ ИМЕЛ УКАЗАНИЕ ЦВЕТА ОТЛИЧНОЕ ОТ НОВОГО, ИЛИ ЦВЕТ ФОНА, ЕСЛИ ОБА УКАЗЫВАЮТ НА ОДИН И ТОТ ЖЕ ЦВЕТ (ЛИБО ФОНА, ЛИБО ЗАКРАШИВАНИЯ. СЛОЖЕНИЕ ПО МОДУЛЮ 2). СМОТРИ ПРИЛОЖЕНИЕ В.

PAPER N

ПОДОБЕН INK, НО УПРАВЛЯЕТ ЦВЕТОМ ФОНА.

PAUSE N

ОСТАНАВЛИВАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ И ЗАДЕРЖИВАЕТ ИЗОБРАЖЕНИЕ НА ЭКРАНЕ НА 'N' КАДРОВ (50 КАДРОВ В СЕК. - ЧАСТОТА КАДРОВОЙ РАЗВЕРТКИ) ИЛИ НАЖАТИЯ ЛЕВОЙ КЛАВИШИ.
 $0 \leq N \leq 65535$, ИНАЧЕ ВЫДАЕТСЯ СООБЩЕНИЕ В, ПРИ $N=0$ ВРЕМЯ ЗАДЕРЖКИ НЕ УЧИТЫВАЕТСЯ И ПРОДОЛЖАЕТСЯ ДО ПЕРВОГО НАЖАТИЯ КЛАВИШИ.

PLOT C:M,N

ВЫВОДИТ ТОЧКУ ЗАКРАШИВАЮЩЕГО ЦВЕТА (ОБРАБОТАННАЯ OVER И INVERSE) С КООРДИНАТАМИ (ABS(M),ABS(N)), СМЕЩАЯ ГРАФИЧЕСКУЮ (PLOTPOSITION) ПОЗИЦИЮ ЕСЛИ ЦВЕТОВОЙ СИМВОЛ 'C' НЕ СПЕЦИФИЦИРОВАН ИНАЧЕ, ТО ЦВЕТ ЗАКРАШИВАНИЯ В ПОЗИЦИИ, ГДЕ РАСПОЛОЖЕНА ЭТА ТОЧКА ИЗМЕНЯЕТСЯ НА ТЕКУЩИЙ СИЛОВОЙ ЗАКРАШИВАЮЩИЙ ЦВЕТ, И ДРУГИЕ УКАЗАНИЯ (ЦВЕТ ФОНА, МЕРЦАНИЕ, ЯРКОСТЬ) ОСТАЮТСЯ БЕЗ ИЗМЕНЕНИЙ.
 $0 \leq \text{ABS}(M) \leq 255$, $0 \leq \text{ABS}(N) \leq 175$, ИНАЧЕ СООБЩЕНИЕ В.

POKE M,N

ЗАПИСЫВАЕТ ЗНАЧЕНИЕ 'N' В БАЙТ ПАМЯТИ ПО АДРЕСУ 'M'.
 $0 \leq M \leq 65535$,
 $-255 \leq N \leq 255$, ИНАЧЕ СООБЩЕНИЕ В.

PRINT....

ГДЕ '....' ПОСЛЕДОВАТЕЛЬНОСТЬ PRINT СИМВОЛОВ, РАЗДЕЛЕННЫХ ЗАПЯТЫМИ ИЛИ ТОЧКАМИ С ЗАПЯТОЙ ИЛИ АПОСТРОФОМ, КОТОРЫЕ ВЫВОДЯТСЯ В ЭКРАННЫЙ ФАЙЛ

ДЛЯ ОТОБРАЖЕНИЯ НА ЭКРАНЕ ТЕЛЕВИЗОРА ТОЧКА С ЗАПЯТОЙ САМА ДЕЙСТВИИ НЕ ВЫЗЫВАЕТ, А ИСПОЛЬЗУЕТСЯ ДЛЯ РАЗГРАНИЧЕНИЯ СИМВОЛОВ, ЗАПЯТАЯ РОЖДАЕТ УПРАВЛЯЮЩИЙ СИМВОЛ 'ЗАПЯТАЯ', А АПОСТРОФ ПОРОЖДАЕТ СИМВОЛ ENTER. В КОНЦЕ ОПЕРАТОРА PRINT, ЕСЛИ ОН ЗАКАНЧИВАЕТСЯ ТОЧКОЙ С ЗАПЯТОЙ, ЗАПЯТОЙ ИЛИ АПОСТРОФОМ, АВТОМАТИЧЕСКИ ВЫВОДИТСЯ СИМВОЛ ENTER.

PRINT-СИМВОЛОМ МОЖЕТ БЫТЬ:

А) ПУСТАЯ СТРОКА (Т.Е. НИЧЕГО)

Б) ЧИСЛОВОЕ ВЫРАЖЕНИЕ.

ЕСЛИ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ ОТРИЦАТЕЛЬНО ТО ВЫВОДИТСЯ ЗНАК МИНУС.

ЕСЛИ $X \leq 10^{**}(-5)$ ИЛИ $X \geq 10^{**}(13)$ ВЫВОД ОСУЩЕСТВЛЯЕТСЯ В ПОКАЗАТЕЛЬНОЙ ФОРМЕ, МАНТИСА ПРЕДСТАВЛЯЕТСЯ 8-МЬЮ ЦИФРАМИ (С НОРМАЛИЗАЦИЕЙ) И ДЕСЯТИЧНОЙ ТОЧКОЙ (ОТСУТСТВУЕТ ТОЛЬКО ТОГДА, КОГДА В МАНТИСЕ ОДНА ЦИФРА) ПОСЛЕ ПЕРВОЙ ЦИФРЫ, ПОКАЗАТЕЛЬ СТЕПЕНИ ЗАПИСЫВАЕТСЯ ПОСЛЕ С ПОСЛЕДУЮЩИМ ЗНАКОМ И ДВУМЯ ЦИФРАМИ ПОРЯДКА, ИНАЧЕ X ВЫВОДИТСЯ КАК ОБЫЧНОЕ ДЕСЯТИЧНОЕ ЧИСЛО С 8-МЬЮ ЗНАЧАЩИМИ ЦИФРАМИ.

В) СТРОКОВОЕ ВЫРАЖЕНИЕ.

В СТРОКЕ ВОЗМОЖНЫ ПРОБЕЛЫ ДО И ПОСЛЕ СИМВОЛОВ. УПРАВЛЯЮЩИЕ СИМВОЛЫ ВЫЗЫВАЮТ ОПРЕДЕЛЯЕМОЕ ИМИ ДЕЙСТВИЕ, НЕ ОТРАЖАЕМЫЕ НА ЭКРАНЕ СИМВОЛЫ ВЫВОДЯТСЯ КАК '7'.

Г) АТ М, N

ВЫВОД В СТРОКУ 'М', ПОЗИЦИЮ 'N'.

Д) ТАВ N

ВЫВОД УПРАВЛЯЮЩЕГО СИМВОЛА ТАВ С ПОСЛЕДУЮЩИМИ ДВУМЯ БАЙТАМИ 'N' (ПЕРВЫЙ БАЙТ СТАРШИЙ), ВЫЗЫВАЕТ ТАВ-ОСТАНОВ.

Е) ЦВЕТОВОЙ СИМВОЛ В ФОРМЕ PAPER, INK, FLASH, BRIGHT, INVERSE ИЛИ OVER ОПЕРАТОРА.

RANDOMIZE

RANDOMIZE N

ТО ЖЕ, ЧТО И RANDOMIZE 0

УСТАНОВЛИВАЕТ СИСТЕМНУЮ ПЕРЕМЕННУЮ SEED, ИСПОЛЬЗУЕМУЮ ДЛЯ ВЫЧИСЛЕНИЯ ОЧЕРЕДНОГО ЗНАЧЕНИЯ ФУНКЦИИ RND. ЕСЛИ $N \neq 0$, ТО SEED ПРИНИМАЕТ ЗНАЧЕНИЕ N. ЕСЛИ $N=0$, ТО SEED ПРИНИМАЕТ ЗНАЧЕНИЕ ДРУГОЙ СИСТЕМНОЙ ПЕРЕМЕННОЙ F.LAMES, ПОДСЧИТЫВАЮЩЕЙ КАДРЫ, ОТОБРАЖАЕМЫЕ НА ЭКРАНЕ, ЧТО ОБЕСПЕЧИВАЕТ ВПОЛНЕ СЛУЧАЙНОЕ ЧИСЛО. ОПЕРАТОР ЗАПУСКАЕТ СОКРАЩЕНИЕ RAND (СМ. КЛАВИШУ), ВЫДАЕТ СООБЩЕНИЕ В, ЕСЛИ 'N'

READ V1, V2, ...VK	НЕ ЛЕЖИТ В ИНТЕРВАЛЕ ОТ 0 ДО 65535, ПРИСВАИВАЕТ ПЕРЕМЕННЫМ ОДНОЙ ЗА ДРУ- ГОЙ ЗНАЧЕНИЯ ПОСЛЕДОВАТЕЛЬНО ПРЕД- СТАВЛЕННЫЕ В СПИСКЕ DATA.
REM...	НЕ ВЫПОЛНЯЕТСЯ. '...' МОЖЕТ БЫТЬ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ (ИСКЛЮ- ЧАЯ ENTER), МОЖЕТ ВКЛЮЧАТЬ ДВОЕТО- ЧИЕ (":") ДЛЯ УКАЗАНИЯ ОТСУТСТВИЯ ОПЕРАТОРОВ В СТРОКЕ С REM.
RESTORE	ТО ЖЕ САМОЕ, ЧТО И В RESTORE 0.
RESTORE N	ПЕРЕЗАПИСЫВАЕТ УКАЗАТЕЛЬ ДАННЫХ В ПЕРВЫЙ ОПЕРАТОР DATA В СТРОКЕ С МЕНЬШИМ, ЧЕМ 'N'. СЛЕДУЮЩИЙ ОПЕРАТОР READ НАЧНЕТ СЧИТЫВАНИЕ ОТСЮДА.
RETURN	ССЫЛАЕТСЯ НА ОПЕРАТОР GO SUB В СТЕКЕ И ПЕРЕДАЕТ УПРАВЛЕНИЕ НА СТРОКУ ПОС- ЛЕ НЕГО, ВЫДАЕТ СООБЩЕНИЕ 7, ЕСЛИ НЕТ УКАЗЫВАЕМОГО ОПЕРАТОРА В СТЕКЕ, ХАРАКТЕРНАЯ ОШИБКА, КОГДА ОПЕРАТОРЫ GO SUB НЕ СБАЛАНСИРОВАННЫ ОПЕРАТО- РАМИ RETURN.
RUN	ТО ЖЕ САМОЕ, ЧТО И RUN 0.
RUN N	CLEAR, И ЗАТЕМ GO TO N
SAVE F	ЗАПИСЫВАЕТ НА ЛЕНТУ ПРОГРАММУ И ПЕРЕМЕННЫЕ.
SAVE F LINE N	ЗАПИСЫВАЕТ НА ЛЕНТУ ПРОГРАММУ И ПЕ- РЕМЕННЫЕ ТАКИМ ОБРАЗОМ, ЧТО ПРИ ЗАГ- РУЗКЕ ПРОГРАММА АВТОМАТИЧЕСКИ ВЫПОЛ- НЯЕТСЯ СО СТРОКИ 'N'
SAVE F DATA ()	ЗАПИСЬ НА ЛЕНТУ ЧИСЛОВОГО МАССИВА
SAVE F DATA* ()	ЗАПИСЬ НА ЛЕНТУ СТРОКОВОГО МАССИВА *
SAVE F CODEM, N	ЗАПИСЫВАЕТ НА ЛЕНТУ 'N' БАЙТОВ НАЧИ- НАЯ С АДРЕСА 'M'.
SAVE F SCREEN*	АНАЛОГИЧНО SAVE F CODE 16384, 6912. ВЫДАЕТ СООБЩЕНИЕ F, ЕСЛИ 'F' ПУСТАЯ СТРОКА ИЛИ ИМЕЕТ ДЛИНУ БОЛЕЕ 10, СМОТРИ ГЛАВУ 20
STOP	ОСТАНАВЛИВАЕТ ВЫПОЛНЕНИЕ ПРОГРАММ С ВЫДАЧЕЙ СООБЩЕНИЯ 9, CONTINUE (ПРО- ДОЛЖЕНИЕ) БУДЕТ ОСУЩЕСТВЛЯТЬСЯ СО СЛЕДУЮЩЕГО ОПЕРАТОРА

VERIFY

ТО ЖЕ, ЧТО И LOAD. ЗА ИСКЛЮЧЕНИЕМ
ТОГО, ЧТО ДАННЫЕ ЗАГРУЖАЮТСЯ В ОЗУ,
НО СРАВНИВАЮТСЯ С НАХОДЯЩИМИСЯ ТАМ.
ВЫДАЕТСЯ СООБЩЕНИЕ В, ЕСЛИ ОБНАРУ-
ЖИТСЯ ХОТЯ БЫ ОДИН НЕ СОВПАДАЮЩИЙ
БАЙТ.

ПРИЛОЖЕНИЕ D

ПРИМЕРЫ ПРОГРАММ

=====

ЭТО ПРИЛОЖЕНИЕ СОДЕРЖИТ НЕКОТОРЫЕ ПРИМЕРЫ ПРОГРАММ, ДЕМОНИ-
СТРИРУЮЩИЕ ВОЗМОЖНОСТИ ZX SPECTRUM.

ПЕРВАЯ ИЗ ЭТИХ ПРОГРАММ ТРЕБУЕТ ВВЕСТИ ДАТУ И ДАЕТ ДНИ
НЕДЕЛИ, КОТОРЫЕ СООТВЕТСТВУЮТ ЭТОЙ ДАТЕ.

```

10 REM KONVERT DATE TO DAY
20 DIM D$(7,6):REM DAYS OF WEEK
30 FOR N=1 TO 7: READ D$(N): NEXT N
40 DIM M(12):REM LENGTHS OF MONTHS
50 FOR N=1 TO 12: READ M(N):NEXT N
100 REM INPUT DATE
110 INPUT "DAY?":DAY
120 INPUT "WONTH?":WONTH
130 INPUT " YEAR (20TH CENTURY ONLY) ?":YEAR
140 IF YEAR<1901 THEN PRINT "20TH CENTURY STARTS AT 1901":
    GO TO 100
150 IF YEAR>2000 THEN PRINT " 20TH CENTURY ENDS AT 2000":
    GO TO 120
160 IF MONTH <1 THEN GO TO 210
170 IF MONTH>12 THEN GO TO 210
180 IF YEAR/4-INT(YEAR/4)=0 THEN LET M(2)=29: REM LEAP YEAR
190 IF DAY>M(MONTH) THEN PRINT "THIS MONTH HAS ONLY":
    M(MONTH):"DAYS": GO TO 500
200 IF DAY>0 THEN GO TO 300
210 PRINT " STUFF AND NONSENSE, GIVE ME A REAL DATE."
220 GO TO 500
300 REM CONVERT DATE TO NUMBER OF DAYS SINCE START OF CENTUR
310 LET Y=YEAR-1901
320 LET B=365*Y+INT(Y/4):REM NUMBER OF DAUS TO START OF YEAR
330 FOR N=1 TO MONTH-1: REM ADD ON PREVIOUS MONTH
340 LET B=B+M(N):NEXT N
350 LET B=B+DAY
400 REM SONVERT TO DAY OF WEEK
410 LET B=B-7*INT(B/7)+1
420 PRINT DAY "/" MONTH "/" YEAR
430 FOR N=6 TO 3 STEP -1:REM REMOVE TRAILING SPACES
440 IF D$(B,N) <>" " THEN GO TO 460
450 NEXT N
460 LET E$=D$(B,TON)
470 PRINT "IS A" E$ "DAY"
500 LET M(2)=28:REM RESTORE FEBRUARY
510 INPUT "AGAIN",A$
520 IF A$="N" THEN GO TO 540
530 IF A$<>"N" THEN GO TO 100
1000 REM DAYS OF WEEK
1010 DATA "MON","TUES","WEDNES"
1020 DATA "THURS","FRI","SATUR","SUN"
ЭТА ПРОГРАММА УСТАНОВЛИВАЕТ СООТВЕТСТВИЕ МЕЖДУ ЯРДОМ, ФУТОМ И
ДЮЙМОМ:
10 INPUT "YARDS?","YD","FEET","FT","INCHES",IN
40 GO SUB 2000:REM PRINT THE VALUES
50 PRINT " = "
70 GO SUB 1000: REM THE ADJUSTMENT
80 GO SUB 2000: REM PRINT THE ADJUSTED VALUES

```

```

90 PRINT
100 GO TO 10
1000 REM SUBROUTINE TO ADJUST YD,FT,IN TO THE NORMAL FORM FOR
    YARDS, FEET AND INCHES
1010 LET IN=35*YD+12*FT+IN:REM NOW EVERYTHING IS IN INCHES
1030 LET S=SGN IN: LET IN=ABS IN: REM WE WORK WITH IN
    POSITIVE, HOLDING ITS SIGN IN S
1060 LET FT=INT(IN/12): LET IN=(IN-12*FT)*S: REM NOW IN IS OR
1080 LET YD=INT(FT/3)*S: LET FT=FT*S-3*YD:RETURN
2000 REM SUBROUTINE TO PRINT YD,FT AND IN " ! PRINT YD "
    YD" FT "FT" IN "IN" :RETURN

```

ЭТА ПРОГРАММА МОДЕЛИРУЕТ ВЫБРАСЫВАНИЕ МОНЕТЫ ДЛЯ ИГРЫ В
"ТАЙКУ":

```

5 RANDOMIZE
10 FOR M=1 TO 6: REM FOR 6 THROWS
20 LET C=0:REM INITIALIZE COINN TOTAL TO 0
30 FOR N=1 TO 3: REM FOR 3 COINS
40 LET C=C+2+INT(2*RND)
50 NEXT N
60 PRINT " "
70 FOR N=1 TO 2: REM 1BT FOR THE THROWN HEXAGRAM,
    2 ND FOR THE CHANGES
80 PRINT " ----"
90 IF C=7 THEN PRINT "--"
100 IF C=8 THEN PRINT " "
110 IF C=6 THEN PRINT "X" : LET C=7
120 IF C=9 THEN PRINT "0" : LET C=8
130 PRINT "-----"
140 NEXT N
150 PRINT
160 INPUT A*
170 NEXT M:NEW

```

ДЛЯ ЗАПУСКА ПРОГРАММЫ ВВЕДИТЕ ЕЕ В КОМПЬЮТЕР, ЗАПУСТИТЕ ИСПО-
ЛНЕНИЕ, А ЗАТЕМ НАЖМИТЕ КЛАВИШУ ENTER ПЯТЬ РАЗ ДЛЯ ПОЛУЧЕНИЯ ДВУХ
ХЕКСАГРАММ, ПРОСМОТРИТЕ "КИТАЙСКУЮ КНИГУ ИЗМЕНЕНИЙ", ОНА БУДЕТ
ОПИСЫВАТЬ СИТУАЦИЮ И ПОСЛЕДОВАТЕЛЬНОСТЬ СООТВЕТСТВУЮЩУЮ
ЭТОМУ ДЕЙСТВИЮ, А ВЫ ДОЛЖНЫ ОЦЕНИТЬ ГЛУБИНУ ПАРАЛЛЕЛЕЙ МЕЖДУ НЕИ
И ВАШЕЙ СОБСТВЕННОЙ ЖИЗНЬЮ. НАЖМИТЕ КЛАВИШУ ENTER МЕСТОИ РАЗ.
ПРОГРАММА БУДЕТ ОБНУЛЯТЬСЯ— ЭТО ИЗБАВИТ ВАС ОТ ЛЕГКОМЫСЛЕННОГО
ИСПОЛЬЗОВАНИЯ РЕЗУЛЬТАТОВ.

МНОГИЕ ПОЛЬЗОВАТЕЛИ НАИДУТ ТЕСТЫ ВСЕГДА БОЛЕЕ ВЕРОЯТНЫМИ,
НЕЖЕЛИ ОНИ САМИ МОГУТ ЭТО ПРЕДПОЛАГАТЬ.

СЛЕДУЮЩАЯ ПРОГРАММА-ИГРА "ЯШЕРЫ". ВЫ ЗАДУМЫВАЕТЕ НАЗВАНИЕ
НЕКОТОРОГО ЖИВОТНОГО, А КОМПЬЮТЕР ПЫТАЕТСЯ ЕГО ОТГАДАТЬ, ЗАДАВАЯ
ВАМ ВОПРОСЫ, НА КОТОРЫЕ ВЫ ДОЛЖНЫ ОТВЕЧАТЬ "ДА" ИЛИ "НЕТ".
ЕСЛИ КОМПЬЮТЕР НЕ БЫЛ РАНЕЕ ЗНАКОМ С ТАКИМ ЖИВОТНЫМ, ТО ОН ПОПРО-
СИТ ВАС ЗАДАТЬ ЕМУ НАВОДЯЩИЕ ВОПРОСЫ, КОТОРЫЕ ПОМОГУТ ЕМУ НАЙТИ
ПРАВИЛЬНЫЙ ОТВЕТ ИЛИ ОН ПОПРОСИТ ВАС ПРЕДЛОЖИТЬ ЕМУ ЗАДАТЬ НАЗ-
ВАНИЕ НОВОГО ЖИВОТНОГО:

```

5 REM PANGOLINS
10 LET NG=100: REM NUMBER OF QUESTIONS AND ANIMALS
15 DIM G*(NQ,60):DIM A(NQ,2): DIM R*(1)
20 LET QF=8
30 FOR N=1 TO QF/2-1
40 READ Q*(N): READ A(N,1): READ A(N,2)
50 NEXT N
60 FOR N=N TO QF-1
70 READ Q*(N): NEXT N

```

```

100 REM 110 PRINT "THINK OF AN ANIMAL.", "PRESS ANY KEY TO
    CONTINUE."
120 PAUSE 0
130 LET C=1. REM START WITH 1ST QUESTION
140 IF A(C,1)=0 THEN GO TO 300
150 LET P=Q(C): GO SUB 910
160 PRINT "?": GO SUB 1000
170 LET IN=1: IF R="Y" THEN GO TO 210
180 IF R="Y" THEN GO TO 210
190 LET IN=2: IF R="N" THEN GO TO 210
200 IF R<>"N" THEN GO TO 150
210 LET C=A(C,IN): GO TO 140
300 REM ANIMAL
310 PRINT "ARE YOU THINKING Q?"
320 LET P=Q(C): GO SUB 900: PRINT "?"
330 GO SUB 1000
340 IF R="Y" THEN GO TO 400
350 IF R="Y" THEN GO TO 400
360 IF R="H" THEN GO TO 500
370 IF R="N" THEN GO TO 500
380 PRINT "ANSWER ME PROPERLY WHEN I", "TALKING TO YOU";
    GO TO 300
400 REM GUESSED IT
410 PRINT "I THOUGHT AS MUCH."; GO TO 800
500 REM NEW ANIMAL
510 IF Q>NQ-1 THEN PRINT "I'M SURE YOUR ANIMAL IS VERY",
    "INTERESTING, BUT I DON'T HAVE", "ROOM FOR IT JUST NOW."
    GO TO 800
520 LET Q(Q)=Q(C): REM MOVE OLD ANIMAL
530 PRINT "WHAT IS IT, THEN?": INPUT Q(Q+1)
540 PRINT "TELL ME A QUESTION WHICH DIST-", "INQUIRES
    BETWEEN"
550 LET P=Q(Q): GO SUB 900: PRINT "AND"
560 LET P=Q(Q+1): GO SUB 900: PRINT " "
570 INPUT S: LET B=LEN S
580 IF S(B)="?" THEN LET B=B-1
590 LET Q(C)=S(TO B): REM INSERT QUESTION
600 PRINT "WHAT IS THE ANSWER FOR"
610 LET P=Q(Q+1): GO SUB 900: PRINT "?"
620 GO SUB 1000
630 LET IN=1: LET IO=2: REM ANSWERS FOR NEW AND OLD ANIMALS
640 IF R="Y" THEN GO TO 700
650 IF R="Y" THEN GO TO 700
660 LET IN=2: LET IO=1
670 IF R="H" THEN GO TO 700
680 IF R="N" THEN GO TO 700
690 PRINT "THAT'S NO GOOD.": GO TO 600
700 REM UPDATE ANSWERS
710 LET A(C,IN)=Q+1: LET A(C,IO)=Q
720 LET Q=Q+2: REM NEXT FREE ANIMAL SPACE
730 PRINT "THAT FOOLED ME."
800 REM AGAIN?
810 PRINT "DO YOU WANT ANOTHER GO?": GO SUB 1000
820 IF R="Y" THEN GO TO 100
830 IF R="Y" THEN GO TO 100
840 STOP
900 REM PRINT WITHOUT TRAILING SPACES
905 PRINT " "
910 FOR N=50 TO 1 STEP-1

```

```

920 IF P*(N)<>" " THEN GO TO 940
930 NEXT N
940 PRINT P*(TO N) : RETURN
1000 REM GET REPLY
1010 INPUT R*: IF R*=" " THEN RETURN
1020 LET R*=R*(1): RETURN
2000 REM INITIAL ANIMALS
2010 DATA " DOES IT LINE IN THE SEA",4,2
2020 DATA " IS IT SCALY",3,5
2030 DATA " DOES IT EAT ANTS",6,7
2040 DATA " A WHALE", " A BLANCMANGE", " A PANGOLIN", " AN ANT"

```

СЛЕДУЮЩАЯ ПРОГРАММА РИСУЕТ НА ЭКРАНЕ "UNION FLAG":

```

5 REM UNION FLAG
10 LET R=2: LET W=7: LET B=1
20 BORDER 0: PAPER 8: INK W: CLS
30 REM BLACK IN BOTTOM OF SCREEN
40 INVERSE 1
50 FOR N=40 TO 0 STEP -5
60 PLOT PAPER 0 7,N: DRAW PAPER 0 24,0
70 NEXT N: INVERSE 0
100 REM DRAW IN WHITE PARTS
105 REM ST. GEORGE
110 FOR N=0 TO 7
120 PLOT 104+N,175: DRAW 0,-35
130 PLOT 151-N,175: DRAW 0,-35
140 PLOT 151,-N,48: DRAW 0,35
150 PLOT 104+N,48: DRAW 0,35
160 NEXT N
200 FOR N=0 TO 11
210 PLOT 0,139-N: DRAW 111,0
220 PLOT 255,139-N: DRAW -111,0
230 PLOT 255,84+N: DRAW -111,0
240 PLOT 0,84+N: DRAW 111,0
250 NEXT N
300 REM ST. ANDREW
310 FOR N=0 TO 35
320 PLOT 1+2*N,175-N: DRAW 32,0
330 PLOT 224-2*N,175-N: DRAW 16,0
340 PLOT 254-2*N,48+N: DRAW -32,0
350 PLOT 17+2*N,48+N: DRAW 16,0
360 NEXT N
370 FOR N=0 TO 19
380 PLOT 185+2*N,140+N: DRAW 32,0
390 PLOT 200+2*N,83-N: DRAW 16,0
400 PLOT 39-2*N,83-N: DRAW 32,0
410 PLOT 54-2*N,140+N: DRAW -16,0
420 NEXT N
430 FOR N=0 TO 15
440 PLOT 255,160+N: DRAW 2*N-30,0
450 PLOT 0,63-N: DRAW 3102*N,0
460 NEXT N
470 FOR N=0 TO 7
480 PLOT 0,160+N: DRAW 14-2*N,0
485 PLOT 255,63-N: DRAW 2*N-15,0
490 NEXT N
500 REM RED STRIPES
510 INVERSE 1
520 REM ST. GEORGE

```

```

530 FOR N=96 TO 120 STEP 8
540 PLOT PAPER R 7,N: DRAW PAPER N 241,0
550 NEXT N
560 FOR N=112 TO 136 STEP 8
570 PLOT PAPER R N,100: DRAW PAPER R 0,-113
580 NEXT N
600 REM ST, PATRICK
610 PLOT PAPER R 170,140: DRAW PAPER R 70,35
620 PLOT PAPER R 179,140: DRAW PAPER R 70,35
630 PLOT PAPER R 199,83: DRAW PAPER R 56,-28
640 PLOT PAPER R 184,83: DRAW PAPER R 70,-35
650 PLOT PAPER R 86,83: DRAW PAPER R -70,-35
660 PLOT PAPER R 72,83: DRAW PAPER R -70,-35
670 PLOT PAPER R 56,140: DRAW PAPER R -56,28
680 PLOT PAPER R 71,140: DRAW PAPER R -70,35
690 INVERSE 0: PAPER 0: INK 7

```

ЕСЛИ ВЫ НЕ АНГЛИЧАНИН, ТО МОЖЕТЕ ИЗООБРАЗИТЬ СВОЙ ФЛАГ.
 СЛЕДУЮЩАЯ ПРОГРАММА - ЭТО ИГРА В СЛОВА, ПЕРВЫЙ ИГРОК ВВОДИТ
 СЛОВО, А ВТОРОЙ ЕГО ОТГАДЫВАЕТ.

```

5 REM HANGMAN
10 REM SET UP SCREEN
20 INK 0: PAPER 7: CLS
30 LET X=240: GO SUB 1000: REM DRAW MAN
40 PLOT 235,128: DRAW 4,0: REM MOUTH
100 REM SET UP WORD
110 INPUT W$: REM WORD TO GUESS
120 LET B=LENW$: LET V$=""
130 FOR N=2 TO B: LET V$=V$+" "
140 NEXT N: REM V$=WORD GUESSED SO FAR
150 LET C=0: LET D=0: REM GUESS & MISTAKE COUNTS
160 FOR N=0 TO B-1
170 PRINT AT 20,N " "
180 NEXT N: REM WRITE- 'S INSTEAD OF LETTERS'
200 INPUT "GUESS A LETTER:" G$
210 IF G$="" THEN GO TO 200
220 LET G$=G$(1): REM 1ST LETTER ONLY
230 PRINT AT 0,C G$
240 LET C=C+1: LET U$=V$
250 FOR N=1 TO B: REM UPDATE GUESSED WORD
260 IF W$(N)=G$ THEN LET V$(N)=G$
270 NEXT N
280 PRINT AT 19,0 V$
290 IF V$=W$ THEN GO TO 500: REM WORD GUESSED
300 IF V$<>U$ THEN GO TO 200: REM GUESSED WAS RIGHT
400 REM DRAW NEXT PART OF GALLOWS
410 IF D=8 THEN GO TO 600: REM HANGED
420 LET D=D+1
430 READ X0,Y0,X,Y
440 PLOT X0,Y0: DRAW X,Y
450 GO TO 200
500 REM FREE MAN
510 OVER1: REM RUB OUT MAN
520 LET X=240: GO SUB 1000
530 PLOT 235,128: DRAW 4,0: REM MOUTH
540 OVER 0: REM REDRAW MAN
550 LET X=146: GO SUB 1000
560 PLOT 143,129: DRAW 6,0,P1/2: REM SMILE
570 GO TO 800

```



```

600 REM HANG MAN
610 OVER 1: REM RUB OUT FLOOR
620 PLOT 255,65: DRAW -48,0
630 DRAW 0, -48: REM OPEN TRAPDOOR
640 PLOT 238,128: DRAW 4,0: REM RUB OUT MOUTH
650 REM MOVE LIMBS
655 REM ARMS
660 PLOT 255,117: DRAW -15,-15: DRAW -15,15
670 OVER 0
680 PLOT 238,81: DRAW 4,21: DRAW 4,21: DRAW 4,-21
690 OVER 1: REM LEGS
700 PLOT 255,56: DRAW -15,15: DRAW -15,-15
710 OVER 0
720 PLOT 236,60: DRAW 4,21: DRAW 4,-21
730 PLOT 237,127: DRAW 6,0,PI/2: REM FROWN
740 PRINT AT 19,0 W$
800 INPUT "AGAIN?" A$
810 IF A$="" THEN GO TO 850
820 LET A$=A$(1)
830 IF A$="N" THEN STOP
840 IF A$(1)="N" THEN STOP
850 RESTORE: GOTO 5
1000 REM DRAW MAN AT COLUMN X
1010 REM HEAD
1020 CIRCLE X,132,8
1030 PLOT X+4,134: PLOT X-4,134: PLOT X,131
1040 REM BODY
1050 PLOT X,123: DRAW 0,-20
1055 PLOT X,101: DRAW 0,-19
1060 REM LEGS
1070 PLOT X-15,66: DRAW 15,15: DRAW 15,-15
1080 REM ARMS
1090 PLOT X-15,117: DRAW 15,-15: DRAW 15,15
1100 RETURN
2000 DATA 120,55,135,0,184,65,0,91
2010 DATA 168,65,16,16,164,81,16,-16
2020 DATA 184,156,68,0,184,140,16,16
2030 DATA 204,156,-20,-22,240,156,0,-16
2040 DATA 184,156,68,0,184,140,16,16
2050 DATA 204,156,-20,-20,240,156,0,-16

```

ПРИЛОЖЕНИЕ Е

ШЕСТНАДЦАТИРИЧНАЯ И ДВОИЧНАЯ СИСТЕМЫ СЧИСЛЕНИЯ

=====

В КОМПЬЮТЕРЕ ZX SPECTRUM ИСПОЛЬЗУЕТСЯ ШЕСТНАДЦАТИРИЧНАЯ СИСТЕМА СЧИСЛЕНИЯ. ПРИ ЭТОМ КАЖДАЯ ШЕСТНАДЦАТИРИЧНАЯ ЦИФРА ЗАПИСЫВАЕТСЯ ЧЕТЫРЬМЯ ДВОИЧНЫМИ (ТЕТРАДА), ТАКИМ ОБРАЗОМ, В ОДНОМ БАЙТЕ МОЖЕТ БЫТЬ ЗАПИСАНО ДВА ШЕСТНАДЦАТИРИЧНЫХ ЧИСЛА.

10-ТИЧНОЕ 16-РИЧНОЕ 2-ИЧНОЕ(БАЙТ)

0	0	0000 0000
1	1	0000 0001
2	2	0000 0010
3	3	0000 0011
4	4	0000 0100
5	5	0000 0101
6	6	0000 0110
7	7	0000 0111
8	8	0000 1000
9	9	0000 1001
10	A	0000 1010
11	B	0000 1011
12	C	0000 1100
13	D	0000 1101
14	E	0000 1110
15	F	0000 1111
16	10	0001 0000
17	11	0001 0001
18	12	0001 0010
19	13	0001 0011
20	14	0001 0100
21	15	0001 0101
22	16	0001 0110
23	17	0001 0111
24	18	0001 1000
25	19	0001 1001
26	1A	0001 1010
27	1B	0001 1011
28	1C	0001 1100
29	1D	0001 1101
30	1E	0001 1110
31	1F	0001 1111
32	20	0010 0000

ДВА БАЙТА ОБРАЗУЮТ МАШИННОЕ СЛОВО. ДЛЯ ЗАПИСИ ДВОИЧНЫХ КОДОВ СЛУЖИТ ФУНКЦИЯ BIN, НАПРИМЕР 'BIN 0' ЗАПИШЕТ В ПАМЯТЬ ДВОИЧНЫЙ 0 'BIN 10' ЗАПИСЫВАЕТ ЧИСЛО ДВА И Т.Д. ДЛЯ ЗАПИСИ '-3' НЕОБХОДИМО УКАЗАТЬ 'BIN 11', НО НЕ 'BIN-11'.

ЧИСЛО НЕ МОЖЕТ ПРЕВЫШАТЬ 65535 Т.Е. ЗАНИМАТЬ БОЛЕЕ ШЕСТНАДЦАТИ ДВОИЧНЫХ РАЗРЯДОВ(БИТОВ).

УКАЗАТЕЛЬ КОМАНД БЕЙСИКА

-----A-----		
ABS.....	[E]-G.....	13
ACS.....	[E]-SH-W.....	18
AND.....	[K],[L],[C]-SIMBOL-SHIFT-Y.....	20
ASN.....	[E]-SN-O.....	18
AT.....	[K],[L],[C]-SYMBOL-SHIFT-I.....	25,70
ATN.....	[E]-SH-E.....	16,18
ATTR.....	[E]-SH-L.....	31,48
-----B-----		
BEEP.....	[E]-SH-Z.....	2,36,37
BIN.....	[E]-B.....	22,34
BORDER.....	[K]-B.....	2
BREAK.....	CAPS-SHIFT-SPACE.....	3,7,42

BRIGHT.....	[E]-SH-B.....	28
CAPS LOCK.....	[K],[L]-CAPS-SHIFT-2.....	3
CAPS SHIFT.....		2
CAT.....	[E]-SYMBOL-SHIFT-9>.....	43
CHR*	[E]-U.....	21,27,31
CIRCLE.....	[N]-SN-H.....	33,69
CLPEAR.....	[K]-X.....	56
CLOSE#.....	[E]-SYMBOL-SHIFT-5.....	43
CLS.....	[K]-V.....	7,26
CODE.....	[E]-I.....	21,42
CONTINUE.....	[K]-C.....	7
COPY.....	[K]-Z>.....	42
COS.....	[E]-W.....	16
DATA.....	[E]-D.....	10,19,39
DEF FN.....	[E]-SYMBOL-SHIFT-1.....	15
DELETE.....	[C],[G]-Ø/[K],[L]-CAPS-SHIFT-Ø.....	3,5,22,6
DIM.....	[K]-D.....	19
DRAW.....	[K]-W.....	53,69
EDIT.....	[K],[L],[C]-CAPS-SHIFT-1.....	3,4
ENTER.....		4,68
ERASE.....	[E]-SYMBOL-SHIFT-7.....	42
EXP.....	[E]-X.....	16
FLASN.....	[E]-SH-V.....	28,33
FN.....	[E]-SYMBOL-SHIFT-2.....	15
FOR.....	[K]-F.....	8,9,70
FOR NEXT ЦИКЛ.....		8,9,70
FORMAT.....	[E]-SYMBOL-SHIFT-Ø.....	42
GO SUB.....	[K]-H.....	9
GO TO.....	[K]-G.....	6,7,8,9
GRAPHICS.....	[K],[L],[G]-CAPS-SHIFT-Ø.....	3,21,67
IF.....	[K]-U.....	6,8
IN.....	[E]-SH-I.....	43
INK.....	[E]-SH-X.....	26,33,64
INKEY*	[E]-N.....	38
INPUT.....	[K]-I.....	6,7,8,69
INVERSE.....	[K]-SH-N.....	28,33,69
INT.....	[K]-R.....	14,18
LEN.....	[E]-K.....	14
LET.....	[K]-L.....	2,4,8,9
LINE.....	[E]-SYMBOL-SHIFT-3.....	27,39,66
LIST.....	[K]-K.....	5
LLIST.....	[E]-V.....	43
LN.....	[E]-Z.....	16
LOAD.....	[K]-J.....	35,56
LPRINT.....	[E]-C.....	42
MERGE.....	[E]-S'-T.....	41
MOVE.....	[E]-SYMBOL-SHIFT-6.....	37
NEW.....	[K]-A.....	5,7
NEXT.....	[K]-N.....	8,10,70
NOT.....	[K],[L],[C]-SYMBOL-SHIFT-5.....	20

O		
OPEN#	[F]-SYMBOL-SHIFT-4	42
OR	[K],[L],[C]-SYMBOL-SHIFT-U	20
OUT	[E]-O	43
OVER	[E]-N	28
P		
PAPER	[E]-C	2,28,33
PAUSE	[K]-M	35
PEEK	[E]-O	22,35
PI	[E]-M	16
PLOT	[Y]-O	33
POKE	[K]-O	23
PRINT	[K]-P	2,4,7,9
R		
RANDOMISE	[K]-T	18
READ	[E]-A	10
REM	[K]-E	5,7,8
RESTORE	[E]-S	10
RETURN	[K]-Y	9
RND	[E]-T	18
RUN	[K]-R	4
S		
SAVE	[K]-S	39,55
SCREEN*	[E]-SH-K	40,48
SGN	[E]-F	14
SIN	[E]-O	18
SGR	[E]-H	14
STEP	[K],[L],[C]-SYMBOL-SHIFT-D	8
STOP	[K],[L],[C]-SYMBOL-SHIFT-A	3,5
STR*	[E]-Y	14
SYMBOL SHIFT		2
T		
TAB	[E]-P	25
TAN	[E]-E	18
THEN	[K],[L],[C]-SYMBOL-SHIFT-G	2,7,20
TO	[K],[L],[C]-SYMBOL-SHIFT-F	8,13
U		
USR	[E]-L	21
V		
VAL	[E]-J	14
VAL*	[E]-SH-J	14
VERIFAY	[E]-SH-R	39

* МННСК 88 *

TASWORD II

ТЕКСТОВЫЙ РЕДАКТОР ZX-SPECTRUM

1983 TASMAN SOFTWARE

* МИНСК 88 *

TASWORD II

ТЕКСТОВЫЙ РЕДАКТОР ZX-SPECTRUM

1983 TASMAN SOFTWARE

ТЕКСТОВЫЙ РЕДАКТОР TASWORD II

TASWORD II ОБЕСПЕЧИВАЕТ ДВА РЕЖИМА ПЕЧАТИ - 64 СИМВОЛА В СТРОКЕ И 32 СИМВОЛА В СТРОКЕ. РЕЖИМ ПЕЧАТИ ЗАГЛАВНЫХ БУКВ УСТАНОВЛИВАЕТСЯ ПУТМ НАЖАТИЯ CS+БУКВА. СИМВОЛЫ ! @ # \$ % & * () + ? " ' < > , . :] [[] - ВЫВОДЯТСЯ ЧЕРЕЗ SS ЛИБО CS.

1. УПРАВЛЯЮЩИЕ КЛАВИШИ.

ПРИ НАЖАТИИ ОДНОГО ИЗ SHIFT И СООТВЕТСТВУЮЩЕЙ КЛАВИШИ ВЫПОЛНЯЮТСЯ СЛЕДУЮЩИЕ ДЕЙСТВИЯ:

EDIT--ВЫВОД HELP PAGE
CAPS LOCK--ПЕЧАТЬ ЗАГЛАВНЫХ БУКВ
TRUE VIDEO--КУРСОР НА СЛОВО ВЛЕВО
INV. VIDEO--КУРСОР НА СЛОВО ВПРАВО
И Т. Д.

ЕСЛИ ВЫ РАБОТАЕТЕ С HELP PAGE, ТО ПРИ НАЖАТИИ SS+CS ВЫ ПОЛУЧИТЕ ВОЗМОЖНОСТЬ ИЗУЧИТЬ ФУНКЦИИ УПРАВЛЯЮЩИХ КЛАВИШ РАСШИРЕННОГО РЕЖИМА.

2. РАСШИРЕННЫЙ РЕЖИМ.

РАСШИРЕННЫЙ РЕЖИМ УСТАНОВЛИВАЕТСЯ НАЖАТИЕМ SS+CS. ПРИ ЭТОМ НИЖНЯЯ ЧАСТЬ ЭКРАНА НАЧН Т МИГАТЬ.

УПРАВЛЯЮЩИЕ КЛАВИШИ В РАСШИРЕННОМ РЕЖИМЕ.

ПРОКРУТКА (НА 22 СТРОКИ)

F--БЫСТРАЯ ВНИЗ
G--БЫСТРАЯ ВВЕРХ

УПР. ПРИНТЕРОМ

P--ПЕЧАТЬ ФАЙЛА, Вых Q
L--МАРКЕР ВКЛЮЧЕНИЯ РЕЖИМА
ПЕЧАТИ УДВОЕННЫХ БУКВ
K--МАРКЕР ОТМЕНЫ ПЕЧАТИ
УДВОЕННЫХ БУКВ

РАЗНОЕ

C--ИЗМЕНЕНИЕ ОКНА
X--ОЧИСТКА ТЕКСТОВОГО ФАЙЛА
R--ЗАМЕНА ПОИСК ТЕКСТА
I--ВСТАВКА Вкл выкл

ФОРМАТИРОВАНИЕ

E--ВЫРАВНИВАНИЕ ВПРАВО (Вкл ОТКЛ)
W--ПЕРЕНОС СЛОВА (Вкл ОТКЛ)
J--ВЫРАВНИВАНИЕ СТРОКИ
K--АНТИВЫРАВНИВАНИЕ

ГРАНИЦЫ

A--УСТАНОВКА ЛЕВОЙ ГРАНИЦЫ
ПО КУРСОРУ
S--УСТАНОВКА НОРМАЛЬНЫХ ГРАНИЦ
D--УСТАНОВКА ПРАВОЙ ГРАНИЦЫ
ПО КУРСОРУ

БЛОКОВЫЕ КОМАНДЫ

B--МЕТКА НАЧАЛА БЛОКА, [
M--ПЕРЕСЫЛКА БЛОКА В МЕСТО
КУРСОРА,]

(1)+(CS)--HELP PAGE
(5-8)+(CS)--УПРАВЛЕНИЕ КУРСОРОМ

ПЕРЕНОС СЛОВА;

ЕСЛИ СТРОКА НЕ УМЕЩАЕТСЯ В УСТАНОВЛЕННЫЕ ГРАНИЦЫ, TASWORD II ПЕРЕНОСИТ ПОСЛЕДНЕЕ СЛОВО НА НОВУЮ СТРОКУ.

ВЫРАВНИВАНИЕ ТЕКСТА.

ПРИ ВЫРАВНИВАНИИ ТЕКСТА СТРОКА ДОПОЛНЯЕТСЯ ПРОБЕЛАМИ ТАК ЧТОБЫ ПОСЛЕДНЯЯ БУКВА ПОСЛЕДНЕГО СЛОВА СТРОКИ НАХОДИЛАСЬ НА ПРАВОЙ ГРАНИЦЕ,

ВЫСОКИЙ КУРСОР.

ПОСЛЕ ТОГО, КАК ВЫ НАПЕЧАТАЛИ ПОСЛЕДНИЙ СИМВОЛ В СТРОКЕ, TASWORD II ПЕРЕНОСИТ КУРСОР В НАЧАЛО СЛЕДУЮЩЕЙ СТРОКИ. ПРИ ЭТОМ КУРСОР УВЕЛИЧИВАЕТСЯ. ПРИ ТАКОМ ВИДЕ КУРСОРА TASWORD II ВОСПРИНИМАЕТ ВВЕДЕННЫЙ СИМВОЛ, КАК ЧАСТЬ ПОСЛЕДНЕГО СЛОВА В ПРЕДИДУЩЕЙ СТРОКЕ И, СЛЕДОВАТЕЛЬНО, ПЕРЕНОСИТ ЭТО СЛОВО (СМ. "ВЫРАВНИВАНИЕ") Т. О. ЕСЛИ СЛОВО В ПОСЛЕДНЕЙ СТРОКЕ ЗАКОНЧЕНО, ПРИ ВЫСОКОМ КУРСОРЕ ВЫ ДОЛЖНЫ НАПЕЧАТАТЬ ПРОБЕЛ,

!!!!!!!!!!!!

ПОЛЕЗНЫЕ СОВЕТЫ

!!!!!!!!!!!!

ВСЕГДА ПЕЧАТАЙТЕ ПРОБЕЛ В КОНЦЕ ПРЕДЛОЖЕНИЯ ПОСЛЕ ЛЮБОГО ЗНАКА ПРЕПИНАНИЯ.

ВСЕГДА ПЕЧАТАЙТЕ НОВЫЙ АБЗАЦ С "КРАСНОЙ" СТРОКИ ИЛИ ВСТАВЛЯЙТЕ МЕЖДУ АБЗАЦАМИ ПУСТЫЕ СТРОКИ (МОЖНО ДЕЛАТЬ ТО И ДРУГОЕ).

КОПИРОВАНИЕ TASWORD II

КОПИРОВАНИЕ И ПЕРЕДАЧА TASWORD II ТРЕТЬИМ ЛИЦАМ ЗАПРЕЩЕНО. ДЛЯ КОПИРОВАНИЯ TASWORD II НЕОБХОДИМО НАЖАТЬ УПРАВЛЯЮЩИЙ СИМВОЛ STOP ВО ВРЕМЯ РАБОТЫ TASWORD II, ПРИ ЭТОМ ПОЯВИТСЯ МЕНЮ, ДАЛЕЕ НЕОБХОДИМО НАЖАТЬ "Т" ДЛЯ ЗАПИСИ TASWORD II НА МАГНИТОФОН.

3. ЗАПИСЬ/ЧТЕНИЕ ТЕКСТОВЫХ ФАЙЛОВ.

ДЛЯ ЗАПИСИ ФАЙЛА НЕОБХОДИМО НАЖАТЬ STOP (SS+A) В НОРМАЛЬНОМ РЕЖИМЕ РАБОТЫ, ДАЛЕЕ ВЫБРАТЬ В МЕНЮ "SAVE TEXT FILE" НАЖАТИЕМ "S". НА ЗАПРОС ИМЕНИ ФАЙЛА ВЫ ДОЛЖНЫ НАВРАТЬ ИМЯ ФАЙЛА (НЕ БОЛЕЕ 10 СИМВОЛОВ) И НАЖАТЬ "ENTER". ПОСЛЕ ЗАПИСИ ВАШЕГО ФАЙЛА TASWORD II ПРЕДОСТАВЛЯЕТ ВОЗМОЖНОСТЬ ЕГО ВЕРИФИКАЦИИ. ПРИ ПОЯВЛЕНИИ СООБЩЕНИЯ "TAPE LOADING ERROR" НЕОБХОДИМО НАЖАТЬ "R" ДЛЯ "RUN" И "ENTER". ПРИ ЗАГРУЗКЕ НОВОГО ФАЙЛА С ЛЕНТЫ, ФАЙЛ, НАХОДЯЩИЙСЯ В ДАННЫЙ МОМЕНТ В TASWORD II СТИРАЕТСЯ. ДЛЯ ЗАГРУЗКИ ФАЙЛА НЕОБХОДИМО ВОЙТИ В МЕНЮ (ПО "STOP") ВЫБРАТЬ "LOAD TEXT FILE" НАЖАТИЕМ "J", ДАЛЕЕ НАВРАТЬ ИМЯ ФАЙЛА И "ENTER".

В TASWORD II СУЩЕСТВУЕТ ВОЗМОЖНОСТЬ ДОГРУЖАТЬ ТЕКСТ С ЛЕНТЫ КОТОРЫЙ РАСПОЛАГАЕТСЯ ПОСЛЕ УЖЕ СУЩЕСТВУЮЩЕГО ТЕКСТА В TASWORD II. ДЛЯ ЭТОГО В МЕНЮ НЕОБХОДИМО ВЫБРАТЬ "M". ПРИ ЭТОМ НЕОБХОДИМО ПОМНИТЬ, ЧТО МАКСИМАЛЬНАЯ ДЛИНА ТЕКСТА В TASWORD II - 320 СТРОК. ЕСЛИ ЭТО УСЛОВИЕ НЕ СОБЛЮДЕНО, ВЫ ОКАЖЕТЕСЬ В BASIC. ДЛЯ ПОВТОРНОГО ЗАПУСКА TASWORD II НАБЕРИТЕ "TASWORD II" + "RUN" + "ENTER".

4. TASWORD II TUTOR

TASWORD II TUTOR ЯВЛЯЕТСЯ ТЕКСТОВЫМ ФАЙЛОМ, ПРЕДНАЗНАЧЕННЫМ ДЛЯ ОБУЧЕНИЯ РАБОТЕ С TASWORD II.

5. УПРАВЛЯЮЩИЕ КЛАВИШИ.

НИЖЕ ОПИСАНЫ УПРАВЛЯЮЩИЕ КЛАВИШИ, РАБОТАЮЩИЕ В НОРМАЛЬНОМ РЕЖИМЕ Т.Е. ТОГДА, КОГДА КОМАНДНЫЕ СТРОКИ НЕ МИГАЮТ.

EDIT	(CS+1)	--HELP PAGE. В ЭТОМ РЕЖИМЕ МОЖНО ВЫВЕСТИ ИНФОРМАЦИЮ О РЕЖИМЕ РАСШИРЕНИЯ НАЖАТИЕМ "SS"+"CS"
ENTER		--ВЫХОД ИЗ HELP PAGE.
CAPS LOCK	(CS+2)	--РЕЖИМ БОЛЬШИХ БУКВ ВКЛ/ВЫКЛ.
TRUE VIDEO	(CS+3)	--КУРСОР В КОНЕЦ ПРЕДИДУЩЕГО СЛОВА.
INV. VIDEO	(CS+4)	--КУРСОР В НАЧАЛО СЛЕДУЮЩЕГО СЛОВА.
DELETE	(CS+0)	--УДАЛЕНИЕ СИМВОЛА НАД КУРСОРОМ.
GRAPHICS	(CS+9)	--РЕЖИМ ПЕЧАТИ ГРАФИЧЕСКИХ СИМВОЛОВ.
<=	(SS+Q)	--СДВИГ ТЕКСТА ПОД КУРСОРОМ ВЛЕВО.
>=	(SS+E)	--СДВИГ ТЕКСТА ПОД КУРСОРОМ ВПРАВО.
<>	(SS+W)	--ЦЕНТРОВКА СТРОКИ.
AND	(SS+Y)	--ВСТАВКА СТРОКИ, СИМВОЛА. ДЛЯ ВСТАВКИ ПУСТОЙ СТРОКИ НЕОБХОДИМО УСТАНОВИТЬ КУРСОР В НАЧАЛО СТРОКИ, ПЕРЕД КОТОРОЙ ВСТАВЛЯЕТСЯ СТРОКА, И НАЖАТЬ "AND".
OR	(SS+U)	--КУРСОР В КОНЕЦ ТЕКСТА.
AT	(SS+I)	--КУРСОР В НАЧАЛО ТЕКСТА.
STOP	(SS+A)	--ВЫХОД В МЕНЮ. ПРИ ВЫХОДЕ В BASIC, В РЕДАКТОР МОЖНО ВЕРНУТЬСЯ НАЖАВ " RUN " И " ENTER ".
NOT	(SS+S)	--УДАЛЕНИЕ СТРОКИ.
STEP	(SS+D)	--ВЫРАВНИВАНИЕ ТЕКСТА ОТ СТРОКИ, СОДЕРЖАЩЕЙ КУРСОР, ДО КОНЦА АБЗАЦА. КОНЕЦ АБЗАЦА ОПРЕДЕЛЯЕТСЯ ПО ПРОБЕЛАМ ИЛИ ПУСТОЙ СТРОКЕ. ТЕКСТ ВЫРАВНИВАЕТСЯ ТОЛЬКО ПРИ ВКЛЮЧЕННОМ RIGHT JUSTIFY И НЕ ВЫРАВНИВАЕТСЯ, КОГДА R. JUSTIFY ВЫКЛЮЧЕНО.
TO	(SS+F)	--СДВИГ ТЕКСТА ВНИЗ НА ОДНУ СТРОКУ.
THEN	(SS+G)	--СДВИГ ТЕКСТА ВВЕРХ НА ОДНУ СТРОКУ.
ENTER		--ПЕРЕХОД НА НОВУЮ СТРОКУ ЛИБО ВСТАВКА ПУСТОЙ СТРОКИ В РЕЖИМЕ "ВСТАВКИ".

6. РАСШИРЕННЫЙ РЕЖИМ.

ЗДЕСЬ ДАНО БОЛЕЕ ПОДРОБНОЕ ОПИСАНИЕ ОТДЕЛЬНЫХ ФУНКЦИЙ ОПИСАННЫХ В ПУНКТЕ 2.

R - ЗАМЕНА ПОИСК ТЕКСТА ОТ КУРСОРА ДО КОНЦА ТЕКСТА (ДЛЯ ПОИСКА ЗАМЕНЫ ОТ НАЧАЛА ТЕКСТА СЛЕДУЕТ ИСПОЛЬЗОВАТЬ AT ПЕРЕД ВХОДОМ В РАСШИРЕННЫЙ РЕЖИМ). В РЕЖИМЕ "R" TASWORD II ЗАПРАШИВАЕТ СЛОВО ДЛЯ ПОИСКА ИЛИ ЗАМЕНЫ (ПРОБЕЛЫ ПРИ НАБОРЕ СЛОВА НЕ ДОПУСКАЮТСЯ). ПОСЛЕ ЭТОГО НАЖИМАЮТ "ENTER" И TASWORD II НАХОДИТ В ТЕКСТЕ НАЙДЕННОЕ ВАМИ СЛОВО. МОЖНО ПРОДОЛЖИТЬ ПОИСК НАЖАВ ЕЩЕ РАЗ "ENTER". ДЛЯ ЗАМЕНЫ НАЙДЕННОГО СЛОВА НЕОБХОДИМО НАБРАТЬ ТРЕБУЕМЫЙ ТЕКСТ (ПРОБЕЛЫ ДОПУСКАЮТСЯ) И НАЖАТЬ "ENTER". TASWORD II БУДЕТ ПЕРЕФОРМАТИРОВАТЬ ТЕКСТ В СООТВЕТСТВИИ С РЕЖИМОМ "R.JUSTIFY"

L - МАРКЕР ВКЛЮЧЕНИЯ ПЕЧАТИ БУКВ С УДВОЕННОЙ ВЫСОТОЙ. МАРК R ПРЕДСТАВЛЯЕТ СОБОЙ СПЕЦИАЛЬНУЮ СТРОКУ, СОДЕРЖАЩУЮ СООБЩЕНИЕ: "PRINT AT DOUBL HEIGHT S", КОТОРАЯ ПЕЧАТАЕТСЯ ПЕРЕД ТРЕБУЕМОЙ СТРОКОЙ. ДЛЯ УДАЛЕНИЯ МАРК RA МОЖНО ИСПОЛЬЗОВАТЬ "NOT" В НОРМАЛЬНОМ РЕЖИМЕ. ПЕРЕД ВКЛЮЧЕНИЕМ ЭТОГО РЕЖИМА КУРСОР ДОЛЖЕН БЫТЬ В НАЧАЛЕ СТРОКИ.

К - ВСТАВКА СООБЩЕНИЯ "PRINT AT DOUBLE HEIGHT OFF". КУРСОР ДОЛЖЕН БЫТЬ В НАЧАЛЕ СТРОКИ.

С - ИЗМЕНЕНИЕ "ОКНА". СЛУЖИТ ДЛЯ ОТКРЫТИЯ ЗАКРЫТИЯ 32-Х СИМВОЛЬНОГО "ОКНА" В ТЕКСТОВОМ ФАЙЛЕ. ПРИ ЗАКРЫТОМ "ОКНЕ" ДЛИНА ТЕКСТОВОЙ СТРОКИ РАВНА 64 СИМВОЛАМ. КОГДА "ОКНО" ОТКРЫТО, ИЗМЕНЯЕТСЯ ЦВЕТ BORDER И НА ЭКРАНЕ ПОЯВЛЯЕТСЯ ЧАСТЬ ТЕКСТОВОГО ФАЙЛА ШИРИНОЙ 32 СИМВОЛА В СТРОКЕ. "ОКНО" МОЖНО ДВИГАТЬ С ПОМОЩЬЮ "ARROWS". БОКОВОЙ СДВИГ ПРОИСХОДИТ АВТОМАТИЧЕСКИ ПРИ ВВОДЕ ТЕКСТА.

В - МАРКЕР НАЧАЛА БЛОКА. БЛОК ТЕКСТА МОЖЕТ БЫТЬ ПЕРЕСЛАН ИЛИ СКОПИРОВАН В ЛЮБУЮ ЧАСТЬ ТЕКСТА. ДЛЯ УДАЛЕНИЯ МАРКА МОЖНО ИСПОЛЬЗОВАТЬ "NOT" В НОРМАЛЬНОМ РЕЖИМЕ. МАРКЕРОМ НАЧАЛА БЛОКА ЯВЛЯЕТСЯ [.

У - МАРК КОНЦА БЛОКА -].

М - ПЕРЕСЫЛКА БЛОКА. БЛОК ВСТАВЛЯЕТСЯ ПЕРЕД СТРОКОЙ, СОДЕРЖАЩЕЙ КУРСОР.

7. ПЕЧАТЬ ТЕКСТА.

ПЕРЕД ПЕЧАТЬЮ НЕОБХОДИМО ПЕРЕДАТЬ В ПРИНТЕР УПРАВЛЯЮЩИЕ КОДЫ. ДЛЯ ЭТОГО НЕОБХОДИМО ВЫЙТИ В МЕНЮ ПО "STOP" В НОРМАЛЬНОМ РЕЖИМЕ И ВЫБРАТЬ "DEFINE GRAPHICS PRINTER" НАЖАТИЕМ "G".

8. ГРАФИЧЕСКИЕ СИМВОЛЫ.

ГРАФИЧЕСКИЕ СИМВОЛЫ ИНТЕРПРЕТИРУЮТСЯ TASWORD II КАК ПОСЛЕДОВАТЕЛЬНОСТЬ УПРАВЛЯЮЩИХ КОДОВ ДЛЯ ПРИНТЕРА. TASWORD II ПОСТАВЛЯЕТСЯ С НАБОРОМ ГРАФИЧЕСКИХ СИМВОЛОВ, УКАЗАННЫХ В "HELP".

ПОСЛЕ ВХОДА В "DEFINE GRAPHICS PRINTER" НА ЭКРАНЕ ПОЯВИТСЯ СПИСОК КОДОВ ГРАФИЧЕСКИХ СИМВОЛОВ ZX - SPECTRUM И СПИСОК НАЗНАЧЕННЫХ ИМ УПРАВЛЯЮЩИХ КОДОВ ПРИНТЕРА. ДЛЯ ПЕРЕНАЗНАЧЕНИЯ НЕОБХОДИМО ИСПРАВИТЬ СООТВЕТСТВУЮЩИЙ ГРАФИЧЕСКИЙ КОД И НАЖАТЬ <ENTER>.

ЕСЛИ ВЫ НАЖАЛИ <ENTER> БЕЗ НАБОРА КОДА, TASWORD II ЗАПРОСИТ ВАС:

1. УПРАВЛЯЮЩИЙ КОД, КОТОРЫЙ ИСПОЛЬЗУЕТ ВАШ ИНТЕРФЕЙС, ВВЕДИТЕ СООТВЕТСТВУЮЩЕЕ ЧИСЛО (27 - ДЛЯ HILDERWAY, 5 - ДЛЯ EUROELEKTRONICS).

2. УПРАВЛЯЮЩИЙ КОД, КОТОРЫЙ ИСПОЛЬЗУЕТ ВАШ ПРИНТЕР, ДЛЯ ПРОТЯЖКИ ЛИСТА И ВОЗВРАТА КАРЕТКИ ЕСЛИ ПРИНТЕР ИМЕЕТ ОДИНАКОВЫЕ КОДЫ И ДЛЯ ТОГО, И ДЛЯ ДРУГОГО, ВВЕДИТЕ <0> ВМЕСТО ВТОРОГО КОДА.

3. ЛЕВАЯ ГРАНИЦА.

ОТВЕТИВ НА ЭТИ ВОПРОСЫ, ВЫ ПОЛУЧИТЕ СОБСТВЕННУЮ ВЕРСИЮ TASWORD II, КОТОРУЮ МОЖНО СОХРАНИТЬ НА ЛЕНТЕ.

ИЗМЕНЕНИЕ ЗНАКОГЕНЕРАТОРА.

ТАБЛИЦА 64-Х СИМВОЛЬНОГО ЗНАКОГЕНЕРАТОРА НАХОДИТСЯ В ОЗУ ПО АДРЕСАМ С 61184 ПО 62079, А БАЗОВЫМ АДРЕСОМ ЯВЛЯЕТСЯ - 60928. КАЖДЫЙ СИМВОЛ ОПРЕДЕЛЯЕТСЯ 8-Ю БАЙТАМИ ОБЫЧНЫМ СПОСОБОМ. ЧЕТЫРЕ СТАРШИХ БИТА КАЖДОГО БАЙТА ДОЛЖНЫ БЫТЬ РАВНЫ НУЛЮ.

ЗА ИСКЛЮЧЕНИЕМ ГРАФИЧЕСКИХ СИМВОЛОВ, 32-Х СИМВОЛЬНЫЙ ЗНАКОГЕНЕРАТОР НЕ МОЖЕТ БЫТЬ ИЗМЕНЕН. ШЕСТНАДЦАТЬ СИМВОЛОВ 32-Х СИМВОЛЬНОГО ЗНАКОГЕНЕРАТОРА ХРАНЯТСЯ ПО АДРЕСАМ 60928-61055 С БАЗОВЫМ АДРЕСОМ - 59904.

ПРОГРАММА "COPY-COPY"

1. "CAT" - клавиша "C" - просмотр списка имен файлов на экране.

2. "LOAD" - клавиша "J" - загрузка файлов в память.

LOAD - загружает программу с очередным номером.

LOAD N - загружает файл на место N. Если N=1, то загруженные перед этим файлы теряются, и загрузка производится в начало рабочей области (с адреса 23296).

LOAD N TO NN - загружает файлы с номерами от N до NN.

LOAD TO N - загружает файлы от очередного номера до N.

LOAD AT NN - загружает файлы с адреса NN, по умолчанию файл с номером 1 загружается с адреса 23296. Можно задать NN=23040, в этом случае величина рабочей области для загрузки файлов увеличивается до 42496 байтов. По умолчанию величина этой области равна 42240 байтов, очевидно, что NN < 23040 задавать нельзя.

LOAD (NN - считывает первые NN байтов файла, например: LOAD (6912 - загрузка только данных для экрана.

3. "SAVE" - клавиша "S" - сохранение файлов.

SAVE - сохраняет все загруженные файлы без пауз.

SAVE N - сохраняет все файлы, начиная с файла с номером

SAVE N TO NN - сохраняет файлы с номерами от N до NN.

SAVE TO N - сохраняет файлы с номерами от 1 до N.

SAVE STEP N - сохраняет все загруженные файлы между файлами делаются паузы в N секунд.

SAVE N TO NN STEP M - сохраняет файлы с номерами от N до NN, N - номер первого файла NN - номер последнего файла M - пауза между файлами в секундах.

4. "VERIFY" - клавиша "V" - проверка сохраненных файлов.

VERIFY - аналогично SAVE.

VERIFY N TO NN - аналогично SAVE N TO NN.

VERIFY N - аналогично SAVE N.

5. "LET" - клавиша "L" - изменение полей заголовка файла.

НАПРИМЕР: LET 2=AAA,,1

файл с номером 2 будет иметь имя AAA, стартовый адрес 1.

LET 3=500,1,5

файл с номером 3 будет иметь длину 500 байтов, стартовый адрес 1, длина программы 5 байтов.

6. "LIST" - клавиша "K" - распечатка памяти.

LIST [NN] - задает адрес памяти (по умолчанию принимается равным 0), по этой команде выводится 16 байтов памяти, для каждого из которых приводятся:

- адрес памяти
- десятичное значение байта
- десятичное значение двух смежных байтов
- символьное значение байта

для вывода следующих 16 байтов нажмите "ENTER".

7. "POKE" - клавиша "O" - изменение десятичного значения байта.

POKE X, NN X - адрес, NN - значение двух смежных байтов (>255).

POKE X, N X - адрес, N - десятичное значение байта. Если значение N лежит в диапазоне 256 - 65535, то считается, что задано значение двух смежных байтов.

8. "USR" - клавиша "U" - вызывает программу пользователя.

USR X - вызывает программу пользователя по адресу X. Для полного сброса системы задать USR 2.

9. "RETURN" - клавиша "Y" - возврат в монитор, инициализируются системные переменные и таблица каналов, полный сброс не выполняется.

10. "COPY" - клавиша "Z" - перевод программы в специальный режим для копирования файлов без заголовка длиной до 49096 байт. После выдачи команды программа загружает файл в память, а затем, по нажатию клавиши "CAPS SHIFT" выгружает ее необходимое число раз. Повторная загрузка возможна только если остается не менее 200 байт не занятой памяти.

COPY NN - копирование файла длиной до 49152 байта. Копирование выполняется только один раз.

ТИПЫ ФАЙЛОВ:

- P - ПРОГРАММА
- B - ВЫЧИСЛИТЕЛЬНЫЙ КОД (BYTES)
- A - ЧИСЛОВОЙ МАССИВ
- # - СИМВОЛЬНЫЙ МАССИВ

"ENTER" - ПОВТОРЯЕТ ПРЕДЫДУЩУЮ КОМАНДУ

"CAPS SHIFT+0" - УНИЧТОЖАЕТ НАБРАННУЮ КОМАНДУ.

АДРЕСА ПОДПРОГРАММ ПЗУ

АДРЕС	КОЛ. БАЙТ	ИМЯ	ОПИСАНИЕ
23552	8	KSTATE	П П ОПРОСА КЛАВИАТУРЫ
23560	1	LAST K	КОД ПОСЛЕДНЕЙ НАЖАТОЙ КЛАВИШИ
23561	1	REPDEL	ВРЕМЯ ОПРОСА НАЖАТОЙ КЛАВИШИ (1/50С.)
			НАЧАЛЬНОЕ ЗНАЧЕНИЕ 35.
23562	1	REPPER	ПЕРИОД ОПРОСА КЛАВИАТУРЫ (1/50С.)
			НАЧАЛЬНОЕ ЗНАЧЕНИЕ 5.
23563	2	DEF ADD	АДРЕС АРГУМЕНТА ФУНКЦИИ FN
23565	1	K DATA	ВТОРОЙ БАЙТ УПРАВЛЕНИЯ ЦВЕТОМ ВВЕДЕННЫХ
			С КЛАВИАТУРЫ.
23566	2	TVDATA	БАЙТЫ УПРАВЛЕНИЯ ЦВЕТОМ И ПОЗИЦИЕЙ (АТ,
			ТАВ) ЗНАКОВ ВЫВОДИМЫХ НА ЭКРАН.
23568	38	STRMS	
			23568 253 КАНАЛ "К"
			23570 254 КАНАЛ "S"
			23572 255 КАНАЛ "R"
			23574 0 КАНАЛ "К"
			23576 1 КАНАЛ "К"
			23578 2 КАНАЛ "S"
			23580 3 КАНАЛ "P"
			23582-23606 4-15
23606	2	CHARS	АДРЕС (-256) ГРАФИЧЕСКИХ СИМВОЛОВ
			ЗНАКОГЕНЕРАТОРА.
23608	1	RASP	ВЫСОТА СИГНАЛА (ТОН).
23609	1	PIP	ВРЕМЯ ЗВУЧАНИЯ СИГНАЛИЗАЦИИ НАЖАТОЙ
			КЛАВИШИ.
23610	1	ERR NR	НОМЕР КОДА ОШИБКИ МИНУС 1 НАЧ. ЗНАЧ. 255
23611	1	FLAGS	ФЛАЖКИ ВВОДА-ВЫВОДА БИТ N. ;
			0 -
			1 -
			2 -
			3 -
			4 -
			5 -
			6 -
			7 -
23612	1	TV FLAG	ФЛАЖКИ ЭКРАНА :
			0 -
			3 -
			4 -
			5 -

АДРЕС	КОЛ. БАЙТ	ИМЯ	ОПИСАНИЕ
23613	2	ERR SP	АДРЕС ПОДПРОГРАММЫ ОБНАРУЖЕНИЯ ОШИБКИ.
23615	2	LIST SP	АДРЕС ОБРАЩЕНИЯ К АВТОЛИСТИНГУ.
23617	1	MODE	ТИП КУРСОР. : 0 ДЛЯ L/C, 1 ДЛЯ E, 2 ДЛЯ G, 4 ДЛЯ K.
23618	2	NEWPPC	НОМЕР ЛИНИИ К КОТОРОЙ ПРОИЗВЕД Н ПЕРЕХОД.
23620	1	NSPPC	НОМЕР ИНСТРУКЦИИ В ЛИНИИ К КОТО- РОЙ ПРОИЗВЕД Н ПЕРЕХОД.
23621	2	PPC	НОМЕР АКТИВНОЙ ЛИНИИ.
23623	4	SUBPPC	НОМЕР ИНСТРУКЦИИ В АКТИВНОЙ ЛИНИИ.
23624	1	BORDCR	ЦВЕТ БОРДЮРА (*8).
23625	2	E PPC	НОМЕР ЛИНИИ В КОТОРОЙ НАХОДИТСЯ КУРСОР ПРОГРАММЫ.
23627	2	VAR\$	АДРЕС ПОЛЯ ПЕРЕМЕННЫХ ПРОГРАММЫ В BASIC.
23629	2	DEST	?
23631	2	CHANS	АДРЕС ПОЛЯ ЗАПИСИ КАНАЛОВ.
23633	2	CURCHL	АДРЕС АКТУАЛЬНОГО КАНАЛА.
23635	2	PROG	АДРЕС ПРОГРАММЫ В BASIC.
23637	2	NXTLIN	АДРЕС СЛЕДУЮЩЕЙ ЛИНИИ В ПРОГРАММЕ.
23639	2	DATADD	АДРЕС НАЧАЛА СЛЕДУЮЩИХ ДАННЫХ В СБОРЕ DATA.
23641	2	E LINE	АДРЕС ТЕКУЩЕЙ ОТРЕДАКТИРОВАННОЙ ИНСТРУКЦИИ.
23643	2	K CUR	?
23645	2	CH ADD	АДРЕС СЛЕДУЮЩЕГО ИНТЕРПРЕТИРУЕМОГО ЗНАКА.
23647	2	X PTR	?
23649	2	WORKSP	АДРЕС РАБОЧЕЙ ОБЛАСТИ ИНТЕРПРЕТАТОРА.
23651	2	STKBOT	?
23653	2	STKEND	?
23655	1	BREG	?
23656	2	MEM	АДРЕС ПАМЯТИ КАЛЬКУЛЯТОРА.
23658	1	FLAGS2	ФЛАЖКИ ; 0- 1- 2- 3- 4-
23659	1	DF SZ	ЧИСЛО ЛИНИЙ В СИСТЕМНОМ ОКНЕ.
23660	2	S TOP	НОМЕР ПЕРВОЙ ЛИНИИ АВТОЛИСТИНГА.
23662	2	OLDPPC	НОМЕР ЛИНИИ ДЛЯ CONTINUE.
23664	1	OSPPC	НОМЕР ИНСТРУКЦИИ В ЛИНИИ ДЛЯ CONTINUE
23665	1	FLAGX	ФЛАЖКИ: (НОМЕРА БИТОВ) 1- 5- 7-
23666	2	STRLEN	ДЛИНА АКТУАЛЬНОЙ СТРОКИ ЗНАКОВ.
23668	2	T ADDR	АДРЕС СЛЕДУЮЩЕЙ ПОЗИЦИИ В ТАБЛИЦЕ СИНТАКСИЧЕСКОГО АНАЛИЗА.
23670	2	SEED	ПОДПРОГРАММА ГЕНЕРАЦИИ СЛУЧАЙНЫХ ЧИСЕЛ (RND).

HISOFT PASCAL4 - VERSION 1.4

31. OCT. 1982

HISOFT PASCAL4 - VERSION 1.4

31. OCT. 1982

ЧАСТЬ 0.1

ТИП FILE НЕОПРЕДЕЛЕН.
 ТИП RECORD НЕ МОЖЕТ ИМЕТЬ ВАРИАНТОВ.
 ПРОЦЕДУРЫ И ФУНКЦИИ КАК ПАРАМЕТРЫ НЕДОПУСТИМЫ.
 КОМПИЛЯТОР ЗАНИМАЕТ ПРИМЕРНО 12 К ПАМЯТИ И 4К ЗАНИМАЕТ RUNTIMES,
 МОГУТ ИСПОЛЬЗОВАТЬСЯ СЛЕДУЮЩИЕ УПРАВЛЯЮЩИЕ КОДЫ:

ENTER - ВВОД СТРОКИ
 DELETE - УДАЛЕНИЕ СИМВОЛА
 EDIT - ВОЗВРАТ В РЕДАКТОР
 CS+8 - ПЕРЕХОД К СЛЕДУЮЩЕЙ ПОЗИЦИИ ТАБУЛЯЦИИ
 CS+5 - УДАЛЕНИЕ ВСЕЙ СТРОКИ
 CHR(16) - ПЕРЕКЛЮЧАТЕЛЬ ЭКРАН/ПРИНТЕР.
 ПОСЛЕ ЗАГРУЗКИ HP4T НА ЭКРАНЕ ПОЯВИТСЯ ВОПРОС:

TOP OF RAM ?

ВЫ ДОЛЖНЫ ВВЕСТИ ЛИБО ЦЕЛОЕ ДЕСЯТИЧНОЕ ЧИСЛО (≤ 65536) ИЛИ СРАЗУ НАЖАТЬ ENTER. ЕСЛИ ВВЕДЕНО ЧИСЛО, ТО ОНО ИСПОЛЬЗУЕТСЯ В КАЧЕСТВЕ ПОСЛЕДНЕГО АДРЕСА RAM+1, В ПРОТИВНОМ СЛУЧАЕ ВЕРШИНА RAM ВЫЧИСЛЯЕТСЯ АВТОМАТИЧЕСКИ. НА ЭТОТ АДРЕС УСТАНАВЛИВАЕТСЯ СТЕК КОМПИЛЯТОРА (Т.О. МОЖНО РЕЗЕРВИРОВАТЬ RAM ДЛЯ КАКИХ-ЛИБО ЦЕЛЕЙ RAMTOP < 65536). В ZX-SPECTRUM ВЕРШИНА RAM ЯВЛЯЕТСЯ НАЧАЛЬНЫМ АДРЕСОМ ДЛЯ UDG. ПОСЛЕ НАЖАТИЯ ENTER HP4T ЗАДАСТ СЛЕДУЮЩИЙ ВОПРОС:

TOP OF RAM FOR 'T' ?

В ОТВЕТ ВЫ МОЖЕТЕ ВВЕСТИ ЛИБО ДЕСЯТИЧНОЕ ЧИСЛО, ЛИБО ИСПОЛЬЗОВАТЬ РАНЕЕ ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЕ TOP OF RAM. ТО, ЧТО ВЫ ВВЕДЕТЕ БУДЕТ ИСПОЛЬЗОВАНО КАК СТЕК ПРИ ВЫПОЛНЕНИИ ОБЪЕКТНОГО КОДА ПОСЛЕ КОМАНДЫ 'T' РЕДАКТОРА. ПОСЛЕ ЭТОГО HP4T ВАС СПРОСИТ:

TABLE SIZE ?

ТО, ЧТО ВЫ ВВЕДЕТЕ БУДЕТ РАССМАТРИВАТЬСЯ КАК ОБЪЕМ ПАМЯТИ, ИСПОЛЬЗУЕМОЙ ПОД ТАБЛИЦУ СИМВОЛОВ. ЕСЛИ СРАЗУ НАЖАТЬ ENTER, ТО TABLE SIZE := ДОСТУПНОЕ RAM/16 (КАК ПРАВИЛО, ЭТОГО БОЛЕЕ ЧЕМ ДОСТАТОЧНО). ЕСЛИ ПЕРЕД ЧИСЛОМ ВВЕСТИ СИМВОЛ E, ТО ВСТРОЕННЫЙ РЕДАКТОР СТАНЕТ НЕДОСТУПНЫМ. ЭТА ПРОЦЕДУРА ПОЛЕЗНА ПРИ ИСПОЛЬЗОВАНИИ ВНЕШНЕГО РЕДАКТОРА

ЧАСТЬ 0.2

КОМПИЛЯЦИЯ И ЗАПУСК В РАБОТУ

КОМПИЛЯТОР ГЕНЕРИРУЕТ ЛИСТИНГ ВИДА:

XXXX NNNN ИСХОДНЫЙ ТЕКСТ СТРОКИ.

ГДЕ XXXX - АДРЕС НАЧАЛА КОДА, ГЕНЕРИРУЕМОГО ЭТОЙ СТРОКОЙ,

NNNN - НОМЕР СТРОКИ.

ЕСЛИ ДЛИНА СТРОКИ БОЛЬШЕ 80 СИМВОЛОВ, КОМПИЛЯТОР ВСТАВЛЯЕТ СИМВОЛ 'НОВАЯ СТРОКА'. ЛИСТИНГ МОЖЕТ БЫТЬ РАСПЕЧАТАН НА ПРИНТЕРЕ С ПОМОЩЬЮ ОПЦИИ *P. ДЛЯ ПРИОСТАНОВКИ ВЫВОДА ЛИСТИНГА НЕОБХОДИМО НАЖАТЬ CS+SPACE (BREAK). ЕСЛИ ПОСЛЕ ЭТОГО НАЖАТЬ EDIT - ВОЗВРАТ В РЕДАКТОР, ПРИ НАЖАТИИ ЛЮБОЙ ДРУГОЙ КЛАВИШИ - ПРОДОЛЖЕНИЕ ВЫВОДА. ЕСЛИ ПРИ КОМПИЛЯЦИИ ОБНАРУЖЕНЫ ОШИБКИ, ТО ВЫВОДЯТСЯ СООТВЕТСТВУЮЩИЕ СООБЩЕНИЯ И РАСПЕЧАТКА ЛИСТИНГА ПРИОСТАНАВЛИВАЕТСЯ. ЕСЛИ ПРИ ЭТОМ ВВЕСТИ СИМВОЛ E - СИСТЕМА ПЕРЕХОДИТ К РЕДАКТИРОВАНИЮ ИНДИЦИРУЕМОЙ СТРОКИ, ЕСЛИ ВВЕСТИ P, - К РЕДАКТИРОВАНИЮ ПРЕДЫДУЩЕЙ СТРОКИ.

ЧАСТЬ 1.

СИНТАКСИС И СЕМАНТИКА

1.1 ИДЕНТИФИКАТОРЫ.

ДОПУСТИМАЯ ДЛИНА ИДЕНТИФИКАТОРА НЕ БОЛЕЕ 10 СИМВОЛОВ. РАЗЛИЧАЮТСЯ МАЛЕНЬКИЕ И БОЛЬШИЕ БУКВЫ.

1.2, 1.3. UNSIGNED INTEGER, UNSIGNED NUMBER

ЦЕЛЫЕ ИМЕЮТ АБСОЛЮТНУЮ ВЕЛИЧИНУ НЕ БОЛЕЕ 32767. ЧИСЛА С БОЛЬШИМ ЗНАЧЕНИЕМ РАССМАТРИВАЮТСЯ КАК REAL. МАНТИССА REAL ИМЕЕТ ДЛИНУ 23 БИТ (Т.Е. 7 ЗНАЧАЩИХ ЦИФР). РЕЗУЛЬТАТЫ ОПЕРАЦИЙ С REAL ПРИБЛИЖЕННЫЕ Т.Е. $2.32002 - 2 <> 0.00002$. МАКСИМАЛЬНОЕ ЧИСЛО ТИПА REAL РАВНО $3.4E38$, МИНИМАЛЬНОЕ - $5.9E-39$.

ДЛЯ ПОВЫШЕНИЯ ТОЧНОСТИ СЛЕДУЕТ ИЗБЕГАТЬ ЛИШНИХ НУЛЕЙ ТАК 0.000123456 ПРЕДСТАВЛЯЕТСЯ МЕНЕЕ ТОЧНО, ЧЕМ $1.23456E-4$ (0 - ТОЖЕ ЧИСЛО, А ИХ МОЖЕТ БЫТЬ НЕ БОЛЕЕ 7).

ДОПУСКАЕТСЯ НЕХ ПРЕДСТАВЛЕНИЕ ЧИСЛА ПРИ ОБОЗНАЧЕНИИ АДРЕСОВ RAM: #03, #FF.

1.4 UNSIGNED CONSTANT

CONSTANT IDENTIFIER

UNSIGNED NUMBER

NIL

CHARACTER

ОТМЕТИМ, ЧТО СТРОКИ НЕ МОГУТ СОДЕРЖАТЬ БОЛЕЕ 255 СИМВОЛОВ.

СТРОКОВЫМ ТИПОМ ЯВЛЯЕТСЯ ARRAY [1..N] OF CHAR, ГДЕ N - ЦЕЛОЕ 0-255 ВКЛЮЧИТЕЛЬНО. ЛИТЕРАЛЬНЫЕ СТРОКИ НЕ ДОЛЖНЫ СОДЕРЖАТЬ СИМВОЛ EOL (CHR(13)) - В ПРОТИВНОМ СЛУЧАЕ СИСТЕМА ГЕНЕРИРУЕТ *ERROR* 68. СИМВОЛ NULL РЕАЛИЗОВАН НЕ КАК "", А КАК CHR(0).

1.5. CONSTANT

ДОПУСКАЕТСЯ ИСПОЛЬЗОВАНИЕ УПРАВЛЯЮЩИХ КОДОВ ДИСПЛЕЯ В КАЧЕСТВЕ КОНСТАНТ:

CONST BS=CHR(10)

CR=CHR(13)

1.6. SIMPLE TYPE

TYPE IDENTIFIER

(IDENTIFIER)

CONSTANT .. CONSTANT

ТИП, ОПРЕДЕЛЕННЫЙ КАК (IDENTIFIER, IDENTIFIER,.....) МОЖЕТ ИМЕТЬ НЕ БОЛЕЕ 255 ЭЛЕМЕНТОВ.

1.7. TYPE

ЗАРЕЗЕРВИРОВАННОЕ СЛОВО PASCED ДОПУСТИМО, НО ИГНОРИРУЕТСЯ Т.К. МАССИВЫ УЖЕ УПАКОВАНЫ.

1.7.1 ARRAYS & SETS

БАЗОВЫЙ ТИП SET МОЖЕТ ИМЕТЬ ДО 256 ЭЛЕМЕНТОВ. ДОПУСКАЮТСЯ МАССИВЫ МАССИВОВ, МАССИВЫ МНОЖЕСТВ, ЗАПИСИ С ПОЛЯМИ ТИПА МНОЖЕСТВО И Т.П. В ИР4Т ИСПОЛЬЗУЕТСЯ ЭКВИВАЛЕНТНОСТЬ ИМЕН, А НЕ ТИПОВ, СЛЕДОВАТЕЛЬНО:

TYPE

TABELA=ARRAY [1..100] OF INTEGER

TABELB=ARRAY [1..100] OF INTEGER

ПЕРЕМЕННОЙ TABELA НЕ МОЖЕТ БЫТЬ ПРИСВОЕНО ЗНАЧЕНИЕ ПЕРЕМЕННОЙ TABELB Т.К. ОНИ ИМЕЮТ РАЗНЫЙ ТИП.

TYPE

TABEL=ARRAY [1..100] OF INTEGER

VAR

TABELA, TABELB : TABEL

В ЭТОМ СЛУЧАЕ ПЕРЕМЕННЫЕ TABELA И TABELB ИМЕЮТ ОДИНАКОВЫЙ ТИП, Т.Е. ПРИСВОЕНИЕ TABELA=TABELB НЕ ВЫЗОВЕТ ОШИБКИ.

1.7.2. POINTERS

ИМЕЮТСЯ НЕКОТОРЫЕ ОГРАНИЧЕНИЯ НА ИСПОЛЬЗОВАНИЕ УКАЗАТЕЛЕЙ:
А) НЕДОПУСКАЮТСЯ УКАЗАТЕЛИ НА НЕОБЪЯВЛЕННЫЙ ТИП. ЭТО НЕ ОТНОСИТСЯ К КОНСТРУКЦИЯМ ТИПА:

TYPE

ITEM=RECORD

VALUE: INTEGER

NEXT: ITEM

END

LINK= ITEM

Б) НЕДОПУСКАЕТСЯ УКАЗАТЕЛЬ НА УКАЗАТЕЛЬ

В) УКАЗАТЕЛИ ОДНОГО ТИПА СЧИТАЮТСЯ ЭКВИВАЛЕНТНЫМИ, Т.Е.:

VAR

FIRST: LINK

CURRENT: ITEM

Г) ОПРЕДЕЛЕНА КОНСТАНТА NIL.

1.7.4. RECORDS

ДОПУСКАЕТСЯ КЛЮЧЕВОЕ СЛОВО WITH ДЛЯ БОЛЕЕ КОМПАКТНОГО ОБРАЩЕНИЯ К ПОЛЯМ ЗАПИСИ.

1.7.5. FIELD LIST

1.9. VARIABLE

ЕСЛИ ПЕРЕМЕННАЯ ОБЪЯВЛЕНА КАК ARRAY [1..10] OF ARRAY [1..10] OF INTEGER, ТО ДЛЯ ДОСТУПА К ЭЛЕМЕНТУ (1,1) ЭТОГО МАССИВА МОЖНО ИСПОЛЬЗОВАТЬ: A[1][1] ИЛИ A[1,1].

1.10. FACTOR

1.11. TERM

НИЖНЯЯ ГРАНИЦА МНОЖЕСТВ ВСЕГДА РАВНА 0, А РАЗМЕР МНОЖЕСТВА ВСЕГДА РАВЕН МАКСИМАЛЬНОМУ РАЗМЕРУ БАЗОВОГО ТИПА. Т.Е. SET OF CHAR ВСЕГДА ЗАНИМАЕТ 32 БАЙТА (ВОЗМОЖНЫ 255 ЭЛЕМЕНТОВ ПО БИТУ НА ЭЛЕМЕНТ), SET OF 0..10 ЭКВИВАЛЕНТНО SET OF 0..255.

1.12. SIMPLE EXPRESSION

1.13. EXPRESSION

1.14. PARAMETER LIST

1.15. STATEMENT

CASE ОПЕРАТОР

НУЛЕВОЙ CASE-СПИСОК НЕДОПУСКАЕТСЯ, Т.Е. CASE OF END ГЕНЕРИРУЕТ ОШИБКУ *ERROR*13. ОПЕРАТОР ELSE ВЫПОЛНЯЕТСЯ, ЕСЛИ СЕЛЕКТОР (СПИСОК ВЫРАЖЕНИЙ) НЕ ОБНАРУЖЕН В СПИСКЕ КОНСТАНТ. ЕСЛИ ИСПОЛЬЗОВАН ОГРАНИЧИТЕЛЬ END И СЕЛЕКТОР НЕ ОБНАРУЖЕН, ТО УПРАВЛЕНИЕ ПЕРЕДАЕТСЯ НА ОПЕРАТОР, СТОЯЩИЙ ПОСЛЕ END.

FOR ОПЕРАТОР

УПРАВЛЯЮЩАЯ ПЕРЕМЕННАЯ ОПЕРАТОРА FOR НЕ МОЖЕТ БЫТЬ СТРУКТУРНОЙ ПЕРЕМЕННОЙ ИЛИ ПАРАМЕТРОМ.

GOTO ОПЕРАТОР

ПЕРЕХОД ПО GOTO ВОЗМОЖЕН ТОЛЬКО В ПРЕДЕЛАХ ТОГО ЖЕ БЛОКА, ГДЕ ИСПОЛЬЗУЕТСЯ ОПЕРАТОР GOTO И НА ТОМ ЖЕ УРОВНЕ. МЕТКА ДОЛЖНА БЫТЬ ОБЪЯВЛЕНА В ТОМ ЖЕ БЛОКЕ, ГДЕ ОНА ИСПОЛЬЗУЕТСЯ. МЕТКА МОЖЕТ СОДЕРЖАТЬ ОТ 1 ДО 4 ЦИФР. ОПЕРАТОР GOTO НЕ МОЖЕТ БЫТЬ ИСПОЛЬЗОВАН ДЛЯ ВЫХОДА ИЗ ЦИКЛОВ FOR ... DO. ПРОЦЕДУР ИЛИ ФУНКЦИЙ.

1.16. BLOCK

ИСПОЛЬЗОВАНИЕ ПРОЦЕДУР И ФУНКЦИЙ ПЕРЕД ИХ ОПРЕДЕЛЕНИЕМ
ПРОЦЕДУРЫ И ФУНКЦИИ МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ ПЕРЕД ИХ ОПРЕ-
ДЕЛЕНИЕМ :

```
PROCEDURE A(Y:T) FORWARD
PROCEDURE B(X:T)
BEGIN
    ...
    A(P)
    ...
END
PROCEDURE A
BEGIN
    ...
    B(P)
    ...
END.
```

ЧАСТЬ 2. ПРЕДОПРЕДЕЛЕННЫЕ ИДЕНТИФИКАТОРЫ

2.1. КОНСТАНТЫ

MAXINT - МАКСИМАЛЬНОЕ ЦЕЛОЕ 32767,
TRUE, FALSE - БУЛЕВСКИЕ КОНСТАНТЫ.

2.2 ТИПЫ

INTEGER
REAL
CHAR
BOOLEAN

2.3 ПРОЦЕДУРЫ И ФУНКЦИИ

2.3.1 ПРОЦЕДУРЫ ВВОДА/ВЫВОДА

2.3.1.1 WRITE (E) - ВЫВОД 8-БИТОВОГО ЗНАЧЕНИЯ ВЫРАЖЕНИЯ E НА ЭК-
РАН ИЛИ ПРИНТЕР.

CHR(8) - (CTRL H) ВОЗВРАТ КУРСОРА НА ШАГ НАЗАД (BACKSPACE)

CHR(12) - (CTRL L) ОЧИСТКА ЭКРАНА ИЛИ НОВАЯ СТРАНИЦА ПРИНТЕ-

RE

CHR(13) - (CTRL M) ВОЗВРАТ КАРЕТКИ И НОВАЯ СТРОКА

CHR(16) - (CTRL P) ПЕРЕКЛЮЧАТЕЛЬ ВЫВОДА ЭКРАН/ПРИНТЕР

WRITE (P1,P2,...,PN) ЭКВИВАЛЕНТНО:

BEGIN WRITE (P1) WRITE(P2) ... WRITE(PN) END

ПАРАМЕТРЫ P1,P2,...,PN МОГУТ ИМЕТЬ ОДИН ИЗ СЛЕДУЮЩИХ ФОРМАТОВ:

[E] или [E:M] или [E:M:N] или [E:M:N]

ГДЕ E,M,N - ВЫРАЖЕНИЯ N - СИМВОЛЬНАЯ КОНСТАНТА.

РАССМОТРИМ 5 СЛУЧАЕВ:

1) E ИМЕЕТ ТИП INTEGER: ИСПОЛЬЗУЕТСЯ ФОРМАТ [E] ИЛИ [E:M];
ЗНАЧЕНИЕ ВЫРАЖЕНИЯ E ПРЕОБРАЗУЕТСЯ К ТИПУ CHARACTER-STRING С
ПОСЛЕДУЮЩИМИ ПРОБЕЛАМИ. ДЛИНА СТРОКИ МОЖЕТ БЫТЬ УВЕЛИЧЕНА (С ЛИ-
ДИРУЮЩИМИ ПРОБЕЛАМИ) ПРИ ИСПОЛЬЗОВАНИИ ПАРАМЕТРА M, ОПРЕДЕЛЯЮЩЕ-
ГО ОБЩЕЕ ЧИСЛО СИМВОЛОВ, КОТОРЫЕ ДОЛЖНЫ БЫТЬ ВЫВЕДЕНЫ

2) E ИМЕЕТ ТИП INTEGER И ИСПОЛЬЗОВАН ФОРМАТ [E:M:N]

В ЭТОМ СЛУЧАЕ E ВЫВОДИТСЯ В ШЕСТНАДЦАТИРИЧНОМ ВИДЕ. ЕСЛИ M=1 ИЛИ
M=2, ТО ВЫВОДИТСЯ ЗНАЧЕНИЕ (E MOD 16^M) ДЛИНОЙ НЕ БОЛЕЕ M СИМ-
ВОЛОВ. ЕСЛИ M=3 ИЛИ M=4, ТО ВЫВОДИТСЯ ПОЛНОЕ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ
E ДЛИНОЙ НЕ БОЛЕЕ 4 СИМВОЛОВ. ЕСЛИ M>4, ТО ПЕРЕД ЧИСЛОМ ВСТАВЛЯ-
ЮТСЯ ПРОБЕЛЫ. ПРИМЕР

WRITE (1025;M:N)

ЕСЛИ M=1, ТО ВЫВОДИТСЯ 1
 M=2, 01
 M=3, 0401
 M=4, 0401
 M=5, 0401.

3) E ИМЕЕТ ТИП REAL, ФОРМАТ [E], [E:M] ИЛИ [E:M:N]
 ЗНАЧЕНИЕ E ПРЕОБРАЗУЕТСЯ В СТРОКУ СИМВОЛОВ, ПРЕДСТАВЛЯЮЩУЮ СОБОЙ
 ЧИСЛО ТИПА REAL. ФОРМА ПРЕДСТАВЛЕНИЯ ОПРЕДЕЛЯЕТСЯ N. ЕСЛИ N ОТ-
 СУТСТВУЕТ, ТО ЧИСЛО ВЫВОДИТСЯ В "НАУЧНОЙ" ФОРМЕ, Т.Е. С МАНТИС-
 СОЙ И ЭКСПОНЕНТОЙ. МИНИМАЛЬНАЯ ДЛИНА "НАУЧНОГО" ПРЕДСТАВЛЕНИЯ
 РАВНА 8 СИМВОЛАМ, ЕСЛИ ВЕЛИЧИНА M<8, ТО ВСЕГДА БУДЕТ ВЫВОДИТЬСЯ
 ПОЛНАЯ ДЛИНА 12 СИМВОЛОВ (ЕСЛИ ЧИСЛО ОТРИЦАТЕЛЬНОЕ).

WRITE (-1.23E10;M)

ЕСЛИ M=7, ТО БУДЕТ ВЫВЕДЕНО -1.23000E+10
 M=8 -1.2E+10
 M=9 -1.23E+10
 M=10 -1.230E+10
 M=11 -1.2300E+10
 M=12 -1.23000E+10
 M=13 -1.23000E+10

ЕСЛИ ИСПОЛЬЗОВАН ФОРМАТ [E:N:M], ТО ЧИСЛО E БУДЕТ ВЫВОДИТЬСЯ
 С ФИКСИРОВАННОЙ ТОЧКОЙ, А N БУДЕТ ОПРЕДЕЛЯТЬ ЧИСЛО ВЫВОДИМЫХ
 ЗНАКОВ ПОСЛЕ ЗАПЯТОЙ. ЕСЛИ N=0, ТО ЧИСЛО E БУДЕТ ВЫВЕДЕНО В ВИДЕ
 INTEGER. ЕСЛИ E ЗНАЧИТЕЛЬНО БОЛЬШЕ ОПРЕДЕЛЯЕМОГО ФОРМАТА, ТО ОНО
 ВЫВОДИТСЯ В "НАУЧНОМ" ВИДЕ.

WRITE(1E2:6:2) 100.0
 WRITE(1E2:8:2) 100.00
 WRITE(23.455:6:1) 23.5
 WRITE(23.455:4:2) 2.34550E+01
 WRITE(23.455:4:0) 23

4) E ИМЕЕТ ТИП CHAR ИЛИ STRING

ВЫВОДИТСЯ ЛИБО ОДИН СИМВОЛ, ЛИБО M СИМВОЛОВ С ЛИДИРУЮЩИМИ ПРОБЕ-
 ЛАМИ, ЕСЛИ ЭТО НЕОБХОДИМО.

5) E ИМЕЕТ ТИП BOOLEAN.

2.3.1.2. WRITELN

WRITELN ДАЕТ НОВУЮ СТРОКУ, ЧТО ЭКВИВАЛЕНТНО WRITE(CHR(13)).

WRITELN (P1,P2,...,PN) ЭКВИВАЛЕНТНО :

BEGIN WRITE(P1,P2,...,PN) WRITELN END

2.3.1.3. PAGE

ПРОЦЕДУРА PAGE ЭКВИВАЛЕНТНА WRITE(CHR(12)), Т.Е. ОЧИСТКА
 ЭКРАНА ИЛИ НОВАЯ СТРАНИЦА ПРИНТЕРА.

2.3.1.4. READ

READ - ВВОД ДАННЫХ С КЛАВИАТУРЫ. ВВОД ОСУЩЕСТВЛЯЕТСЯ ЧЕРЕЗ
 БУФЕР, КОТОРЫЙ ИЗНАЧАЛЬНО СОДЕРЖИТ ТОЛЬКО СИМВОЛ EOL. ДОСТУП К
 БУФЕРУ ОСУЩЕСТВЛЯЕТСЯ ЧЕРЕЗ ТЕКСТОВОЕ ОКНО РАЗМЕРОМ В ОДИН СИМ-
 ВОЛ. ЕСЛИ ТЕКСТОВОЕ ОКНО УКАЗЫВАЕТ НА EOL-МАРКЕР, ТО ПЕРЕД ОКОН-
 ЧАНИЕМ ОПЕРАЦИИ ВВОДА В БУФЕР БУДЕТ ВВЕДЕНА НОВАЯ СТРОКА С КЛА-
 ВИАТУРЫ.

РАССМОТРИМ ПРОЦЕДУРУ READ(V) ПРИ РАЗЛИЧНЫХ ТИПАХ V:

1) V ИМЕЕТ ТИП CHAR:

В ЭТОМ СЛУЧАЕ READ(V) ПРОСТО СЧИТЫВАЕТ СИМВОЛ ИЗ БУФЕРА ВВО-
 ДА И ПРИСВАИВАЕТ ЕГО V. ЕСЛИ ТЕКСТОВОЕ ОКНО В БУФЕРЕ УСТАНОВЛЕНО
 НА МАРКЕР EOL (СИМВОЛ CHR(13)), ТО ФУНКЦИЯ EOLN БУДЕТ ВОЗВРАЩАТЬ
 ЗНАЧЕНИЕ TRUE И С КЛАВИАТУРЫ БУДЕТ СЧИТЫВАТЬСЯ НОВАЯ СТРОКА ТЕК-
 СТА. ПРИ ПОВТОРНОМ ВЫПОЛНЕНИИ ПРОЦЕДУРЫ READ, ТЕКСТОВОЕ ОКНО БУ-
 ДЕТ УСТАНОВЛЕНО НА НАЧАЛО НОВОЙ СТРОКИ.

ОТМЕТИМ, ЧТО EOLN=TRUE В НАЧАЛЕ ПРОГРАММЫ, ЭТО ОЗНАЧАЕТ, ЧТО ЕСЛИ ПЕРЕМЕННАЯ В ПЕРВОМ READ ИМЕЕТ ТИП CHAR, ТО ПРОЦЕДУРА READ ВОЗВРАТИТ ЗНАЧЕНИЕ CHR(13) С ПОСЛЕДУЮЩИМ ВВОДОМ НОВОЙ СТРОКИ С КЛАВИАТУРЫ. ПОСЛЕДУЮЩЕЕ ВЫПОЛНЕНИЕ READ ТИПА CHAR БУДЕТ ВОЗВРАЩАТЬ ПЕРВЫЙ СИМВОЛ ЭТОЙ НОВОЙ СТРОКИ.

2) V ИМЕЕТ ТИП STRING;

ПРОЦЕДУРЫ READ МОГУТ БЫТЬ СЧИТАНЫ СТРОКИ ДЛИНОЙ, ОПРЕДЕЛЕННОЙ РАНЕЕ ТИПОМ STRING ИЛИ ДО EOLN=TRUE. ЕСЛИ ПРИЗНАК EOLN УСТАНОВЛЕН ДО ОКОНЧАНИЯ ВВОДА ВСЕХ СИМВОЛОВ СТРОКИ, КОЛИЧЕСТВО КОТОРЫХ ОПРЕДЕЛЕНО ТИПОМ STRING, ТО СТРОКА БУДЕТ ДОПОЛНЕНА ДО ПОЛНОЙ ДЛИНЫ СИМВОЛАМИ CHR(0). ОБРАТИТЕ ВНИМАНИЕ НА ЗАМЕЧАНИЕ К П. 1).

3) V ИМЕЕТ ТИП INTEGER;

МАКСИМАЛЬНОЕ ЧИСЛО, КОТОРОЕ МОЖЕТ БЫТЬ ВВЕДЕНО В ЭТОМ СЛУЧАЕ, РАВНО 32767.

2.3.1.5. READLN

READLN (V1, V2, ..., VN) ЭКВИВАЛЕНТНО

BEGIN READ(V1, V2, ..., VN) READLN END

READLN ПРОСТО СЧИТЫВАЕТ НОВЫЙ БУФЕР С КЛАВИАТУРЫ. Т.О., ПОСЛЕ ВЫПОЛНЕНИЯ READLN EOLN СТАНОВИТСЯ РАВНЫМ FALSE, ЕСЛИ СЛЕДУЮЩАЯ СТРОКА НЕ ПУСТАЯ.

READLN МОЖЕТ БЫТЬ ИСПОЛЬЗОВАН ДЛЯ ПРОПУСКА ПУСТОЙ СТРОКИ В НАЧАЛЕ ВЫПОЛНЕНИЯ ОБЪЕКТНОГО КОДА. ЭТО ПОЛЕЗНО, ЕСЛИ ВЫ ЖЕЛАЕТЕ СЧИТАТЬ СИМВОЛ ТИПА CHAR В НАЧАЛЕ ПРОГРАММЫ, НО НЕ ОБЯЗАТЕЛЬНО, ЕСЛИ ВЫ ВВОДИТЕ СИМВОЛ ТИПА INTEGER ИЛИ REAL ИЛИ СИМВОЛ ИЗ ПОСЛЕДУЮЩЕЙ СТРОКИ.

2.3.2. ФУНКЦИИ ВВОДА.

2.3.2.1. EOLN

ФУНКЦИЯ EOLN ЯВЛЯЕТСЯ БУЛЕВОЙ ФУНКЦИЕЙ И ВОЗВРАЩАЕТ ЗНАЧЕНИЕ TRUE, ЕСЛИ СЛЕДУЮЩИМ ВВОДИМЫМ СИМВОЛОМ ЯВЛЯЕТСЯ EOL-СИМВОЛ. В ПРОТИВНОМ СЛУЧАЕ EOLN=FALSE.

2.3.2.2. INCH

ФУНКЦИЯ СКАНИРОВАНИЯ КЛАВИАТУРЫ. ЕСЛИ НИ ОДНА ИЗ КЛАВИШ НЕ НАЖАТА, ТО ФУНКЦИЯ ВОЗВРАЩАЕТ ЗНАЧЕНИЕ CHR(0), В ПРОТИВНОМ СЛУЧАЕ - СИМВОЛ ТИПА CHAR, СООТВЕТСТВУЮЩИЙ КОДУ НАЖАТОЙ КЛАВИШИ.

2.3.3 ФУНКЦИИ ПРЕОБРАЗОВАНИЯ

2.3.3.1. TRUNC(X); TRUNC(-1.5) = -1 TRUNC(1.9) = 1.

2.3.3.2. ROUND(X); ROUND(-6.5) = -6 ROUND(11.7) = 12
ROUND(-6.51) = -7 ROUND(23.5) = 24.

2.3.3.3. ENTIER(X); X ДОЛЖЕН ИМЕТЬ ТИП INTEGER ИЛИ REAL. ФУНКЦИЯ ВОЗВРАЩАЕТ НАИБОЛЬШЕЕ ЦЕЛОЕ ЧИСЛО, КОТОРОЕ МЕНЬШЕ ИЛИ РАВНО X ДЛЯ ЛЮБОГО X.

ENTIER(-6.5) = -7 ENTIER(11.7) = 11.

2.3.3.4. ORD(X);

2.3.3.5. CHR(X);

2.3.4. АРИФМЕТИЧЕСКИЕ ФУНКЦИИ

ABS(X) = X

SQR(X) = X*X, РЕЗУЛЬТАТ ТОГО ЖЕ ТИПА, ЧТО И X

SQRT(X) = X, РЕЗУЛЬТАТ ВСЕГДА ТИПА REAL

FRAC(X) = X-ENTIER(X), FRAC(1,5) = 0.5, FRAC(-12.56) = 0.44
 SIN(X) , РЕЗУЛЬТАТ ТИПА REAL, X В РАДИАНАХ
 COS(X)
 TAN(X)
 ARCTAN(X)
 EXP(X) = E^X, ГДЕ E=2.71828
 LN(X).

2.3.5. ДРУГИЕ ФУНКЦИИ

NEW(P)
 MARK(V1) СОХРАНИТЬ СОСТОЯНИЕ "КУЧИ" В ПЕРЕМЕННОЙ V1
 RELEASE(V1) ОЧИСТИТЬ "КУЧУ" ОТ АДРЕСА, УКАЗЫВАЕМОГО ПЕРЕМЕННОЙ V1, И ДО КОНЦА
 INLINE (C1, C2, C3 ...), CN - НЕХ ЧИСЛА ВИДА #XX
 USER(V) ВЫЗОВ ПОДПРОГРАММЫ ПО АДРЕСУ V. ДЛЯ ОБОЗНАЧЕНИЯ АДРЕСА, БОЛЬШЕ #7FFF (32767) НЕОБХОДИМО ИСПОЛЬЗОВАТЬ ОТРИЦАТЕЛЬНЫЕ ЗНАЧЕНИЯ: #C000 = (-16384). ПОДПРОГРАММА ДОЛЖНА ЗАКАНЧИВАТЬСЯ КОМАНДОЙ RET.
 HALT ОСТАНОВ ПРОГРАММЫ С ВЫДАЧЕЙ СООБЩЕНИЯ: 'HALT AT PC=XXXX', ГДЕ XXXX - НЕХ АДРЕС ЗУ В КОТОРОМ СТОИТ HALT.
 POKE(X,Y) , X - АДРЕС ПАМЯТИ ТИПА INTEGER, Y - ВЫРАЖЕНИЕ ЛЮБОГО ТИПА. POKE(#6000, 'A') - В ЯЧЕЙКУ #6000 ЗАПИСАТЬ #41.
 POKE(-16384, 3.6E3) - РАЗМЕСТИТЬ 00 0B 80 70 ПО АДРЕСУ #C000.

TOUT(NAME, START, SIZE) - ЗАПИСЬ ПЕРЕМЕННЫХ НА ЛЕНТУ. ПЕРВЫЙ ПАРАМЕТР NAME ДОЛЖЕН ИМЕТЬ ТИП ARRAY [1..8] OF CHAR. SIZE БАЙТ ДАННЫХ ВЫВОДИТСЯ НА ЛЕНТУ НАЧИНАЯ С АДРЕСА START (ОБА ПАРАМЕТРА ТИПА INTEGER) МОЖНО ИСПОЛЬЗОВАТЬ РЕАЛЬНЫЕ НЕХ ВЕЛИЧИНЫ.

TIN(NAME, START) - ВВОД С ЛЕНТЫ (СМ. TOUT). TIN МОЖНО ИСПОЛЬЗОВАТЬ ДЛЯ ВВОДА ТЕКСТОВ В МАССИВ ТИПА CHAR.

OUT(P,C) - ЗНАЧЕНИЕ ПАРАМЕТРА P ТИПА INTEGER ЗАГРУЖАЕТСЯ В РЕГИСТР ВС, СИМВОЛЬНЫЙ ПАРАМЕТР C ЗАГРУЖАЕТСЯ В РЕГИСТР A, ВЫПОЛНЯЕТСЯ КОМАНДА OUT (C), А ПРОЦЕССОРА Z80.

2.3.6. ДРУГИЕ ФУНКЦИИ

RANDOM - ГЕНЕРАЦИЯ ПСЕВДОСЛУЧАЙНОГО ЧИСЛА 0-255. ФУНКЦИЯ МАЛОЭФФЕКТИВНА, ЦЕЛЕСООБРАЗНЕЕ В НЕКОТОРЫХ СЛУЧАЯХ ИСПОЛЬЗОВАТЬ САМОПИСНЫЕ ПОДПРОГРАММЫ

SUCC(X)

PRED(X)

ODD(X)

ADDR(V) - V ПАРАМЕТР ЛЮБОГО ТИПА, РЕЗУЛЬТАТ - АДРЕС ПЕРЕМЕННОЙ V

PEEK(X, T) X ИМЕЕТ ТИП INTEGER - АДРЕС ПАМЯТИ, ВТОРОЙ АРГУМЕНТ - ТИП РЕЗУЛЬТАТА. ЕСЛИ В ЗУ, НАЧИНАЯ С #5000 ЗАПИСАНО: 50 61 73 63 61 60, ТО:

WRITE PEEK(#5000, ARRAY [1..6] OF CHAR) НАПЕЧАТАЕТ PASCAL

PEEK(#5000, CHAR)

P

PEEK(#5000, INTEGER)

24912

PEEK(#5000, REAL)

2.46227E+29

SIZE(V) - ВОЗВРАЩАЕТ ОБЪЕМ ПАМЯТИ В БАЙТАХ, ЗАНИМАЕМЫЙ ПЕРЕМЕННОЙ V.

INP(P) = IN A, (C) OF Z80.

ЧАСТЬ 3.
КОМЕНТАРИИ И ОПЦИИ КОМПИЛЯЦИИ.

3.2 ОПЦИИ КОМПИЛЯЦИИ

ОПЦИЯ L (ПО УМОЛЧАНИЮ L+)

КОНТРОЛЬ ЛИСТИНГА

L+ ПЕЧАТЬ ВСЕГО ЛИСТИНГА

L- ПЕЧАТЬ ТОЛЬКО СТРОК, СОДЕРЖАЩИХ ОШИБКИ

ОПЦИЯ O (ПО УМОЛЧАНИЮ O+), КОНТРОЛЬ ПЕРЕПОЛНЕНИЯ. ВСЕ ОПЕРАЦИИ С ЧИСЛАМИ ТИПА REAL И * И / С ЧИСЛАМИ INTEGER ВСЕГДА ПОДВЕРГАЮТСЯ ПРОВЕРКЕ НА ПЕРЕПОЛНЕНИЕ.

O+ ПРОИЗВОДИТЬ КОНТРОЛЬ ОПЕРАЦИИ + И - НАД ЧИСЛАМИ INTEGER

O- КОНТРОЛЬ НЕ ПРОИЗВОДИТЬ

ОПЦИЯ C (ПО УМОЛЧАНИЮ C+). КОНТРОЛЬ КЛАВИАТУРЫ ПРИ ВЫПОЛНЕНИИ ОБЪЕКТНОГО КОДА. НАЖАТИЕ EDIT ВЫЗЫВАЕТ ВОЗВРАТ В СИСТЕМУ С СООБЩЕНИЕМ КОМАНДЫ HALT.

C- КОНТРОЛЬ ОТСУТСТВУЕТ.

ОПЦИЯ S (ПО УМОЛЧАНИЮ S+), S+ КОНТРОЛЬ ПЕРЕПОЛНЕНИЯ СТЕКА ПРИ ВЫЗОВЕ ПРОЦЕДУР И ФУНКЦИЙ. В СЛУЧАЕ ПЕРЕПОЛНЕНИЯ СТЕКА ВЫВОДИТСЯ СООБЩЕНИЕ: "OUT OF RAM AT PC = XXXX".

ОПЦИЯ A (ПО УМОЛЧАНИЮ A+), КОНТРОЛЬ ГРАНИЦ МАССИВОВ. В СЛУЧАЕ ОШИБКИ ВЫДАЕТСЯ СООБЩЕНИЕ: "INDEX TOO HIGH" "INDEX TOO LOW".

ОПЦИЯ I (ПО УМОЛЧАНИЮ I-), ПРИ ВЫПОЛНЕНИИ ОПЕРАЦИИ СРАВНЕНИЯ >, <, >=, <= НАД 16-РАЗРЯДНЫМИ ЦЕЛЫМИ ЧИСЛАМИ, ЕСЛИ АРГУМЕНТЫ ДЕЛЯТСЯ НА ЧИСЛО >MAXINT=32767, ТО ВОЗНИКАЕТ ПЕРЕПОЛНЕНИЕ. ПРИ ЭТОМ РЕЗУЛЬТАТ СРАВНЕНИЯ СТАНОВИТСЯ НЕВЕРНЫМ. ДЛЯ ПРАВИЛЬНОГО СРАВНЕНИЯ ТАКИХ ВЕЛИЧИН НЕОБХОДИМО ИСПОЛЬЗОВАТЬ ОПЦИЮ I+. АНАЛОГИЧНАЯ СИТУАЦИЯ МОЖЕТ ВОЗНИКНУТЬ И С ЧИСЛАМИ ТИПА REAL ПРИ ПОПЫТКЕ ДЕЛЕНИЯ НА 3.4E38.

ОПЦИЯ P (ИСПОЛЬЗУЕТСЯ БЕЗ + И -), ПЕРЕКЛЮЧАТЕЛЬ ПРИНТЕР/ЭКРАН АНАЛОГИЧЕН CTRL P В IBM PC.

ОПЦИЯ F. ПОСЛЕ ИМЕНИ ЭТОЙ ОПЦИИ ДОЛЖНО ЧЕРЕЗ ПРОБЕЛ СЛЕДОВАТЬ 8-СИМВОЛЬНОЕ ИМЯ ФАЙЛА (ЕСЛИ ИМЯ < 8 СИМВОЛОВ, ЕГО ДОПОЛНЯЕТ ПРОБЕЛАМИ). ПРИ ЭТОМ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ ИЗ УКАЗАННОГО ФАЙЛА ВСТАВЛЯЕТСЯ ПОСЛЕ СТРОКИ, СОДЕРЖАЩЕЙ ОПЦИЮ F. (СМ. 4Т ВЕРСИЯ 1.5)

ЧАСТЬ 4.
ВСТРОЕННЫЙ РЕДАКТОР

КОМАНДЫ РЕДАКТОРА ИМЮТ ВИД:

C N1,N2,S1,S2

C - ВЫПОЛНЯЕМАЯ КОМАНДА,

N1,N2 - ЧИСЛА ОТ 1 ДО 32767,

S1,S2 - СТРОКИ ДО 20 СИМВОЛОВ.

РЕДАКТОР ХРАНИТ ЗНАЧЕНИЯ ПАРАМЕТРОВ, ИСПОЛЬЗОВАННЫХ В ПРЕДЫДУЩИХ КОМАНДАХ, Т.О. В РЯДЕ СЛУЧАЕВ ПАРАМЕТРЫ МОЖНО ОПУСКАТЬ. ЕСЛИ ДЛИНЫ СТРОК S1 И S2 ПРЕВЫШАЮТ 20 СИМВОЛОВ, ТО ЛИШНИЕ СИМВОЛЫ ИГНОРИРУЮТСЯ. ПАРАМЕТРЫ, УКАЗАННЫЕ ДАЛЕЕ В [], ДОЛЖНЫ ПРИСУТСТВОВАТЬ ОБЯЗАТЕЛЬНО.

КОМАНДА: I N,M. РЕЖИМ АВТОНУМЕРАЦИИ СТРОК, ВЫХОД ИЗ РЕЖИМА - CS+1 (EDIT).

КОМАНДА: L N,M. РАСПЕЧАТКА ТЕКСТА ИСХОДНОГО ФАЙЛА СО СТРОКИ N ДО СТРОКИ M ПОРЦИЯМИ, КОТОРЫЕ МОГУТ БЫТЬ ИЗМЕНЕНЫ КОМАНДОЙ 'K'. ПРИ НАЖАТИИ EDIT - ВЫХОД В РЕДАКТОР.

КОМАНДА: K N, УСТАНОВКА ПОРЦИИ ВЫВОДА ПО КОМАНДЕ L. N - ЧИСЛО СТРОК В ПОРЦИИ.

КОМАНДА: D [N,M], УДАЛЕНИЕ СТРОК НАЧИНАЯ ОТ N ДО M.

КОМАНДА: M N,M, ПЕРЕСЫЛКА СТРОКИ С НОМЕРОМ N В СТРОКУ С НОМЕРОМ M. ПРИ ЭТОМ СТРОКА С НОМЕРОМ N СТИРАЕТСЯ, А СОДЕРЖИМОЕ СТРОКИ M ЗАМЕНЯЕТСЯ НА СОДЕРЖИМОЕ СТРОКИ N.

КОМАНДА: N [N,M], ПЕРЕНУМЕРАЦИЯ ТЕКСТОВОГО ФАЙЛА: ПЕРВАЯ СТРОКА ПОЛУЧАЕТ НОМЕР N, ШАГ НУМЕРАЦИИ M.

КОМАНДА: F N,M,F,S. В ИСХОДНОМ ФАЙЛЕ МЕЖДУ СТРОКАМИ N И M ПРОИЗВОДИТСЯ ПОИСК ТЕКСТА F. ПРИ ЕГО ОБНАРУЖЕНИИ НАЙДЕННАЯ СТРОКА ОТОБРАЖАЕТСЯ НА ЭКРАНЕ И СИСТЕМА ПЕРЕХОДИТ В РЕЖИМ EDIT. ПОСЛЕ ЭТОГО, ИСПОЛЬЗУЯ КОМАНДЫ РЕЖИМА EDIT, МОЖНО ПРОДОЛЖИТЬ ПОИСК ТЕКСТА F ИЛИ ЗАМЕНИТЬ НАЙДЕННЫЙ ТЕКСТ F НА ТЕКСТ S И ДАЛЕЕ ПРОДОЛЖИТЬ ПОИСК. ОТМЕТИМ, ЧТО СТРОКИ F И S МОГУТ БЫТЬ ОПРЕДЕЛЕНЫ РАНЕЕ ДРУГИМИ КОМАНДАМИ.

КОМАНДА: E N, ВКЛЮЧЕНИЕ РЕЖИМА РЕДАКТИРОВАНИЯ EDIT.

ПОДКОМАНДЫ:

ПРОБЕЛ - УВЕЛИЧЕНИЕ ТЕКСТОВОГО УКАЗАТЕЛЯ НА +1,

DELETE - УМЕНЬШЕНИЕ ТЕКСТОВОГО УКАЗАТЕЛЯ НА -1,

CS+8 - ТАБУЛЯЦИЯ,

ENTER - КОНЕЦ РЕДАКТИРОВАНИЯ С СОХРАНЕНИЕМ ВНЕСЕННЫХ ИЗМЕНЕНИЙ,

Q - ВЫХОД ИЗ РЕДАКТИРОВАНИЯ, СТРОКА ОСТАЕТСЯ БЕЗ ИЗМЕНЕНИЙ,

R - ПЕРЕЗАГРУЗКА БУФЕРА РЕДАКТИРОВАНИЯ, Т.Е. УДАЛЕНИЕ ВСЕХ ИЗМЕНЕНИЙ И ПОВТОРНОЕ РЕДАКТИРОВАНИЕ СТРОКИ,

L - ПЕЧАТЬ ОСТАВШЕЙСЯ ЧАСТИ СТРОКИ (ЗА ТЕКСТОВЫМ УКАЗАТЕЛЕМ) И ПОВТОРНОЕ РЕДАКТИРОВАНИЕ,

K - УДАЛЕНИЕ СИМВОЛА В ПОЗИЦИИ УКАЗАТЕЛЯ,

Z - УДАЛЕНИЕ ВСЕХ СИМВОЛОВ ОТ ПОЗИЦИИ УКАЗАТЕЛЯ ВКЛЮЧИТЕЛЬНО И ДО КОНЦА СТРОКИ,

F - ПОИСК СЛЕДУЮЩЕЙ СТРОКИ, СОДЕРЖАЩЕЙ ТЕКСТ F, ОПРЕДЕЛЕННЫЙ РАНЕЕ В КОМАНДЕ F,

S - ЗАМЕНА РАНЕЕ ОПРЕДЕЛЕННЫМ ТЕКСТОМ S ТЕКСТА F И АВТОМАТИЧЕСКОЕ ВЫПОЛНЕНИЕ ПОДКОМАНДЫ 'F', Т.Е. ПОИСК СЛЕДУЮЩЕЙ СТРОКИ, СОДЕРЖАЩЕЙ F.

НЕОБХОДИМО ОТМЕТИТЬ, ЧТО В ДАННОЙ ВЕРСИИ ПОДКОМАНДА 'S' ДОЛЖНА ВЫПОЛНЯТЬСЯ НЕПОСРЕДСТВЕННО ПОСЛЕ КОМАНДЫ 'F' ИЛИ ПОДКОМАНД 'F' ИЛИ 'S'.

I - ВСТАВКА СИМВОЛА В ТЕКУЩУЮ ПОЗИЦИЮ УКАЗАТЕЛЯ. РЕЖИМ СОХРАНЯЕТСЯ ДО НАЖАТИЯ КЛАВИШИ ENTER. ДОПУСКАЕТСЯ ИСПОЛЬЗОВАНИЕ DELETE И CS+8 (УДАЛЕНИЕ И ТАБУЛЯЦИЯ),

X - УСТАНОВКА ТЕКСТОВОГО УКАЗАТЕЛЯ В КОНЕЦ СТРОКИ.

КОМАНДЫ ОБМЕНА С МАГНИТОФОНОМ

КОМАНДА: P N,M,S - ЗАПИСЬ НА ЛЕНТУ СТРОК N-M ИСХОДНОГО ТЕКСТА С ИМЕНЕМ S,

КОМАНДА: G,,S N - ЧТЕНИЕ С ЛЕНТЫ ФАЙЛА С ИМЕНЕМ S. ЕСЛИ НЕКОТОРЫЙ ТЕКСТ УЖЕ СУЩЕСТВУЕТ В ПАМЯТИ, ТО ВВОДИМЫЙ С ЛЕНТЫ ТЕКСТ ПРИСТЫКУЕТСЯ К ИМЕЮЩЕМУСЯ И ВСЬ ФАЙЛ ПРОНУМЕРУЕТСЯ С ШАГОМ 1. ДЛЯ ПРЕРЫВАНИЯ ВВОДА МОЖНО ИСПОЛЬЗОВАТЬ BREAK.

КОМПИЛЯЦИЯ И ЗАПУСК В РАБОТУ

КОМАНДА: C N — КОМПИЛЯЦИЯ СО СТРОКИ N,

КОМАНДА: R — ВЫПОЛНЕНИЕ РАНЕЕ СКОМПИЛИРОВАННОГО ОБЪЕКТ-НОГО КОДА, ХРАНИМОГО В БУФЕРЕ;

КОМАНДА: T N — КОМАНДА ТРАНСЛЯЦИИ. ИСХОДНЫЙ ТЕКСТ КОМПИЛИРУЕТСЯ НАЧИНАЯ СО СТРОКИ N И, В СЛУЧАЕ УСПЕШНОЙ КОМПИЛЯЦИИ, СИСТЕМА ВЫВОДИТ ВОПРОС 'OK?'. ПРИ ОТВЕТЕ 'Y' ОБЪЕКТНЫЙ КОД, ПРОИЗВЕДЕННЫЙ ПРИ КОМПИЛЯЦИИ, ПЕРЕСЫЛАЕТСЯ В КОНЕЦ RUNTIMES (РАЗРУШАЯ ПРИ ЭТОМ КОМПИЛЯТОР), А ЗАТЕМ RUNTIMES И ОБЪЕКТНЫЙ КОД ВЫВОДЯТСЯ НА ЛЕНТУ С ИМЕНЕМ, ОПРЕДЕЛЕННЫМ РАНЕЕ В КАЧЕСТВЕ ПАРАМЕТРА F. ПОСЛЕ ЭТОГО ФАЙЛ F МОЖНО ЗАГРУЗИТЬ В ПАМЯТЬ ИСПОЛЬЗУЯ КОМАНДУ LOAD"".

ДРУГИЕ КОМАНДЫ

КОМАНДА: B — ВЫХОД В BASIC.

КОМАНДА: O N,M — ПРЕОБРАЗОВАНИЕ ИСХОДНОГО ТЕКСТА, ПОДГОТОВЛЕННОГО В ДРУГОМ РЕДАКТОРЕ HP4T, В ФОРМАТ, ПРИНЯТЫЙ В HISOFT HP4T,

КОМАНДА: S,D — ИЗМЕНЕНИЕ РАЗДЕЛИТЕЛЯ ПАРАМЕТРОВ НА D. ПРОБЕЛ В КАЧЕСТВЕ РАЗДЕЛИТЕЛЯ НЕ ДОПУСКАЕТСЯ.

ПРИЛОЖЕНИЕ 1. СООБЩЕНИЯ ОБ ОШИБКАХ

1. ЧИСЛО СЛИШКОМ ВЕЛИКО.
2. ОЖИДАЛОСЬ " ".
3. НЕОБЪЯВЛЕННЫЙ ИДЕНТИФИКАТОР.
4. ОЖИДАЛСЯ ИДЕНТИФИКАТОР.
5. ИСПОЛЬЗУЙТЕ В ОБЪЯВЛЕНИИ КОНСТАНТ НЕ ":", А "=".
6. ОЖИДАЛОСЬ "=".
7. ВЫРАЖЕНИЕ НЕ МОЖЕТ НАЧИНАТЬСЯ ЭТИМ ИДЕНТИФИКАТОРОМ.
8. ОЖИДАЛОСЬ ":",
9. ОЖИДАЛОСЬ ")",
10. ОШИБОЧНЫЙ ТИП.
11. ОЖИДАЛОСЬ ".",
12. ОЖИДАЛСЯ ФАКТОР.
13. ОЖИДАЛАСЬ КОНСТАНТА.
14. ЭТОТ ИДЕНТИФИКАТОР НЕ КОНСТАНТА.
15. ОЖИДАЛОСЬ "THEN",
16. ОЖИДАЛОСЬ "DO",
17. ОЖИДАЛОСЬ "TO" ИЛИ "DOWNT",
18. ОЖИДАЛОСЬ "(",
19. НЕЛЬЗЯ ВЫВЕСТИ ЭТОТ ТИП ВЫРАЖЕНИЯ.
20. ОЖИДАЛОСЬ "OF",
21. ОЖИДАЛОСЬ ",",
22. ОЖИДАЛОСЬ ":",
23. ОЖИДАЛОСЬ "PROGRAM",
24. ОЖИДАЛАСЬ ПЕРЕМЕННАЯ, Т.К. ПАРАМЕТР ЯВЛЯЕТСЯ ПАРАМЕТРОМ-ПЕРЕМЕННОЙ.
25. ОЖИДАЛОСЬ "BEGIN",
26. В ВЫЗОВЕ READ ОЖИДАЛАСЬ ПЕРЕМЕННАЯ.
27. НЕЛЬЗЯ СРАВНИВАТЬ ВЫРАЖЕНИЯ ЭТОГО ТИПА.
28. ДОЛЖЕН БЫТЬ ТИП INTEGER ИЛИ REAL.
29. НЕВОЗМОЖНО СЧИТАТЬ ЭТОТ ТИП ПЕРЕМЕННЫХ.

30. ЭТОТ ИДЕНТИФИКАТОР НЕ ЯВЛЯЕТСЯ ТИПОМ.
31. В ЧИСЛЕ REAL ОЖИДАЛАСЬ ЭКСПОНЕНТА.
32. ОЖИДАЛОСЬ СКАЛЯРНОЕ ВЫРАЖЕНИЕ (НЕ ЧИСЛОВОЕ).
33. ПУСТАЯ СТРОКА НЕДОПУСТИМА (ИСПОЛЬЗУЙТЕ CHR(0)).
34. ОЖИДАЛОСЬ "[".
35. ОЖИДАЛОСЬ "]".
36. ИНДЕНКС МАССИВА ДОЛЖЕН ИМЕТЬ СКАЛЯРНЫЙ ТИП.
37. ОЖИДАЛОСЬ "...".
38. ОЖИДАЛОСЬ "]" ИЛИ "," В ОБЪЯВЛЕНИИ ARRAY.
39. НИЖНЯЯ ГРАНИЦА БОЛЬШЕ ВЕРХНЕЙ.
40. МНОЖЕСТВО СЛИШКОМ ВЕЛИКО (БОЛЬШЕ 255 ЭЛЕМЕНТОВ).
41. РЕЗУЛЬТАТОМ ФУНКЦИИ ДОЛЖЕН БЫТЬ ИДЕНТИФИКАТОР ТИПА.
42. В МНОЖЕСТВЕ ОЖИДАЛОСЬ "," ИЛИ "]".
43. В МНОЖЕСТВЕ ОЖИДАЛОСЬ "...", ИЛИ "," ИЛИ "]".
44. ТИП ПАРАМЕТРА ДОЛЖЕН БЫТЬ ИДЕНТИФИКАТОРОМ ТИПА.
45. ПУСТОЕ МНОЖЕСТВО НЕ МОЖЕТ БЫТЬ ПЕРВЫМ ФАКТОРОМ В ОПЕРАТОРЕ.
46. ОЖИДАЛАСЬ СКАЛЯРНАЯ ВЕЛИЧИНА, ВКЛЮЧАЯ REAL.
47. ОЖИДАЛАСЬ СКАЛЯРНАЯ ВЕЛИЧИНА, ИСКЛЮЧАЯ REAL.
48. МНОЖЕСТВА НЕСОВМЕСТИМЫ.
49. "<" ИЛИ ">" НЕ МОГУТ ИСПОЛЬЗОВАТЬСЯ ДЛЯ СРАВНЕНИЯ МНОЖЕСТВ.
50. ОЖИДАЛИСЬ "FORWARD", "LABEL", "CONST", "VAR", "TYPE" ИЛИ "BEGIN".
51. ОЖИДАЛОСЬ НЕХ ЧИСЛО.
52. НЕЛЬЗЯ ПРИМЕНЯТЬ РОКЕ КО МНОЖЕСТВАМ.
53. МАССИВ СЛИШКОМ ВЕЛИК (>64K).
54. В ОПРЕДЕЛЕНИИ "RECORD" ОЖИДАЛИСЬ "END" ИЛИ " ".
55. ОЖИДАЛСЯ ИДЕНТИФИКАТОР ПОЛЯ.
56. ПОСЛЕ "WITH" ОЖИДАЛАСЬ ПЕРЕМЕННАЯ.
57. ПЕРЕМЕННАЯ В "WITH" ДОЛЖНА ИМЕТЬ ТИП "RECORD".
58. ИДЕНТИФИКАТОР ПОЛЯ НЕ БЫЛ СВЯЗАН С ОПЕРАТОРОМ "WITH".
59. ПОСЛЕ "LABEL" ОЖИДАЛОСЬ ЦЕЛОЕ БЕЗ ЗНАКА..
60. ПОСЛЕ "GOTO" ОЖИДАЛОСЬ ЦЕЛОЕ БЕЗ ЗНАКА.
61. МЕТКА НА ОШИБОЧНОМ УРОВНЕ.
62. НЕОБЪЯВЛЕННАЯ МЕТКА.
63. ПАРАМЕТРОМ "SIZE" ДОЛЖНА БЫТЬ ПЕРЕМЕННАЯ.
64. ДЛЯ УКАЗАТЕЛЕЙ МОЖЕТ ИСП. ТОЛЬКО ПРОВЕРКА НА РАВЕНСТВО.
67. ТОЛЬКО ВЫВОДИМЫЙ ПАРАМЕТР ДЛЯ ЦЕЛЫХ С ДВУМЯ ";" ИМЕЕТ ФОРМУ E:M:N.
68. СТРОКА НЕ МОЖЕТ СОДЕРЖАТЬ СИМВОЛ EOL.

ПРИЛОЖЕНИЕ 3. ПРЕДСТАВЛЕНИЕ ДАННЫХ В ЗУ.

3.1. ПРЕДСТАВЛЕНИЕ ДАННЫХ

3.1.1. ЦЕЛЫЕ, ЦЕЛЫЕ ЗАНИМАЮТ В ПАМЯТИ 2 БАЙТА.

1 -- #0001
256 -- #0100
-256 -- #FF00

ДЛЯ ОПЕРАЦИИ С ЦЕЛЫМИ ИСПОЛЬЗУЕТСЯ РЕГИСТР HL.

3.1.2. СИМВОЛЬНЫЕ, БУЛЕВСКИЕ И ДРУГИЕ СКАЛЯРНЫЕ.

ЗАНИМАЮТ ПО 1 БАЙТУ В ПАМЯТИ.

TRUE ИМЕЕТ ЗНАЧЕНИЕ 1, А FALSE - 0. ДЛЯ ОПЕРАЦИИ С ЭТИМИ ТИПАМИ ИСПОЛЬЗУЕТСЯ РЕГИСТР A.

3.1.3. REAL. ВЕЩЕСТВЕННЫЕ ХРАНЯТСЯ В ФОРМЕ МАНТИССА-ПОРЯДОК В 4Х БАЙТАХ.

ДЛЯ ОПЕРАЦИИ НАД ВЕЩЕСТВЕННЫМИ ЧИСЛАМИ ИСПОЛЬЗУЮТСЯ РЕГИСТРЫ DE И HL. D - ДЛЯ ХРАНЕНИЯ ПОРЯДКА, E, H, L - ДЛЯ ХРАНЕНИЯ МАНТИССЫ, СТАРШИЙ РАЗРЯД РЕГИСТРА H - ДЛЯ ХРАНЕНИЯ ЗНАКА ЧИСЛА. Т.О. :

2 = 01000000 00000000 00000000 00000001 = #40 #00 #00 #01
1 = 01000000 00000000 00000000 00000000 = #40 #00 #00 #00
-12.5 = 11100100 00000000 00000000 00000011 = #E4 #00 #00 #03

0.1 = 01100110 01100110 01100110 11111100 = #66 #66 #66 #FC
 H L E D

3.1.4. ЗАПИСИ И МАССИВЫ.

ЗАПИСИ ЗАНИМАЮТ В ПАМЯТИ ПРОСТРАНСТВО, РАВНОЕ СУММЕ ИХ КОМПОНЕНТ. МАССИВЫ ЗАНИМАЮТ N*S БАЙТ, ГДЕ N - РАЗМЕРНОСТЬ МАССИВА, S - ЧИСЛО БАЙТ, ЗАНИМАЕМОЕ ОДНИМ ЭЛЕМЕНТОМ.

3.1.5. МНОЖЕСТВА.

МНОЖЕСТВА ХРАНЯТСЯ В ВИДЕ БИТОВЫХ СТРОК И, ЕСЛИ БАЗОВЫЙ ТИП ИМЕЕТ N ЭЛЕМЕНТОВ, ТО ЧИСЛО ЗАНИМАЕМЫХ БАЙТ РАВНО:

(N-1) DIV (8+1), Т. Е.

SET OF CHAR ЗАИМ Т 256-1 DIV 8+1 = 32 БАЙТА.

SET OF (BLUE, GREEN, YELLOW) - 3-1 DIV 8+1 = 1 БАЙТ.

3.1.6. УКАЗАТЕЛИ.

УКАЗАТЕЛИ ЗАНИМАЮТ 2 БАЙТА ПАМЯТИ, СОДЕРЖАЩИХ АДРЕС ПЕРЕМЕННОЙ (МЛАДШИЙ БАЙТ ПЕРВЫЙ).

3.2. ХРАНЕНИЕ ПЕРЕМЕННЫХ В RUNTIMES

3.2.1 ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ ХРАНЯТСЯ, НАЧИНАЯ ОТ 'МАКУШКИ' СТЕКА RUNTIMES ВНИЗ, Т.Е. ЕСЛИ СТЕК НАЧИНАЕТСЯ С #0000 И ГЛОБАЛЬНЫМИ ПЕРЕМЕННЫМИ ЯВЛЯЮТСЯ:

VAR I:INTEGER

CH:CHAR

X:REAL

ТО I РАСПОЛОЖИТСЯ В AFEE-AFFF,

CH AFFD,

X AFF9-AFFC.

ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ. К НИМ НЕТ ДОСТУПА ЧЕРЕЗ СТЕК. ВМЕСТО ЭТОГО В НАЧАЛЕ КАЖДОГО ВНУТРЕННЕГО БЛОКА УСТАНАВЛИВАЕТСЯ РЕГИСТР IX ТАК, ЧТОБЫ (IX-4) УКАЗЫВАЛО НА НАЧАЛО БЛОКА ЛОКАЛЬНЫХ ПЕРЕМЕННЫХ. Т.О., ЕСЛИ:

PROCEDURE TEST

VAR I,J:INTEGER

ТО I БУДЕТ НАХОДИТСЯ В IX-4-2 И IX-4-1, Т.Е. В IX-6, IX-5,
 J IX-8 И IX-7.

3.2.3. ПАРАМЕТРЫ И ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ

ПАРАМЕТРЫ-ЗНАЧЕНИЯ ТРАКТУЮТСЯ КАК ЛОКАЛЬНЫЕ ПЕРЕМЕННЫЕ И, КАК И ЭТИ ПЕРЕМЕННЫЕ, ЗАНИМАЮТ СТАРШИЕ АДРЕСА ПАМЯТИ. ОДНАКО, В ОТЛИЧИЕ ОТ ПЕРЕМЕННЫХ, В (IX+2) ФИКСИРУЕТСЯ МЛАДШИЙ АДРЕС, А НЕ ВЕРШИНА ОБЛАСТИ. ЕСЛИ:

PROCEDURE TEST (I:REAL J:INTEGER)

ТО J (РАСПОЛАГАЕТСЯ ПЕРВЫМ) БУДЕТ НАХОДИТСЯ В IX+2 И IX+3,
 A I ОТ IX+4 ДО IX+7.

ПАРАМЕТРЫ-ПЕРЕМЕННЫЕ РАСМАТРИВАЮТСЯ ТАКЖЕ, КАК И ПАРАМЕТРЫ-ЗНАЧЕНИЯ, С ТОЙ ЛИШЬ РАЗНИЦЕЙ, ЧТО ПЕРВЫЕ ЗАНИМАЕТ 2 БАЙТА ПАМЯТИ, ХРАНЯЩИХ АДРЕСА ЭТИХ ПЕРЕМЕННЫХ.

ЕСЛИ: PROCEDURE TEST (I:INTEGER VAR X:REAL)

ТО ССЫЛКА НА X РАЗМЕЩАЕТСЯ В IX+2 И IX+3 (ЭТИ ЯЧЕЙКИ СОДЕРЖАТ АДРЕС ПЕРЕМЕННОЙ X), А I - В IX+4, IX+5.

ВОЗВРАЩАЕМЫЕ ЗНАЧЕНИЯ РАЗМЕЩАЮТСЯ НАД ПЕРВЫМ ПАРАМЕТРОМ.

FUNCTION TEST (I:INTEGER):REAL

I РАЗМЕСТИТСЯ В IX+2, IX+3, А ЯЧЕЙКИ С IX+4 ПО IX+7 ЗАРЕЗЕРВИРУЮТСЯ ПОД ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ.

РЕАЛИЗАЦИЯ NISOFT PASCAL 4T НА 48K ZX-SPECTRUM.

НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ ОДИНОЧНЫЕ СИМВОЛЫ <=, <>, >=, А ТОЛЬКО КОМБИНАЦИИ <, > И =.

УПРАВЛЕНИЕ ОСУЩЕСТВЛЯЕТСЯ ТОЛЬКО ВРЕМЕННЫМИ АТТРИБУТАМИ ЗНАКОМЕСТ ПОСРЕДСТВОМ СТАНДАРТНЫХ УПРАВЛЯЮЩИХ КОДОВ (WRITE(SNR(17)), WRITE(SNR(4) - ЗЕЛЕННЫЙ 'PAPER') НЕЛЬЗЯ ИЗМЕНЯТЬ ПОСТОЯННЫЕ АТТРИБУТЫ.

ВМЕСТЕ С ОБЕКТНЫМ КОДОМ ВСЕГДА АВТОМАТИЧЕСКИ ЗАПИСЫВАЕТСЯ ЗАГРУЗЧИК. ЕСЛИ ВЫ ИСПОЛЬЗОВАЛИ КОМАНДУ 'Т' ДЛЯ ЗАПИСИ ОБЕКТНОГО КОДА И RUNTIMES НА ЛЕНТУ, ТО ДЛЯ ЗАПУСКА КОДА В РАБОТУ ДОСТАТОЧНО ВЫПОЛНИТЬ LOAD "".

ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ КОДА ЕГО МОЖНО ЗАПУСТИТЬ ПОВТОРНО, ВВЕДЯ GOTO 7.

ЕСЛИ ВЫ ВЫШЛИ ИЗ ПАСКАЛЯ ПО КОМАНДЕ 'В' ДЛЯ ВОЗВРАТА В РЕДАКТОР ВВЕДИТЕ GOTO 9 ('ТЕПЛЫЙ' СТАРТ С СОХРАНЕНИЕМ ИСХОДНОГО ТЕКСТА) ИЛИ GOTO 12 ('ХОЛОДНЫЙ' СТАРТ С ИНИЦИАЛИЗАЦИЕЙ).

ДЛЯ УСТАНОВКИ ЦВЕТА INK ИСПОЛЬЗУЙТЕ SNR(15).

PRODUCT APPLICATION NOTE PS1.1 ЗВУК И ГРАФИКА ZX-SPECTRUM В ПАСКАЛЕ 4T

1. ЗВУК (СМ. ПАКЕТ MPACK К НР88).
- ПРИМЕР: ВЕЕР(262,0.5) ВЕЕР(0,1) - ГЕНЕРАЦИЯ СРЕДНЕГО ДО В ТЕЧЕНИЕ 0,5 СЕК., ЗАТЕМ ПАУЗА НА 1 СЕК.
2. ГРАФИКА (СМ. ПАКЕТ MPACK).
3. ВЫВОД ЧЕРЕЗ СИСТЕМНОЕ ROM.

СУЩЕСТВУЮТ СИТУАЦИИ, КОГДА ПОЛЕЗНО ОСУЩЕСТВЛЯТЬ ВЫВОД ЧЕРЕЗ СИСТЕМНУЮ ТОЧКУ ВХОДА RST #10. К ПРИМЕРУ, ПРИ ИСПОЛЬЗОВАНИИ УПРАВЛЯЮЩЕГО КОДА PRINT AT, ЗА КОТОРЫМ ДОЛЖНЫ СЛЕДОВАТЬ 8-БИТНЫЕ ЗНАЧЕНИЯ X И Y. ОДНАКО С ПОМОЩЬЮ ОПЕРАТОРА WRITE ОПРЕДЕЛЕННЫЕ ЗНАЧЕНИЯ ВЫВЕСТИ НЕВОЗМОЖНО (К ПРИМЕРУ, КОД 8 СООТВЕТСТВУЕТ VASKSPACE (DELETE)). ДЛЯ ЭТОГО ЦЕЛЕСООБРАЗНО ИСПОЛЬЗОВАТЬ ПРОЦЕДУРУ SROUT (C:SNAR).

ИСПОЛЬЗОВАНИЕ ГРАФИЧЕСКОГО ПАКЕТА TURTLE

ПАКЕТ ПО КОМАНДЕ G, TURTLE ПРИСОЕДИНЯЕТСЯ К ЛЮБОЙ ПРОГРАММЕ (ОТМЕТИМ, ОДНАКО, ЧТО ДЛЯ НОРМАЛЬНОЙ РАБОТЫ ПАКЕТ ДОЛЖЕН БЫТЬ СНАБЖЕН PROGRAM-ЗАГОЛОВКОМ И VAR-ДЕКЛАРАЦИЕЙ, ТИПЕ, CONST И LABEL ОБЯЗАТЕЛЬНО. КРОМЕ ТОГО, ПРОГРАММА НЕ ДОЛЖНА СОДЕРЖАТЬ ПРЕДВАРИТЕЛЬНО ОБЯВЛЕННЫХ ПРОЦЕДУР И ФУНКЦИЙ.). ИНФОРМАЦИЯ О 'ЧЕРЕПАШКЕ' ХРАНИТСЯ В ГЛОБАЛЬНЫХ ПЕРЕМЕННЫХ.

ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ

HEADING

ИСПОЛЬЗУЕТСЯ ДЛЯ ХРАНЕНИЯ УГЛОВОГО ЗНАЧЕНИЯ НАПРАВЛЕНИЯ, В КОТОРОМ В ДАННЫЙ МОМЕНТ ОРИЕНТИРОВАНА 'ЧЕРЕПАШКА'. ПРИНИМАЕТ ЗНАЧЕНИЯ REAL В ГРАДУСАХ И МОЖЕТ БЫТЬ УСТАНОВЛЕНА В 0 ПРОЦЕДУРОЙ TURTLE. ЗНАЧЕНИЕ 0 СООТВЕТСТВУЕТ ВОСТОКУ.

XCOR, YCOR - ХРАНЯТ ТЕКУЩИЕ КООРДИНАТЫ (X,Y) 'ЧЕРЕПАШКИ' ИМЕЮТ ТИП REAL И ИЗНАЧАЛЬНО РАВНЫ 127 И 87 СООТВЕТСТВЕННО.

PENSTATUS - ИМЕЕТ ТИП INTEGER. ИСПОЛЬЗУЕТСЯ ДЛЯ ХРАНЕНИЯ СОСТОЯНИЯ 'ПЕРА'

0 - 'ПЕРО' ОПУЩЕНО, 1 - ПОДНЯТО.

ПРОЦЕДУРЫ

INK (C:INTEGER), C = 0 - 8

PAPER (C:INTEGER), C = 0 - 8

COPY - КОПИРОВАНИЕ КРАНА НА ПРИНТЕР

PENDOWN (C:INTEGER) - ПЕРЕВОД 'ЧЕРЕПАШКИ' В СОСТОЯНИЕ, ПРИ КОТОРОМ ОНА ПРИ ДВИЖЕНИИ БУДЕТ ОСТАВЛЯТЬ СЛЕД ЦВЕТОМ C.

PENUP - ПРОЦЕДУРА, ОБРАТНАЯ PENDOWN. (ОБЕ ПРОЦЕДУРЫ УСТАНОВЛИВАЮТ ПЕРЕМЕННУЮ PENSTATUS)

SETHD (A:REAL) - НАЗНАЧАЕТ ПАРАМЕТР ТИПА REAL ГЛОБАЛЬНОЙ ПЕРЕМЕННОЙ.

HEADING, - УСТАНАВЛИВАЕТ НАПРАВЛЕНИЕ ДВИЖЕНИЯ. 0-EAST, 90-NORTH, 180-WEST, 270-SOUTH. Т.Е. ПРОТИВ ЧАСОВОЙ СТРЕЛКИ.

SETXY (X,Y:REAL)

УСТАНАВЛИВАЕТ АБСОЛЮТНУЮ ПОЗИЦИЮ 'ЧЕРЕПАХИ'.

FWD (L:REAL) - ДВИГАЕТ 'ЧЕРЕПАХУ' ВПЕРЕД НА L ЕДИНИЦ В НАПРАВЛЕНИИ, ОПРЕДЕЛЯЕМОМ ПЕРЕМЕННОЙ HEADING.

BACK (L:REAL) - ДВИГАЕТ 'ЧЕРЕПАХУ' НАЗАД НА L ЕДИНИЦ (ОБРАТНА FWD).

TURN (A:REAL) - ПОВОРОТ ГОЛОВЫ 'ЧЕРЕПАХИ' НА A ГРАДУСОВ. УВЕЛИЧЕНИЕ A СООТВЕТСТВУЕТ ПОВОРОТУ ПРОТИВ ЧАСОВОЙ СТРЕЛКИ.

VECTOR (A,L:REAL) - СМЕЩАЕТ ПОЗИЦИЮ 'ЧЕРЕПАХИ' НА L ЕДИНИЦ В НАПРАВЛЕНИИ A. ПОСЛЕ ПЕРЕМЕЩЕНИЯ ГОЛОВА ОСТАЕТСЯ В A-НАПРАВЛЕНИИ.

RIGHT (A:REAL)/ LEFT (A:REAL) - ПОВОРОТ ГОЛОВЫ 'ЧЕРЕПАХИ' ВПРАВО/ВЛЕВО НА A ГРАДУСОВ.

ARC R (R:REAL, A:INTEGER) - 'ЧЕРЕПАХА' ИДЕТ ПО ДУГЕ ОКРУЖНОСТИ РАЗМЕРОМ R, ДЛИНА ДУГИ ОПРЕДЕЛЕНА В A. 'ЧЕРЕПАХА' ИДЕТ ПО ЧАСОВОЙ СТРЕЛКЕ. ОБЫЧНО R = ДО 0.5.

TURTLE - ИНИЦИАЛИЗАЦИЯ ГЛОБАЛЬНЫХ ПЕРЕМЕННЫХ.

ПРИМЕР ПРОГРАММЫ

ОТМЕТИМ, ЧТО ПАКЕТ TURTLE РАЗМЕЩАЕТСЯ, НАЧИНАЯ СО СТРОКИ 10 И ЗАКАНЧИВАЕТСЯ СТРОКОЙ 1350.

1. КРУГИ

```

1 PROGRAM CIRCLES
2 VAR I:INTEGER
1360 BEGIN
1370 TURTLE
1380 FOR I:=1 TO 9 DO .
1390 BEGIN
1400     ARC R (0.5,360)
1410     RIGHT (40)
1420 END
1430 END.
```

2. СПИРАЛИ

```

1 PROGRAM SPIRALS
2 VAR
1360 PROCEDURE SPIRALS (L,A:REAL)
1370 BEGIN
1380     FWD(L)
1390     RIGHT(A)
1400     SPIRALS(L+1;A)
1410 END
1420 BEGIN
1430 TURTLE
1440 SPIRALS (9,95) ИЛИ (9,90) ИЛИ (9,121)...
```

ОШИБКИ В TURTLE

СТРОКА 570 USER(#EAC) ДОЛЖНА БЫТЬ ИЗМЕНЕНА:

```

INLINE (#FD, #21, #3A, #5C, #FD, #CB, 1, #CE, #CD, #AC, #E,
#FD, #CB, 1, #8E, #F3)
```


PASCAL 4T VERSION 5 (С 1 АПРЕЛЯ 1983 Г.)

1. ФУНКЦИИ МОГУТ ВОЗВРАЩАТЬ ЗНАЧЕНИЯ POINTER (УКАЗАТЕЛИ)
 2. ИСПРАВЛЕНА ОШИБКА В NEW
 3. НОВЫЙ РЕДАКТОР ДОБАВЛЕНА КОМАНДА V БЕЗ АРГУМЕНТОВ, КОТОРАЯ ВЫДАЕТ ЗНАЧЕНИЯ РАЗДЕЛИТЕЛЯ, LINE RANGE, FIND И SUBSTITUTE СТРОК
 - 4 РЕАЛИЗОВАНА ОПЦИЯ *F ('INCLUDE'). ЕСЛИ ВЫ ХОТИТЕ ЕЕ ИСПОЛЬЗОВАТЬ, ТО ИСХОДНЫЙ ТЕКСТ, КОТОРЫЙ ПРЕДПОЛАГАЕТСЯ 'ПОДКАЧИВАТЬ', ДОЛЖЕН БЫТЬ ВЫВЕДЕН НА ЛЕНТУ С ПОМОЩЬЮ КОМАНДЫ W. W РАБОТАЕТ АНАЛОГИЧНО КОМАНДЕ P, ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО ПО W ИНФОРМАЦИЯ ВЫВОДИТСЯ НЕ В HP4T ФОРМАТЕ И ЗАПИСЬ НАЧИНАЕТСЯ СРАЗУ ПОСЛЕ ОКОНЧАНИЯ НАБОРА КОМАНДЫ.
ТЕКСТ, ЗАПИСАННЫЙ ПО W, КОМАНДОЙ G,, СЧИТАТЬ НЕЛЬЗЯ.
- ПРИМЕР:
- ```
W 50,120,PLOT - ЗАПИСЬ ПРОЦЕДУРЫ PLOT.
ДЛЯ ВКЛЮЧЕНИЯ PLOT:
100 END
110
120 [*F PLOT] 'ВКЛЮЧЕНИЕ' ПРОЦЕДУРЫ PLOT
130
140 PROCEDURE MORE - ОСТАВШАЯСЯ ЧАСТЬ ПРОГРАММЫ.
```

HISOFT C COMPILER V1.1

COPYRIGHT C 1984 HISOFT

HISOFT C COMPILER V1.1  
COPYRIGHT C 1984 HISOFT

ИТАК, ПЕРВЫЙ ПРИМЕР;

```
MAIN()
[PRINTF ("PIERWSZY PROGRAM")
]
```

ПОСЛЕ НАБОРА ПРОГРАММЫ ВЫ ДОЛЖНЫ ОТМЕТИТЬ ЕЕ КОНЕЦ (EOF), НАЖАВ SS+I, ПОСЛЕ ЧЕГО ПОЯВИТСЯ СООБЩЕНИЕ:

TYPE Y TO RUN

К ЭТОМУ МОМЕНТУ ПРОГРАММА БУДЕТ СКОМПИЛИРОВАНА И ОТВЕТ Y ПРИВЕДЕТ К ВЫПОЛНЕНИЮ ПРОГРАММЫ!!!

ПРОИЗВЕДЯ ВСЕ ВЫШЕУКАЗАННЫЕ МАНИПУЛЯЦИИ, ВЫ, ВОЗМОЖНО, УВИДИТЕ НА ЭКРАНЕ:

```
PIERWSZY PROGRAM
TYPE Y TO RUN;
```

ЕСЛИ ВМЕСТО Y ВЫ НАЖМЕТЕ ЛЮБУЮ ДРУГУЮ КЛАВИШУ, ТО СИСТЕМА NISOFT ПЕРЕЙДЕТ В ИЗНАЧАЛЬНОЕ СОСТОЯНИЕ ВВОДА ТЕКСТА ПРОГРАММЫ.

ОПИСАННЫЙ СПОСОБ КОМПИЛЯЦИИ ИСПОЛЬЗУЕТСЯ ТОЛЬКО ДЛЯ КОМПИЛЯЦИИ ПРОСТЫХ ПРОГРАММ. ЗНАЧИТЕЛЬНАЯ ЖЕ ЧАСТЬ ПРОГРАММ ПОДГОТОВЛЯЕТСЯ СНАЧАЛА РЕДАКТОРОМ И ТОЛЬКО ПОСЛЕ ЭТОГО КОМПИЛИРУЕТСЯ.

ЕСЛИ СИСТЕМА NISOFT НАХОДИТСЯ В РЕЖИМЕ КОМПИЛИРОВАНИЯ, ТО ДЛЯ ВЫЗОВА РЕДАКТОРА НАЖМИТЕ EDIT (CS+I), А ЗАТЕМ ENTER. ПРИ ЭТОМ НА ЭКРАНЕ ПОЯВИТСЯ ">". КАЖДАЯ СТРОКА В РЕЖИМЕ РЕДАКТИРОВАНИЯ НУМЕРУЕТСЯ. ПОСЛЕДОВАТЕЛЬНОСТЬ ВВОДА НОМЕРОВ РОЛИ НЕ ИГРАЕТ РЕДАКТОР САМ РАСПОЛОЖИТ ИХ В ПОРЯДКЕ ВОЗРАСТАНИЯ.

ПРОЦЕСС ЗАМЕНЫ ИЛИ УДАЛЕНИЯ СТРОКИ С КАКИМ-ЛИБО НОМЕРОМ АНАЛОГИЧЕН ЗАМЕНЕ И УДАЛЕНИЮ В BASIC. ДЛЯ ПОЛУЧЕНИЯ ЛИСТИНГА НАБЕРИТЕ ДИРЕКТИВУ L.

ПОСЛЕ ТОГО, КАК ВЫ ОТРЕДАКТИРОВАЛИ ИСХОДНЫЙ ТЕКСТ, ЕГО МОЖНО СКОМПИЛИРОВАТЬ, НАБРАВ ДИРЕКТИВУ C. ПОСЛЕ ВЫЗОВА КОМПИЛЯТОРА НЕОБХОДИМО ИСПОЛЬЗОВАТЬ СПЕЦИАЛЬНУЮ ДИРЕКТИВУ #INCLUDE - ЕЕ ИНТЕРПРЕТАЦИЯ ПРИВЕДЕТ К ЕЕ ЗАМЕНЕ ИСХОДНЫМ ТЕКСТОМ КОМПИЛИРУЕМОЙ ПРОГРАММЫ И НЕПОСРЕДСТВЕННОЙ КОМПИЛЯЦИИ ПРОГРАММЫ. ЕСЛИ ПРОГРАММА НЕ ИМЕЕТ ОШИБОК, ТО ПОСЛЕ ЭТИХ МАНИПУЛЯЦИЙ НАЖМИТЕ SS+I (EOF)

ЕСЛИ ПОСЛЕ КОМПИЛЯЦИИ ВЫ ВЕРНУТЕСЬ В РЕДАКТОР И ВВЕДЕТЕ ДИРЕКТИВУ L, ТО ВЫ ОПЯТЬ УВИДИТЕ ИСХОДНЫЙ ТЕКСТ ПРОГРАММЫ, СЛЕДОВАТЕЛЬНО, ОН НЕ УДАЛЯЕТСЯ ПРИ КОМПИЛЯЦИИ. ДЛЯ НАБОРА НОВОЙ ПРОГРАММЫ НЕОБХОДИМО СТАРУЮ УДАЛИТЬ ДИРЕКТИВОЙ DM,N - ГДЕ M,N - СТРОКИ ПРОГРАММЫ.

МЕР;

```
200 MAIN ()
210 [
220 PRINTF(TRZECI PROGRAM")
230]
```

ПОСЛЕ ВВОДА ДИРЕКТИВ C И #INCLUDE КОМПИЛЯТОР ВЫДАСТ ВАМ СООБЩЕНИЕ: ERROR 37 - НЕОПРЕДЕЛЕННАЯ ПЕРЕМЕННАЯ, В ДАННОМ СЛУЧАЕ - ОТСУТСТВИЕ ПЕРВЫХ КАВЫЧЕК ". ЕСЛИ ПОСЛЕ ЭТОГО ВЫЗВАТЬ ДИРЕКТИВУ Q. ТО НА ЭКРАНЕ ВЫ ПОЛУЧИТЕ СТРОКУ С ОШИБКОЙ.

## 2. СТРУКТУРА ПРОГРАММ

СИСТЕМЕ HISOFT ЗАРЕЗЕРВИРОВАННЫ СЛЕДУЮЩИЕ КЛЮЧЕВЫЕ СЛОВА:

|          |        |          |          |
|----------|--------|----------|----------|
| AUTO     | ELSE   | LONG     | TYPDEF   |
| BREAK    | ENTRY  | REGISTER | UNION    |
| CASE     | EXTERN | RETURN   | UNSIGNED |
| CHARE    | FLOAT  | SHORT    | WHILE    |
| CONTINUE | FOR    | SIZEOF   |          |
| DEFAULT  | GOTO   | STATIC   |          |
| DO       | IF     | STRUCT   | FORTRAN  |
| DOUBLE   | INT    | SWITCH   | ASM      |

КРОМЕ ТОГО: INLINE И CAST.

ИДЕНТИФИКАТОРАМИ В С ЯВЛЯЮТСЯ ОТЛИЧНЫЕ ОТ КЛЮЧЕВЫХ СЛОВ ПОСЛЕДОВАТЕЛЬНОСТИ ЦИФР И БУКВ, НАЧИНАЮЩИЕСЯ С БУКВ. БОЛЬШИЕ И МАЛЫЕ БУКВЫ РАСПОЗНАЮТСЯ КАК РАЗНЫЕ ИДЕНТИФИКАТОРЫ. РАСПОЗНАВАЕМАЯ ДЛИНА ИДЕНТИФИКАТОРА РАВНА 8 ЗНАКАМ.

ЛИТЕРАЛАМИ НАЗЫВАЮТСЯ ЧИСЛОВЫЕ (ЧИСЛА), СИМВОЛЬНЫЕ (СИМВОЛЫ) И СТРОКОВЫЕ КОНСТАНТЫ. ЧИСЛОВОЙ ЛИТЕРАЛ СОСТОИТ ИЗ ПОСЛЕДОВАТЕЛЬНОСТИ ДЕСЯТИЧНЫХ ЦИФР. СИМВОЛЬНЫЙ ЛИТЕРАЛ ИМЕЕТ ВИД 'С', ГДЕ С ЯВЛЯЕТСЯ СИМВОЛОМ, ОТЛИЧНЫМ ОТ 'И', ИЛИ ОПИСАНИЕМ СИМВОЛА (Ø - ОПИСАНИЕ СИМВОЛА С КОДОМ РАВНЫМ НУЛЮ, N - ОПИСАНИЕ СИМВОЛА 'НОВАЯ СТРОКА', R - ОПИСАНИЕ СИМВОЛА 'ВОЗВРАТ КАРЕТКИ', T - ТАБУЛЯЦИЯ, ' - АПОСТРОФ, - НАКЛОННАЯ ЧЕРТА). СТРОЧНЫЙ ЛИТЕРАЛ ИМЕЕТ ВИД 'S', ГДЕ S - ЛЮБАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ ИЛИ ИХ ОПИСАНИЙ. СТРОКОВЫЙ ЛИТЕРАЛ ПРЕДСТАВЛЯЕТ СОБОЙ ССЫЛКУ НА ПЕРВЫЙ СИМВОЛ ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ В АПОСТРОФАХ, ДОПОЛНЕННУЮ СИМВОЛОМ С КОДОМ Ø В КАЧЕСТВЕ ОГРАНИЧИТЕЛЯ ЭТОЙ ПОСЛЕДОВАТЕЛЬНОСТИ.

В ОТЛИЧИЕ ОТ ДРУГИХ ЯЗЫКОВ, СИМВОЛЬНЫЙ ЛИТЕРАЛ НЕ ЯВЛЯЕТСЯ ЧАСТНЫМ СЛУЧАЕМ СТРОКОВОГО. ТАК 'J' ПРЕДСТАВЛЯЕТ СОБОЙ ОДНОСИМВОЛЬНОЕ ДАННОЕ, КОТОРОЕ ИМЕЕТ ВЕЛИЧИНУ, РАВНУЮ КОДУ БУКВЫ J, В ТО ВРЕМЯ КАК "J" ПРЕДСТАВЛЯЕТ СОБОЙ ДАННОЕ, УКАЗЫВАЮЩЕЕ НА ПЕРВЫЙ СИМВОЛ ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ, СОСТОЯЩЕЙ ИЗ ЗНАКА J И СИМВОЛА С КОДОМ Ø.

ТАКИМ ОБРАЗОМ ПРИМЕР:

```
INT AGE=44 /*ДЛЯ 1987 ГОДА*/
MAIN ()
[
 PRINTF("JAN %C IS %D " %S "'B", AGE, "NOW")
]
```

СОДЕРЖИТ КОМЕНТАРИИ /\*...\*/, КЛЮЧЕВОЕ СЛОВО INT, ИДЕНТИФИКАТОРЫ AGE, MAIN И PRINTF, СИМВОЛЬНЫЙ ЛИТЕРАЛ 'B', СТРОКОВЫЙ ЛИТЕРАЛ "NOW", ОПЕРАТОР = И Т.Д. В ИТОГЕ ПОЛУЧИМ JANB IS 44 "NOW".

## 3. ОПИСАНИЕ ПЕРЕМЕННЫХ, ФУНКЦИЙ И ТИПОВ

ДАННЫЕ МОГУТ БЫТЬ ПЕРЕМЕННЫМИ И КОНСТАНТАМИ И ПРЕДСТАВЛЯЮТСЯ СВОИМИ ИМЕНАМИ. ПРОСТЕЙШИМИ ИМЕНАМИ КОНСТАНТ ЯВЛЯЮТСЯ ЛИТЕРАЛЫ, А ПЕРЕМЕННЫХ - ИДЕНТИФИКАТОРЫ. В ОБЩЕМ СЛУЧАЕ ДАННЫЕ МОГУТ ПРЕДСТАВЛЯТЬСЯ ВЫРАЖЕНИЯМИ.

ТИП ДАННЫХ, ПРЕДСТАВЛЕННЫХ ЛИТЕРАЛАМИ, СЛЕДУЕТ ИЗ САМОЙ ЗАПИСИ ЛИТЕРАЛОВ, СЛЕДОВАТЕЛЬНО ОПИСАНИЕ ЛИТЕРАЛОВ ИЗЛИШНЕ. ПЕРЕ-



МЕННЫЕ ЖЕ ДОЛЖНЫ БЫТЬ ОБЯЗАТЕЛЬНО ОПИСАНЫ ПЕРЕД ИХ ИСПОЛЬЗОВАНИЕМ В ПРОГРАММЕ.

ОПИСАНИЕ (ДЕКЛАРАЦИЯ) ИМЕЕТ ВИД:  
ПРЕДЕЛЕНИЕ-ТИПА СПИСОК-ДЕКЛАРАТОРОВ

К ПРИМЕРУ INT A,B,C. СПИСОК ДЕКЛАРАТОРОВ МОЖЕТ ИМЕТЬ И БОЛЕЕ СЛОЖНЫЙ ВИД INT\*((\*FUN())[2])().

ПРАВИЛА КОНСТРУИРОВАНИЯ ДЕКЛАРАТОРОВ СЛЕДУЮЩИЕ: ЕСЛИ НЕКОТОРАЯ ЗАПИСЬ DEC ЯВЛЯЕТСЯ ДЕКЛАРАТОРОМ, ТО ДЕКЛАРАТОРАМИ ТАКЖЕ ЯВЛЯЮТСЯ ЗАПИСИ: (DEC), ОЗНАЧАЮЩАЯ ТО ЖЕ САМОЕ, ЧТО И DEC[W], ОЗНАЧАЮЩАЯ МАССИВ ИЗ W ЭЛЕМЕНТОВ \*DEC, ОЗНАЧАЮЩАЯ УКАЗАТЕЛЬ НА ОБЪЕКТ И DEC(), ОЗНАЧАЮЩАЯ ФУНКЦИЮ. НАИВЫСШИМ ПРИОРИТЕТОМ ОБЛАДАЮТ КРУГЛЫЕ СКОБКИ, ОТДЕЛЯЮЩИЕ САМ ДЕКЛАРАТОР, А НАИНИЗШИМ - ЗНАК \*.

ПРИМЕР:

INT (\*NAME())[2] РАВНОСИЛЬНО INT(\*(NAME()))[2]  
- ДЕКЛАРАЦИЯ ФУНКЦИИ NAME, РЕЗУЛЬТАТОМ КОТОРОЙ ЯВЛЯЕТСЯ УКАЗАТЕЛЬ ДВУХЭЛЕМЕНТНОГО МАССИВА ДАННЫХ ТИПА (INT). ОШИБОЧНОЙ ДЕКЛАРАЦИЕЙ ЯВЛЯЕТСЯ: CHAR FUN()[3], ЭКВИВАЛЕНТНАЯ CHAR (FUN())[3], Т.К. ОНА ОБЪЯВЛЯЕТ ФУНКЦИЮ, РЕЗУЛЬТАТОМ КОТОРОЙ ЯВЛЯЕТСЯ ТР ЭЛЕМЕНТНЫЙ МАССИВ ДАННЫХ ТИПА (CHAR), А РЕЗУЛЬТАТОМ ФУНКЦИИ МОЖЕТ БЫТЬ ТОЛЬКО СКАЛЯРНАЯ ВЕЛИЧИНА.

ОСТАЕТСЯ ОТМЕТИТЬ, ЧТО ЕСЛИ НЕКОТОРЫЙ ОБЪЕКТ ПРОГРАММЫ ОБЪЯВЛЕН С ПОМОЩЬЮ ДЕКЛАРАЦИИ;

ОПРЕДЕЛЕНИЕ-ТИПА ДЕКЛАРАТОР,  
ТО ОН ПРИНИМАЕТСЯ РАВНЫМ ОПРЕДЕЛЕНИЮ:

(ОПРЕДЕЛЕНИЕ-ТИПА ПСЕВДОДЕКЛАРАТОР),  
В КОТОРОМ ДЕКЛАРАТОР ОТЛИЧАЕТСЯ ОТ ПСЕВДОДЕКЛАРАТОРА ТЕМ, ЧТО ПОСЛЕДНИЙ НЕ СОДЕРЖИТ ИДЕНТИФИКАТОРА ОБЪЕКТА.

ПРИМЕР: ПЕРЕМЕННАЯ PTR, ОБЪЯВЛЕННАЯ КАК INT(\*PTR)[2][3],  
ИМЕЕТ ТИП (INT(\*)[2][3])

# -10Б ЯВЛЕНИЕ ПРОСТЫХ ПЕРЕМЕННЫХ -0

# ПЕРЕМЕННЫЕ ТИПА (CHAR) МОГУТ ПРИНИМАТЬ ЗНАЧЕНИЯ 0...255  
# ПЕРЕМЕННЫЕ ТИПА (INT) МОГУТ ПРИНИМАТЬ ЗНАЧЕНИЯ -32768...32767  
# ПЕРЕМЕННЫЕ ТИПА (UNSIGNED) МОГУТ ПРИНИМАТЬ ЗНАЧЕНИЯ 0...65535  
# УКАЗЫВАЮЩИЕ ПЕРЕМЕННЫЕ ТИПА (CHAR) МОГУТ ПРИНИМАТЬ ЗНАЧЕНИЯ  
УКАЗАТЕЛЯ НА ПЕРЕМЕННУЮ.

ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ МОЖНО СОВМЕСТИТЬ С ПРИСВОЕНИЕМ ИМ НАЧАЛЬНОГО ЗНАЧЕНИЯ С ТЕМ УСЛОВИЕМ, ЧТО ЭТО ЗНАЧЕНИЕ МОЖЕТ БЫТЬ ВЫЧИСЛЕННО ЕЩЕ ПЕРЕД ВЫПОЛНЕНИЕМ ПРОГРАММЫ.

-10WASA -0; В ДАННОЙ РЕАЛИЗАЦИИ ЯЗЫКА С ПРИСВАИВАНИЕ НАЧАЛЬНЫХ ДАННЫХ ДОПУСКАЕТСЯ ТОЛЬКО ВО ВНЕШНИХ ПО ОТНОШЕНИЮ К ОПРЕДЕЛЕНИЮ ФУНКЦИИ ДЕКЛАРАЦИЯХ:

```
CHAR INITIAL = 'J',
 NAME = 'B'
INT AGE = 44
```

```
MAIN()
```

```
{
 INT YEAR
 YEAR = 1987
 PRINTF("%C%C IS %D IN %D", INITIAL, NAME, AGE, YEAR)
}
```

ОПИСАНИЕ И ПРИСВОЕНИЕ ЗНАЧЕНИЯ ПЕРЕМЕННОЙ YEAR РЕАЛИЗОВАНЫ РАЗНЫМИ ОПЕРАТОРАМИ (Т.К. ОНА ОПРЕДЕЛЯЕТСЯ ВНУТРИ ФУНКЦИИ), А ПЕРЕМЕННОЙ AGE — ОДНИМ.

ЕСЛИ ПЕРЕМЕННОЙ НАЧАЛЬНЫЕ ДАННЫЕ НЕ ПРИСВАИВАЮТСЯ, ОНА ПРИНИМАЕТ ЗНАЧЕНИЕ 0.

РАССМОТРИМ УКАЗЫВАЮЩИЕ ПЕРЕМЕННЫЕ. ИМИ ЯВЛЯЮТСЯ ДАННЫЕ, ОПРЕДЕЛЯЮЩИЕ ПОЛОЖЕНИЕ ДРУГИХ ДАННЫХ. ДЛЯ ПРЕОБРАЗОВАНИЯ ВЫРАЖЕНИЙ ПРЕДСТАВЛЯЮЩИХ ОПРЕДЕЛЕННЫЕ ДАННЫЕ, В ВЫРАЖЕНИЯ, ПРЕДСТАВЛЯЮЩИЕ УКАЗАТЕЛИ НА ЭТИ ДАННЫЕ, ИСПОЛЬЗУЕТСЯ &. ОБРАТНЫМ ОПЕРАТОРОМ ЯВЛЯЕТСЯ \*.

К ПРИМЕРУ, ТАК КАК СТРОКОВЫЙ ЛИТЕРАЛ ПРЕДСТАВЛЯЕТ ДАННОЕ, УКАЗЫВАЮЩЕЕ НА ПЕРВЫЙ ЗНАК ПОСЛЕДОВАТЕЛЬНОСТИ, ОПРЕДЕЛЕННОЙ ЭТИМ ЛИТЕРАЛОМ (ЛИТЕРАЛ "JB" ПРЕДСТАВЛЯЕТ ДАННОЕ, УКАЗЫВАЮЩЕЕ НА ЗНАК J), ТО ВЫРАЖЕНИЕ "&"JB" ПРЕДСТАВЛЯЕТ ЗНАК J ПОСЛЕДОВАТЕЛЬНОСТИ, СОСТОЯЩЕЙ ИЗ ЗНАКОВ J, B И ЗНАКА С КОДОМ НОЛЬ.

В С В ВЫРАЖЕНИЯХ С УКАЗАТЕЛЯМИ ДОПУСКАЮТСЯ ОПЕРАЦИИ СЛОЖЕНИЯ И ВЫЧИТАНИЯ. ТАК, ЕСЛИ PTR ЕСТЬ ВЫРАЖЕНИЕ, УКАЗЫВАЮЩЕЕ НА I-ЫЙ ЭЛЕМЕНТ ПОСЛЕДОВАТЕЛЬНОСТИ ДАННЫХ, ТО PTR + NUM, ГДЕ NUM — ЦЕЛОЕ, УКАЗЫВАЕТ НА I + NUM ЭЛЕМЕНТ ЭТОЙ ПОСЛЕДОВАТЕЛЬНОСТИ.

\*(PTR + NUM) ЭКВИВАЛЕНТНО PTR[NUM] ИЛИ NUM[PTR].

ПАРУ СКОБОК [ ] В ДАЛЬНЕЙШЕМ БУДЕМ НАЗЫВАТЬ ОПЕРАТОРОМ ИНДЕКСИРОВАНИЯ.

ПРИМЕР:

```
CHAR *REF="JAN"
MAIN()
[
 PRINTF("%S=%C%C%C", REF,*REF,*(&REF+1),REF[2])
]
```

В ЭТОЙ ПРОГРАММЕ REF ОБ ЯВЛЕНА КАК ПЕРЕМЕННАЯ, КОТОРОЙ МОЖНО ПОСТАВИТЬ В СООТВЕТСТВИЕ УКАЗАТЕЛЬ ДАННЫХ ТИПА (CHAR). Т.О., ПЕРЕМЕННАЯ, ПРЕДСТАВЛЕННАЯ В ПРОГРАММЕ ИДЕНТИФИКАТОРОМ REF, ИМЕЕТ ТИП (CHAR \*). ПРИ ОБ ЯВЛЕНИИ ПЕРЕМЕННОЙ REF ПРИСВОЕНО ЗНАЧЕНИЕ, УКАЗЫВАЮЩЕЕ НА БУКВУ J 4-ХЗНАКОВОЙ ПОСЛЕДОВАТЕЛЬНОСТИ J,A,N И ЗНАК С КОДОМ НОЛЬ. В ПРОЦЕССЕ ВЫПОЛНЕНИЯ КОМАНДЫ, В КОТОРОЙ ВЫЗЫВАЕТСЯ ФУНКЦИЯ PRINTF, ПРОИСХОДИТ СВЯЗЫВАНИЕ ШАБЛОНА %S С УКАЗАТЕЛЕМ, ПРИСВОЕННЫМ ПЕРЕМЕННОЙ REF. Т.О. НА ЭКРАН ВЫВОДИТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ЗНАКОВ, НАЧИНАЯ С УКАЗАННОГО МЕСТА, ДО ЗНАКА С КОДОМ НОЛЬ, ИСКЛЮЧАЯ ПОСЛЕДНИЙ, Т.Е. СЛОВА JAN. ЕСЛИ REF ПРЕДСТАВЛЯЕТ УКАЗАТЕЛЬ НА ЛИТЕРУ J, ТО \*REF ПРЕДСТАВЛЯЕТ САМУ БУКВУ J СЛЕДОВАТЕЛЬНО. СВЯЗЫВАНИЕ ПЕРВОГО ШАБЛОНА % С АРГУМЕНТОМ \*REF ПРИВЕДЕТ К ПЕЧАТИ БУКВЫ J. REF + 1 ПРЕДСТАВЛЯЕТ УКАЗАТЕЛЬ НА БУКВУ A, А \*(&REF + 1) — БУКВУ A. ВЫРАЖЕНИЕ REF[2] ЯВЛЯЕТСЯ УПРОЩЕННЫМ ПРЕДСТАВЛЕНИЕМ \*(&REF + 2), А, СЛЕДОВАТЕЛЬНО, ПРЕДСТАВЛЯЕТ БУКВУ N. В ИТОГЕ ПОЛУЧИМ СЛОВО JAN = JAN.

ТАКИМ ОБРАЗОМ:

```
INT NUM — ДЕКЛАРАЦИЯ ПЕРЕМЕННОЙ ТИПА (INT)
INT *PTR — (INT *)
INT **REF — (INT **)
```

-10Б ЯВЛЕНИЕ МАССИВОВ -2

МАССИВ — ПЕРЕМЕННАЯ СЛОЖНОГО ТИПА, ВСЕ ЭЛЕМЕНТЫ КОТОРОЙ ИМЕЮТ ТОТ ЖЕ ТИП, ЧТО И САМ МАССИВ. МОЖНО ОБ ЯВЛЯТЬ ОДНОМЕРНЫЕ И МНОГОМЕРНЫЕ МАССИВЫ. МАССИВАМ, ОБ ЯВЛЕННЫМ ДО ОПРЕДЕЛЕНИЯ ФУНКЦИИ, МОЖНО ПРИСВАИВАТЬ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ. ЗАПИСЬ, ОПРЕДЕЛЯЮЩАЯ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ, НАЗЫВАЕТСЯ ИНИЦИАТОРОМ И СОСТОИТ ИЗ СПИСКА ЛИТЕРАЛОВ В [ ]. ПО УМОЛЧАНИЮ ЭЛЕМЕНТАМ МАССИВА ПРИСВАИВАЕТСЯ 0.

```
CHAR SOURCE [3][3] = [['E', 'W', 'A'], ['I', 'Z', 'A'],
 ['J', 'A', 'N']]
MAIN()
[
 CHAR TARGET [4] /* ОБЯВЛЕНИЕ ВТОРОГО МАССИВА */
 INT I
 FOR (I=0 I<4 I++)
 TARGET[I]=SOURCE[2][I]
 TARGET[3]='0'
 PRINTF ("%S", TARGET)
]
```

Т.О. TARGET [0]... TARGET [3] = J A N КОД 0  
В ЯЗЫКЕ ПРИНЯТО, ЧТО КАЖДОЕ ВЫРАЖЕНИЕ, ПРЕДСТАВЛЯЮЩЕЕ МАССИВ, СРАЗУ ЖЕ НЕЯВНО ПРЕОБРАЗУЕТСЯ В ВЫРАЖЕНИЕ, ПРЕДСТАВЛЯЮЩЕЕ ПЕРВЫЙ ЭЛЕМЕНТ МАССИВА, СЛЕДОВАТЕЛЬНО:

PRINTF("%S", TARGET) ЭКВИВАЛЕНТНО PRINTF("%S", &TARGET[0])  
ТАКОЙ СПОСОБ ТРАКТОВКИ ИМЕН МАССИВОВ, КАК ИМЕН УКАЗАТЕЛЕЙ, ПРИВОДИТ К ТОМУ, ЧТО ВЫПОЛНЕНИЕ ОПЕРАЦИИ НАД МАССИВАМИ МОЖЕТ БЫТЬ В ОБЩЕМ СЛУЧАЕ ЗАМЕНЕНО ВЫПОЛНЕНИЕМ ОПЕРАЦИИ НАД УКАЗЫВАЮЩИМИ ДАННЫМИ.

КРОМЕ ТОГО, ЭЛЕМЕНТАМИ МАССИВОВ МОГУТ БЫТЬ СТРУКТУРЫ, УНИИ И УКАЗЫВАЮЩИЕ ПЕРЕМЕННЫЕ;

```
CHAR *FAMILY[3] = ["EWA", "IZA", "JAN"]
MAIN ()
[
 PRINTF("%C&%C", **FAMILY, *FAMILY[2])
]
```

ЗДЕСЬ ЭЛЕМЕНТАМИ МАССИВА ЯВЛЯЮТСЯ УКАЗАТЕЛИ ДАННЫХ ТИПА (CHAR). ПРИ ОПРЕДЕЛЕНИИ МАССИВА, ЕГО ЭЛЕМЕНТУ FAMILY[0] ПРИСВОЕНО УКАЗАНИЕ НА ПЕРВЫЙ ЗНАК ПОСЛЕДОВАТЕЛЬНОСТИ EWA, АНАЛОГИЧНО, FAMILY[1] - IZA, А FAMILY[2] - JAN. Т.К. ВЫРАЖЕНИЕ FAMILY ПРЕДСТАВЛЯЕТ СОБОЙ УКАЗАТЕЛЬ НА FAMILY[0], ВЫРАЖЕНИЕ \*FAMILY ПРЕДСТАВЛЯЕТ ЭЛЕМЕНТ FAMILY[0], А, СЛЕДОВАТЕЛЬНО, \*\*FAMILY ПРЕДСТАВЛЯЕТ СОБОЙ ОБЪЕКТ, УКАЗАННЫЙ ВЫРАЖЕНИЕМ FAMILY[0], Т.Е. БУКВУ Е. ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРИВОДИТ К ПЕЧАТИ E & J.

МОЖНО СКАЗАТЬ, ЧТО ЕСЛИ САМ МАССИВ FAMILY ИМЕЕТ ТИП (CHAR \*[3]) (3-Х ЭЛЕМЕНТАРНЫЙ МАССИВ УКАЗАТЕЛЕЙ ДАННЫХ ТИПА (CHAR)), ТО ВЫРАЖЕНИЕ FAMILY ИМЕЕТ ТИП (CHAR \*), ВЫРАЖЕНИЕ \*FAMILY - (CHAR\*) А ВЫРАЖЕНИЕ \*\*FAMILY - (CHAR).

В ОТЛИЧИЕ ОТ ПРЕДЫДУЩЕГО, В ДАННОМ ПРИМЕРЕ ИСПОЛЬЗУЕТСЯ ПЕРЕМЕННАЯ, УКАЗЫВАЮЩАЯ НА МАССИВ:

```
INT ARRAY[2][2] = [[3,2], [1,0]],
 (*PTR)[2] = ARRAY + 1
MAIN ()
[
 PRINTF ("%D%D", **PTR, PTR[-1][0])
]
```

ЗДЕСЬ ARRAY - 2-Х ЭЛЕМЕНТНЫЙ МАССИВ, ЭЛЕМЕНТАМИ КОТОРОГО ЯВЛЯЮТСЯ 2-Х ЭЛЕМЕНТНЫЕ МАССИВЫ С ЭЛЕМЕНТАМИ ТИПА INT. МАССИВ ARRAY МОЖНО РАССМАТРИВАТЬ КАК ДВУМЕРНЫЙ МАССИВ С ЭЛЕМЕНТАМИ 3,2,1,0. В ТО ЖЕ ВРЕМЯ PTR ЯВЛЯЕТСЯ ПРОСТОЙ ПЕРЕМЕННОЙ, УКАЗЫВАЮЩЕЙ НА 2-Х ЭЛЕМЕНТНЫЙ МАССИВ ТИПА INT. ЕЙ ПРИПИСАНО ЗНАЧЕНИЕ ARRAY+1 И ОНА ИМЕЕТ ТИП (INT \*[2]). ТАКОЕ ПРИСВОЕНИЕ ПРАВОМОЧНО, Т.К. PTR И ARRAY+1 ИМЕЮТ ОДИНАКОВЫЙ ТИП (INT \*[2]).

-10.3 ЯВЛЕНИЕ СТРУКТУР И УНИИ -0

СТРУКТУРАМИ И УНИИ ЯВЛЯЮТСЯ СЛОЖНЫЕ ПЕРЕМЕННЫЕ, СОСТОЯЩИЕ ИЗ КОМПОНЕНТОВ РАЗНОГО ТИПА. ЭТИ КОМПОНЕНТЫ НАЗЫВАЮТСЯ ПОЛЯМИ, ПОЛЯМИ МОГУТ БЫТЬ НЕ ТОЛЬКО ПРОСТЫЕ ПЕРЕМЕННЫЕ, НО И МАССИВЫ, СТРУКТУРЫ И УНИИ.

ПОЛЯ СТРУКТУР РАЗМЕЩАЮТСЯ В ПАМЯТИ В ПОРЯДКЕ ИХ ОБЯВЛЕНИЯ, А КАЖДОЕ ПОЛЕ УНИИ РАЗМЕЩАЕТСЯ НАЧАНАЯ ОТ ОДНОГО И ТОГО ЖЕ МЕСТА. СЛЕДОВАТЕЛЬНО, ЕСЛИ В ПРОГРАММЕ ОБЯВЛЕНА СТРУКТУРА RECORD:

```
STRUCT[
 INT FIX
 CHAR CHR
 INT ARR[3]
 CHAR *PTR[2]
] RECORD
```

С ПОЛЯМИ FIX, CHR, ARR И PTR, ТО ОНА ЗАЙМЕТ СТОЛЬКО МЕСТА, СКОЛЬКО ЗАНИМАЮТ ВСЕ ПОЛЯ В СУММЕ. ТОГДА КАК, ЕСЛИ ОБЯВЛЕНА УНИЯ:

```
UNION [
 INT FIX
 CHAR CHR
 INT ARR[3]
 CHAR *PTR[2]
] RECORD
```

ТО ОНА ЗАЙМЕТ СТОЛЬКО МЕСТА, СКОЛЬКО ЗАНИМАЕТ САМОЕ ДЛИННОЕ ПОЛЕ УЧИТЫВАЯ, ЧТО В HISOFT ПЕРЕМЕННАЯ ТИПА (CHAR) ЗАНИМАЕТ ОДИН БАЙТ А ТИПА (INT) И УКАЗЫВАНИЕ — ПО 2 БАЙТА, ТО СТРУКТУРА ЗАЙМЕТ 11 БАЙТ, А УНИЯ — 6 БАЙТ ОЗУ, Т.О. РАЗЛИЧИЯ МЕЖДУ УНИЯМИ И СТРУКТУРАМИ

ЗАКЛЮЧАЮТСЯ ТОЛЬКО В СПОСОБЕ РАЗМЕЩЕНИЯ В ОЗУ И В ТОМ, ЧТО УНИЯМ НЕЛЬЗЯ ПРИПИСАТЬ НАЧАЛЬНОЕ ЗНАЧЕНИЕ.

ОБЯВЛЕНИЕ СТРУКТУРЫ СОСТОИТ ИЗ КЛЮЧЕВОГО СЛОВА STRUCT ПОСЛЕ КОТОРОГО В [ ] СТОИТ ПЕРЕЧЕНЬ ПОЛЕЙ СТРУКТУРЫ, А ПОСЛЕ НЕГО — ДЕКЛАРАТОР И ЗНАК . ДЕКЛАРАТОР МОЖЕТ СОСТОЯТЬ ИЗ ИДЕНТИФИКАТОРА СТРУКТУРЫ, А МОЖЕТ ИМЕТЬ И БОЛЕЕ СЛОЖЕННЫЙ ВИД:

```
STRUCT [
 INT FIX
 CHAR CHR[2][2]
] STR, ARR[3], *PTR[4]
```

— ДЕКЛАРАЦИЯ СТРУКТУРЫ STR С ПОЛЯМИ FIX И CHR, ДЕКЛАРАЦИЯ 3X-ЭЛЕМЕНТНОЙ ТАБЛИЦЫ СТРУКТУР ARR, А ТАК ЖЕ ДЕКЛАРАЦИЯ 4X-ЭЛЕМЕНТНОЙ ТАБЛИЦЫ УКАЗАТЕЛЕЙ СТРУКТУР ТАКИХ КАК STR И ARR[1].

ЕСЛИ МЕЖДУ КЛЮЧЕВЫМ СЛОВОМ STRUCT И ПЕРВОЙ СКОБКОЙ СТОИТ ИДЕНТИФИКАТОР, ТО В ДАЛЬНЕЙШИХ ОБЯВЛЕНИЯХ ОН МОЖЕТ БЫТЬ ИСПОЛЬЗОВАН ДЛЯ ИДЕНТИФИЦИРОВАНИЯ СТРУКТУРЫ ДАННОГО ВИДА. В ЧАСТНОСТИ, ЭТО ОЗНАЧАЕТ, ЧТО ТАКОЕ ОБЯВЛЕНИЕ, КАК

```
STRUCT TAG [
 INT FIX
 CHAR CHR
] ONE, TWO [2]
```

В КОТОРОЙ TAG ЯВЛЯЕТСЯ ИДЕНТИФИКАТОРОМ, В ДАЛЬНЕЙШЕМ НАЗЫВАЕМЫМ ОПИСАТЕЛЕМ СТРУКТУРЫ, МОЖЕТ С РАВНЫМ УСПЕХОМ БЫТЬ ЗАМЕНЕНО ОБЯ-

ВЛЕНИЕМ;

```
STRUCT TAG [
 INT FIX
 CHAR CHR
] ONE
STRUCT TAG TWO[2]
```

ИЛИ ПАРЫ ОБ ЯВЛЕНИИ:

```
STRUCT TAG [
 INT FIX
 CHAR CHR
]
STRUCT TAG ONE, TWO[2]
```

ИСПОЛЬЗОВАНИЕ ОПИСАТЕЛЯ СТРУКТУРЫ ЦЕЛЕСООБРАНО В ТЕХ СЛУЧА-  
ЯХ, КОГДА В ПРОЦЕССЕ ОБ ЯВЛЕНИЯ ПОЛЯ СТРУКТУРЫ НЕОБХОДИМО ОБРА-  
ЩЕНИЕ К ТИПУ УЖЕ ОБ ЯВЛЕННОЙ СТРУКТУРЫ:

```
STRUCT LIST [
 INT DATA
 STRUCT LIST *LINK
] STR
```

ГДЕ ПОЛЮ LINK МОЖЕТ БЫТЬ ПРИСВОЕНЫ УКАЗАТЕЛИ СТРУКТУР ТИПА (STR-  
UCT LIST).

ПОЛЯМ СТРУКТУР (НО НЕ УНИИ) МОГУТ БЫТЬ ПРИСВОЕНЫ ИСХОДНЫЕ  
ДАННЫЕ. ИНИЦИАТОР СТРУКТУРЫ СОСТОИТ, В ОБЩЕМ СЛУЧАЕ, ИЗ СПИСКА В  
[ ], ЭЛЕМЕНТАМИ СПИСКА МОГУТ БЫТЬ КОНСТАНТЫ, ЧАЩЕ ВСЕГО ЛИТЕРАЛЫ  
ИЛИ ПОДСПИСКИ КОНСТАНТЫ ДЛЯ ИНИЦИАЛИЗАЦИИ СКАЛЯРНЫХ ПОЛЕЙ, А  
ПОДСПИСКИ ДЛЯ ИНИЦИАЛИЗАЦИИ ПОЛЕЙ, КОТОРЫЕ ЯВЛЯЮТСЯ МАССИВАМИ И  
СТРУКТУРАМИ. ПОДСПИСОК, ИСПОЛЬЗУЕМЫЙ ДЛЯ ИНИЦИАЛИЗАЦИИ ПОЛЯ, ИМ-  
ЕЕТ ВИД ИДЕНТИФИКАТОРА ОБ ЕКТА С ТИПОМ ЭТОГО ПОЛЯ. ЕСЛИ НЕКОТО-  
РЫЙ ПОДСПИСОК СОДЕРЖИТ ПОЛНЫЙ КОМПЛЕКТ КОНСТАНТНЫХ ВЫРАЖЕНИЙ,  
ИНИЦИИРУЮЩИХ ДАННОЕ ПОЛЕ, ТО [ ] МОГУТ БЫТЬ ОПУЩЕНЫ. ЕСЛИ ДЛЯ ОП-  
РЕДЕЛЕННЫХ ПОЛЕЙ НЕ ОПРЕДЕЛЕНЫ ЯВНО НАЧАЛЬНЫЕ ДАННЫЕ, А ОБ ЯВЛЕ-  
НИЕ СТРУКТУРЫ НАХОДИТСЯ ВНЕ ОПРЕДЕЛЕНИЯ ФУНКЦИИ, ТО ТАКИМ ПОЛЯМ  
ПРИСВАИВАЕТСЯ ЛИТЕРАЛ НОЛЬ. ПОЛЯМ СТРУКТУРЫ, ОБ ЯВЛЕННОЙ ВНУТРИ  
ФУНКЦИИ, НЕ МОГУТ БЫТЬ ПРИСВОЕНЫ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ.

```
STRUCT [
 CHAR ONE, TWO
 STRUCT [
 CHAR UNO, DUE
] INSTR
 CHAR ARR[2],
 *REF
] OUTSTR=[1,2,[3,4],[5]]
```

ПОЛЯМ СТРУКТУРЫ OUTSTR ПРИСВОЕНЫ ДАННЫЕ ТАКИМ ОБРАЗОМ, ЧТО ПОЛЮ  
ARR[0] БУДЕТ ПРИСВОЕНО 5, А ПОЛЮ REF - 0.

-1UWAGA -0; В HISOFT ДЛЯ УПРОЩЕНИЯ КОМПИЛЯТОРА ПРИНЯТО,  
ЧТО ИНИЦИАТОР СТРУКТУРЫ НЕ МОЖЕТ СОДЕРЖАТЬ ПОДСПИСОК. ИЗ ЭТОГО  
СЛЕДУЕТ, ЧТО В ПРИВЕДЕННОЙ ВЫШЕ ДЕКЛАРАЦИИ ИНИЦИАТОР ДОЛЖЕН БЫТЬ  
ИЗМЕНЕН НА СЛЕДУЮЩИЙ: [1,2,3,4,5]. ДРУГИМ ОТСТУПЛЕНИЕМ ЯВЛЯЕТСЯ  
ТО, ЧТО ПРИСВОЕНИЕ ДАННЫХ ОТНОСИТСЯ НЕ К ОТДЕЛЬНЫМ ПОЛЯМ СТРУКТУ-  
РЫ, А К ОБЛАСТИ ПАМЯТИ, ВЫДЕЛЕННОЙ СТРУКТУРЕ. УЧИТЫВАЯ, ЧТО ЭЛЕ-  
МЕНТОМ СПИСКА ИНИЦИАТОРА ЯВЛЯЕТСЯ ВЫРАЖЕНИЕ СО ЗНАЧЕНИЕМ 0 - 255



ТО ИНИЦИИРУЕТСЯ ОДИН ИЛИ ДВА БАЙТА ОБЛАСТИ ПАМЯТИ, ПРИСВОЕННОЙ СТРУКТУРЕ:

```
STRUCT [
 INT UNO, DUE
] STR = [1,1]
```

ПРИСВОЕНИЕ ПОЛЮ UNO ДАННОГО, ВЕЛИЧИНОЙ 257, А ПОЛЮ DUE - 0, ВМЕСТО 1 ДЛЯ ОБОИХ ПОЛЕЙ.

#### -106 ЯВЛЕНИЕ ФУНКЦИИ -0

ОБ ЯВЛЕНИЕ ФУНКЦИИ СОСТОИТ ИЗ ЗАГОЛОВКА И ТЕЛА ФУНКЦИИ. ЗАГОЛОВОК СОДЕРЖИТ ИМЯ ФУНКЦИИ, В КРУГЛЫХ СКОБКАХ СПИСОК ПАРАМЕТРОВ ФУНКЦИИ И ОБ ЯВЛЕНИЕ ПАРАМЕТРОВ. ТЕЛО ФУНКЦИИ ЯВЛЯЕТСЯ ГРУППОЙ КОМАНД, НАЧИНАЮЩЕЙСЯ [ И КОНЧАЮЩЕЙСЯ ]. ЕСЛИ ВЫПОЛНЕНИЕ ФУНКЦИИ ЗАКАНЧИВАЕТСЯ КОМАНДОЙ:

RETURN EXP

ГДЕ EXP - ВЫРАЖЕНИЕ, ТО В МЕСТЕ ВЫЗОВА ФУНКЦИИ СТАНУТ ДОСТУПНЫМИ ДАННЫЕ EXP. ТИП ДАННЫХ ОПРЕДЕЛЯЕТСЯ НА ОСНОВЕ АНАЛИЗА ЗАГОЛОВКА ФУНКЦИИ. В СЛЕДУЮЩЕМ ПРИМЕРЕ НА ЭКРАНЕ ПЕЧАТАЕТСЯ BIELECKI:

```
MAIN ()
[
 CHAR *SURNAME ()
 PRINTF ("%S", SURNAME ("JAN BIELECKI",4))
]

CHAR *
SURNAME (STRING, NUMBER)
 CHAR *STRING
 INT NUMBER
[RETURN STRING + NUMBER]
```

ЗДЕСЬ ОПРЕДЕЛЯЮТСЯ ДВЕ ФУНКЦИИ: ФУНКЦИЯ MAIN() БЕЗ ПАРАМЕТРОВ И ФУНКЦИЯ SURNAME С ДВУМЯ ПАРАМЕТРАМИ. ПАРАМЕТР STRING ТИПА (CHAR\*) СВЯЗАН С АРГУМЕНТОМ "JAN BIELECKI", А ПАРАМЕТР NUMBER ТИПА (INT) СВЯЗАН С АРГУМЕНТОМ 4. РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ ДАННОЕ, ВЫСТУПАЮЩЕЕ В КОМАНДЕ RETURN. ЭТО ДАННОЕ ИМЕЕТ ТИП (CHAR\*) В СООТВЕТСТВИИ С ОБ ЯВЛЕННЫМ В ЗАГОЛОВКЕ ФУНКЦИИ ТИПОМ. ТАК КАК ВЫЗОВ ФУНКЦИИ ПРОИСХОДИТ ПЕРЕД ЕЕ ОПРЕДЕЛЕНИЕМ, В ФУНКЦИИ MAIN ПРОИЗВОДИТСЯ ПРЕДВАРИТЕЛЬНАЯ ДЕКЛАРАЦИЯ. ЭТА ДЕКЛАРАЦИЯ ИМЕЕТ ВИД ЗАГОЛОВКА ОПРЕДЕЛЕНИЯ ИЗ КОТОРОГО ИЗЯТ СПИСОК ИМЕН ПАРАМЕТРОВ И ИХ ОБ ЯВЛЕНИЕ.

ЕСЛИ ВЫПОЛНЕНИЕ ФУНКЦИИ НЕ ЗАКАНЧИВАЕТСЯ КОМАНДОЙ RETURN, СОДЕРЖАЩЕЙ ВЫРАЖЕНИЕ, ТО ОПРЕДЕЛЕНИЕ ТИПА РЕЗУЛЬТАТА ИЗЛИШНЕ. ВСЕ МОЖНО ОПУСКАТЬ ОБ ЯВЛЕНИЕ ТИПА РЕЗУЛЬТАТА ИЛИ ПАРАМЕТРОВ ФУНКЦИИ, ЕСЛИ ОНИ ИМЕЮТ ТИП (INT).

```
INT _13=-13
INT
MAIN ()
[
 INT NEGATE ()
 PRINTF ("%D", NEGATE(_13))
]
INT
```

```

NEGATE (PAR)
 INT PAR
 [RETURN -PAR]
- УПРОЩАЕТСЯ ДО:
INT _13 = -13
MAIN ()
[
 PRINTF ("%D", NEGATE (_13))
]
NEGATE (PAR)
[
 RETURN -PAR
].

```

В СЛЕДУЮЩЕЙ ПРОГРАММЕ ОПРЕДЕЛЯЮТСЯ ФУНКЦИИ FIRST И SECOND С ПРЕДВАРИТЕЛЬНОЙ ДЕКЛАРАЦИЕЙ ЭТИХ ФУНКЦИЙ.

```

CHAR NAME [2] = "BJ"
EXTERN CHAR *FIRST () /* FIRST */
MAIN ()
[
 PRINTF ("%C", *FIRST())
]
CHAR *
FIRST ()
[
 CHAR *SECOND() /* SECOND */
 PRINTF ("%C", *SECOND (NAME))
 RETURN NAME
]
CHAR *
SECOND (INITIALS)
 CHAR INITIAL [2]
[
 RETURN INITIAL + 1
]

```

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ - ПЕЧАТЬ JB.

#### -ОПИСАНИЕ ТИПОВ -0

ОБЪЯВЛЕНИЕ (ОПИСАНИЕ, ПРЕОБРАЗОВАНИЕ) ТИПА АНАЛОГИЧНО ОПРЕДЕЛЕНИЮ ПЕРЕМЕННОЙ ЗА ИСКЛЮЧЕНИЕМ КЛЮЧЕВОГО СЛОВА TYPEDEF.

```

- TYPEDEF CHAR *PTR,
 VEC [3]
- ОПРЕДЕЛЯЕТ ТИП PTR, ИДЕНТИФИЦИРУЮЩИЙ УКАЗАТЕЛЬ ДАННЫХ ТИПА
 (CHAR) И И ТИП VEC, ИДЕНТИФИЦИРУЮЩИЙ 3-Х ЭЛЕМЕНТНЫЙ МАССИВ ДАН-
 НЫХ ТИПА (CHAR). ПОСЛЕ ТАКОГО ОПРЕДЕЛЕНИЯ ТИПОВ МОЖНО НАПИСАТЬ:
 PTR ARR[2], CHRPTR
 VEC *REF

```

В ПЕРВОЙ СТРОКЕ ОБЪЯВЛЯЕТСЯ 2-Х ЭЛЕМЕНТНЫЙ МАССИВ ПЕРЕМЕННЫХ ТИПА (PTR), А, СЛЕДОВАТЕЛЬНО, МАССИВ УКАЗАТЕЛЕЙ ДАННЫХ ТИПА (CHAR), И ПЕРЕМЕННАЯ CHRPTR, УКАЗЫВАЮЩАЯ НА ТАКИЕ ДАННЫЕ. ВО ВТОРОЙ - УКАЗАТЕЛЬ ДАННЫХ ТИПА VEC, СЛЕДОВАТЕЛЬНО, УКАЗАТЕЛЬ 3-Х ЭЛЕМЕНТНЫХ МАССИВОВ ТИПА (CHAR).

ЛЕГКО ЗАМЕТИТЬ, ЧТО ТАКОЕ ОПРЕДЕЛЕНИЕ РАВНОСИЛЬНО:

```

CHAR *ARR[2], CHRPTR
CHAR (*REF)[3]

```

ПРИМЕР:

```

TYPEDEF STRUCT LIST [
 INT DATA
 STRUCT LIST *LINCK
] LIST

```

- LIST ЯВЛЯЕТСЯ ИМЕНЕМ ТИПА (STRUCT LIST). ТАКИМ ОБРАЗОМ ДЕКЛАРАЦИЯ LIST STR[3], \*REF АНАЛОГИЧНА STRUCT LIST STR[3], \*REF

#### 4. ПОСТРОЕНИЕ ВЫРАЖЕНИЙ

ОСНОВНЫМ ЭЛЕМЕНТОМ ЯЗЫКА СИ ЯВЛЯЮТСЯ ПЕРВИЧНЫЕ ВЫРАЖЕНИЯ. ИМИ ЯВЛЯЮТСЯ:

- ИДЕНТИФИКАТОР: VOLUME, FUN И ДР.
- ЛИТЕРАЛ: 13, "JB", 'N' И ДР.
- НЕКОТОРОЕ ВЫРАЖЕНИЕ В КРУГЛЫХ СКОБКАХ: (A + B)
- ВЫРАЖЕНИЕ, ПРЕДСТАВЛЯЮЩЕЕ ФУНКЦИЮ, ЗА КОТОРОЙ СЛЕДУЕТ СПИСОК АРГУМЕНТОВ В КРУГЛЫХ СКОБКАХ: FUN (A,B)
- ПЕРВИЧНОЕ ВЫРАЖЕНИЕ, ПОСЛЕ КОТОРОГО СТОИТ ВЫРАЖЕНИЕ В КВАДРАТНЫХ СКОБКАХ: ARR[3+1]
- ПЕРВИЧНОЕ L - ВЫРАЖЕНИЕ (Т.Е. ВЫРАЖЕНИЕ, КОТОРОЕ СТОИТ СЛЕВА ОТ ЗНАКА ИЛИ СИМВОЛА = ), ПРЕДСТАВЛЯЮЩЕЕ СОБОЙ СТРУКТУРУ, СРАЗУ ПОСЛЕ КОТОРОЙ СТОИТ ЗНАК. И ИДЕНТИФИКАТОР ПОЛЯ СТРУКТУРЫ: STR.CNR
- ПЕРВИЧНОЕ ВЫРАЖЕНИЕ, ПРЕДСТАВЛЯЮЩЕЕ УКАЗАТЕЛЬ СТРУКТУРЫ, НЕПОСРЕДСТВЕННО ПОСЛЕ КОТОРОГО СТОИТ СИМВОЛ -> И ИДЕНТИФИКАТОР ПОЛЯ СТРУКТУРЫ: PTR -> CNR.

ДОПОЛНИТЕЛЬНЫЕ ПРАВИЛА:

- L-ВЫРАЖЕНИЕМ НЕ ЯВЛЯЕТСЯ ЗАПИСЬ, ПРЕДСТАВЛЯЮЩАЯ УКАЗАТЕЛЬ ОБЪЕКТА: &FIX
- В ЗАПИСИ ВИДА: ПЕРВИЧНОЕ-ВЫРАЖЕНИЕ (СПИСОК-ВЫРАЖЕНИЙ). ПЕРВИЧНОЕ ВЫРАЖЕНИЕ ДОЛЖНО ПРЕДСТАВЛЯТЬ ФУНКЦИЮ: FUN(A,B)
- В ЗАПИСИ ВИДА: ПЕРВИЧНОЕ-ВЫРАЖЕНИЕ [ВЫРАЖЕНИЕ], ОБЯЗАТЕЛЬНО ОДНО ИЗ ВЫРАЖЕНИЙ ДОЛЖНО БЫТЬ УКАЗАТЕЛЕМ НА ЭЛЕМЕНТЫ МАССИВА: 2["JAN"].

ВЫРАЖЕНИЕМ ЯВЛЯЕТСЯ ВЫРАЖЕНИЕ С ПРЕДШЕСТВУЮЩИМ ОПЕРАТОРОМ КОНВЕРСИИ (ИМЯ-ТИПА). ЕСЛИ EXR ЯВЛЯЕТСЯ ВЫРАЖЕНИЕМ, ПРЕДСТАВЛЯЮЩИМ НЕКОТОРОЕ ДАННОЕ, ТО (ИМЯ-ТИПА)EXR ЯВЛЯЕТСЯ ВЫРАЖЕНИЕМ, ПРЕДСТАВЛЯЮЩИМ ДАННОЕ ТИПА (ИМЯ-ТИПА). ИМЕНЕМ ТИПА ЯВЛЯЕТСЯ ЗАПИСЬ, СОСТОЯЩАЯ ИЗ ОПИСАНИЯ ТИПА И ДЕКЛАРАТОРА. ЕСЛИ, К ПРИМЕРУ, EXR-ВЫРАЖЕНИЕ ТИПА

(INT \*), ТО (CNR \*)EXR ПРЕДСТАВЛЯЕТ ДАННОЕ ТИПА (CNR \*).

- 1UWAGA -0: В СИСТЕМЕ NISOFT ЗАПИСЬ (ИМЯ-ТИПА), ВЫСТУПАЮЩАЯ В ЗНАЧЕНИИ ОПЕРАТОРА КОНВЕРСИИ, ДОЛЖНА БЫТЬ УПРЕЖДЕНА КЛЮЧЕВЫМ СЛОВОМ CAST: CAST(CNR \*)EXR.

ВЫРАЖЕНИЕМ ЯВЛЯЕТСЯ ИМЯ ТИПА В КРУГЛЫХ СКОБКАХ С ПРЕДШЕСТВУЮЩИМ ОПЕРАТОРОМ РАЗМЕРА SIZEOF. ЕСЛИ NAM - ИМЯ ТИПА, ТО SIZEOF (NAM) ЯВЛЯЕТСЯ ВЫРАЖЕНИЕМ, ПРЕДСТАВЛЯЮЩИМ ДАННЫЕ, ВЕЛИЧИНОЙ, РАВНОЙ ЧИСЛУ БАЙТ, НЕОБХОДИМЫХ ДЛЯ ХРАНЕНИЯ В ОЗУ ВНУТРЕННЕГО ДАННОГО ТИПА (NAM).

- 1UWAGA -0: В СИСТЕМЕ NISOFT ИМЯ ТИПА ДОЛЖНО БЫТЬ КЛЮЧЕВЫМ СЛОВОМ, ОЗНАЧАЮЩИМ ТИП, К ПРИМЕРУ, INT ИЛИ ИДЕНТИФИКАТОР ТИПА.

- 1UWAGA -0: В СИСТЕМЕ NISOFT ОПЕРАТОР SIZEOF НЕ МОЖЕТ ОТНОСИТЬСЯ К ВЫРАЖЕНИЮ.

ВЫРАЖЕНИЕМ ЯВЛЯЕТСЯ ЗАПИСЬ ВИДА:

L-ВЫРАЖЕНИЕ ОПЕРАТОР-ПРИСВАИВАНИЯ ВЫРАЖЕНИЕ

ОПЕРАТОР -ПРИСВАИВАНИЯ - =, \*=, =, %=, +=, -=, <<=, >>=, &=, ^= И !=. ВЫПОЛНЕНИЕ ОПЕРАЦИИ A@=B, ГДЕ @ - ОДИН ИЗ ВЫШЕУКАЗАННЫХ

СИМВОЛОВ, РАВНОСИЛЬНО ВЫПОЛНЕНИЮ ОПЕРАЦИИ  $A=A \oplus B$ .

ВЫРАЖЕНИЕМ ЯВЛЯЕТСЯ ЗАПИСЬ ВИДА:

ВЫРАЖЕНИЕ? ВЫРАЖЕНИЕ-1: ВЫРАЖЕНИЕ-2

ЕСЛИ ВЫРАЖЕНИЕ ИМЕЕТ ЗНАЧЕНИЕ, ОТЛИЧНОЕ ОТ 0, ТО РЕЗУЛЬТАТОМ 3Х АРГУМЕНТНОЙ ОПЕРАЦИИ ? : ЯВЛЯЕТСЯ ДАННОЕ, ПРЕДСТАВЛЕННОЕ ВЫРАЖЕНИЕМ-1, В ПРОТИВНОМ СЛУЧАЕ - ВЫРАЖЕНИЕМ-2.

ПРИМЕР:

```
CHAR ARR[2][2] = {'J', 'A', 'N', 'B'},
* PTR
MAIN ()
[
 TYPEDEF CHAR *CHRPTR
 PTR = CAST(CHRPTR)(ARR+1)+1
 PRINTF ("%C%C", **ARR, *PTR)
]
```

КОНВЕРСИЯ МОЖЕТ БЫТЬ ЯВНОЙ И НЕЯВНОЙ. ЯВНАЯ КОНВЕРСИЯ ЯВЛЯЕТСЯ ВЫРАЖЕНИЕМ С ПРЕДШЕСТВУЮЩИМ КЛЮЧЕВЫМ СЛОВОМ CAST (ТОЛЬКО В HISOFT). В ПРИМЕРЕ ПРИМЕНЕН ОПЕРАТОР КОНВЕРСИИ К ВЫРАЖЕНИЮ (ARR+1). ПРИ ЭТОМ ВЫРАЖЕНИЕ (ARR + 1), ПРЕДСТАВЛЯЮЩЕЕ ДАННОЕ ТИПА (CHAR(\*)[2]), ПРЕОБРАЗУЕТСЯ В ВЫРАЖЕНИЕ, ПРЕДСТАВЛЯЮЩЕЕ ДАННОЕ ТИПА (CHAR \*), Т.Е. В ВЫРАЖЕНИЕ ТАКОГО ЖЕ ТИПА, КАК И PTR.

ЕСЛИ ДАННЫЕ ТИПА (CHAR) ПРИСВАИВАЮТСЯ ПЕРЕМЕННОЙ ТИПА (INT) ИЛИ (UNSIGNED), ТО ПРИ ЭТОМ САМЫЙ СТАРШИЙ БАЙТ ЭТОЙ ПЕРЕМЕННОЙ ЗАПОЛНИТСЯ НУЛЯМИ. И, НАОБОРОТ, ЕСЛИ ДАННЫЕ ТИПА (INT) ИЛИ (UNSIGNED)

БУДУТ ПРИСВОЕНЫ ПЕРЕМЕННОЙ ТИПА (CHAR), ТО ПЕРЕМЕННОЙ ПРИСВОИТСЯ МЛАДШИЙ БАЙТ ДАННОГО. А ЕСЛИ ДАННОЕ ТИПА (INT) ПРИСВОИТЬ ПЕРЕМЕННОЙ ТИПА (UNSIGNED) И НАОБОРОТ, ТО МОЖНО ПОТЕРЯТЬ ДАННЫЕ.

НЕЯВНАЯ КОНВЕРСИЯ МОЖЕТ ВЫПОЛНЯТЬСЯ И ПРИ ВЫПОЛНЕНИИ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ:

- ЕСЛИ НЕКОТОРЫЙ АРГУМЕНТ ИМЕЕТ ТИП (CHAR), ТО ОН БУДЕТ ПРЕОБРАЗОВАН К ТИПУ (INT)

- ЕСЛИ ТОЛЬКО ОДИН АРГУМЕНТ 2Х-АРГУМЕНТНОЙ ОПЕРАЦИИ ИМЕЕТ ТИП (UNSIGNED), ТО ВТОРОЙ БУДЕТ ПРЕОБРАЗОВАН К ЭТОМУ ЖЕ ТИПУ

- В ДРУГИХ СЛУЧАЯХ ОБА АРГУМЕНТА ДОЛЖНЫ БЫТЬ ТИПА (INT), ТАКИМ ЖЕ БУДЕТ И РЕЗУЛЬТАТ.

## 5. ВЫПОЛНЕНИЕ КОМАНД

В СИ ИМЕЮТСЯ КОМАНДЫ-ВЫРАЖЕНИЯ ПУСТЫЕ, ГРУППОВЫЕ И НАЧИНАЮЩИЕСЯ С КЛЮЧЕВОГО СЛОВА. КАЖДОЙ КОМАНДЕ, ЗА ИСКЛЮЧЕНИЕМ ГРУППОВОЙ, ОБРАЗУЮЩЕЙ ТЕЛО ФУНКЦИИ, МОЖЕТ ПРЕДШЕСТВОВАТЬ МЕТКА ИЛИ ГРУППА МЕТОК ЧЕРЕЗ ДВОЕТОЧИЕ:

```
FIN: RETURN A+B
```

КОМАНДА-ВЫРАЖЕНИЕ ИМЕЕТ ВИД: EXP - ГДЕ EXP ЯВЛЯЕТСЯ ПРОИЗВОЛЬНЫМ ВЫРАЖЕНИЕМ. ПРИ ВЫПОЛНЕНИИ ТАКОЙ КОМАНДЫ ПРОИСХОДИТ ВЫЧИСЛЕНИЕ ДАННОГО, ПРЕДСТАВЛЕННОГО ВЫРАЖЕНИЕМ EXP, А ЗАТЕМ ЕГО ИГНОРИРОВАНИЕ. ПРИ ВЫЧИСЛЕНИИ  $2+(A=5)$  ВЫЧИСЛЯЕТСЯ ДАННОЕ, РАВНОЕ 7 И ЭТО ДАННОЕ ЗАТЕМ ИГНОРИРУЕТСЯ. Т.О  $2+(A=5)$  АНАЛОГИЧНО  $A=5$ .

## 6. ВЫЗОВ ФУНКЦИИ

ВЫЗОВ ФУНКЦИИ ИМЕЕТ ВИД:  
FUN (ARG1, ARG2, ..., ARGN)

ПРИМЕР:

```
MAIN ()
[
 INT PRINT ()
 OUTPUT (13, PRINT)
]

OUTPUT (NUM, FUN)
 INT NUM, (*FUN)()
[
 (*FUN)("%D", NUM)
]

PRINT (FMT, VAL)
 CHAR *FMT
 INT VAL
[
 PRINTF(FMT, -VAL)
]
```

- ПРИ ВЫПОЛНЕНИИ ПРИМЕРА БУДЕТ НАПЕЧАТАНО -13. АРГУМЕНТ PRINT НЕЯВНО ПРЕОБРАЗУЕТСЯ НА УКАЗАНИЕ ФУНКЦИИ PRINT, СЛЕДОВАТЕЛЬНО, \*FUN ПРЕДСТАВЛЯЕТ ФУНКЦИЮ PRINTF, А КОМАНДА:

(\*FUN)("%D", NUM)

ВЫПОЛНЯЕТСЯ КАК КОМАНДА PRINTF("%D", NUM)

## 7. ВХОДНОЕ ПРЕОБРАЗОВАНИЕ ПРОГРАММ

ИНТЕГРАЛЬНАЯ ЧАСТЬ КОМПИЛЯТОРА NISOFT<sup>®</sup> ОБРАЗУЕТ ПРОЦЕССОР - ПОДСИСТЕМУ, ОСУЩЕСТВЛЯЮЩУЮ ВХОДНОЕ ПРЕОБРАЗОВАНИЕ ПРОГРАММ. ДИРЕКТИВЫ ПРОЦЕССОРА ОТЛИЧАЮТСЯ ОТ ДРУГИХ ЗАПИСЕЙ ТЕМ, ЧТО ОНИ НАХОДЯТСЯ В ОТДЕЛЬНЫХ СТРОКАХ, НАЧИНАЮЩИХСЯ С 'РЕШЕТКИ' - #.

НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕМОЙ ДИРЕКТИВОЙ ПРОЦЕССОРА ЯВЛЯЕТСЯ ОПРЕДЕЛЯЮЩАЯ ДИРЕКТИВА. В ОБЩЕМ СЛУЧАЕ:

#DEFINE ИДЕНТИФИКАТОР ПОСЛЕДОВАТЕЛЬНОСТЬ-ЛЕКСИЧЕСКИХ-ЕДИНИЦ. ПРИ ИНТЕРПРЕТАЦИИ ЭТОЙ ДИРЕКТИВЫ ПОСЛЕДОВАТЕЛЬНОСТЬ-ЛЕКСИЧЕСКИХ-ЕДИНИЦ СВЯЗЫВАЕТСЯ С ИДЕНТИФИКАТОРОМ И ЗАМЕНЯЕТ ЕГО В ТЕКСТЕ ПРОГРАММЫ.

```
#DEFINE NULL 0
#DEFINE TRUE 1
#DEFINE VOID INT
MAIN ()
[
 VOID OUT()
 OUT ("IZABELA")
]
VOID
OUT (NAME)
 CHAR *NAME
[
 IF (NAME == NULL)
 RETURN
 WHILE (TRUE) [
 PRINTF ("%C", *NAME)
```



```
IF (*NAME++ == NULL)
 RETURN
]
```

#### ДИРЕКТИВА # INCLUDE

ПРИ ИНТЕРПРЕТАЦИИ ДИРЕКТИВЫ БЕЗ ПАРАМЕТРОВ В ПРОЦЕССЕ КОМПИЛЯЦИИ ПРОГРАММЫ ТЕКСТ ИЗ БУФЕРА РЕДАКТОРА ВКЛЮЧАЕТСЯ В МЕСТО ПОЯВЛЕНИЯ ЭТОЙ ДИРЕКТИВЫ.

ДИРЕКТИВА #INCLUDE С ПАРАМЕТРАМИ:

#INCLUDE "ИМЯ" — ВКЛЮЧЕНИЕ В МЕСТО ПОЯВЛЕНИЯ ДИРЕКТИВЫ ТЕКСТА ИЗ ФАЙЛА "ИМЯ". ПРИ ИСПОЛЬЗОВАНИИ МАГНИТОФОНА, СРАЗУ ПОСЛЕ ИСПОЛЬЗОВАНИЯ ЭТОЙ ДИРЕКТИВЫ НЕОБХОДИМО ВКЛЮЧИТЬ МАГНИТОФОН.

В ТОМ СЛУЧАЕ, ЕСЛИ ФАЙЛ "ИМЯ" ЯВЛЯЕТСЯ БИБЛИОТЕКОЙ ФУНКЦИЙ К КОТОРЫМ ЕСТЬ ОБРАЩЕНИЯ В ПРОГРАММЕ, ТО ДЛЯ ИЗБИРАТЕЛЬНОГО ВКЛЮЧЕНИЯ ИХ В ПРОГРАММУ НЕОБХОДИМО ИСПОЛЬЗОВАТЬ ДИРЕКТИВУ:

```
#INCLUDE ? ИМЯ ?
```

ПРИМЕР: ЕСЛИ ИМЕЕМ ПРОГРАММУ:

```
MAIN ()
[
 SUB ()
 FUN ()
]
SUB ()
 [PRINTF ("JANB SUB N")]
```

А.В. ФАЙЛЕ LIBRA ИМЕЕТСЯ ТЕКСТ:

```
SUB ()
 [PRINTF ("LIBRARY SUB ")]
FUN ()
 [PRINTF ("LIBRARY FUN")]
```

ТО ВЫПОЛНЕНИЕ В ПРОЦЕССЕ КОМПИЛЯЦИИ ДВУХ ДИРЕКТИВ:

```
#INCLUDE
#INCLUDE ? LIBRA ?
```

ПРИВЕДЕТ К ПОЯВЛЕНИЮ ПРОГРАММЫ:

```
MAIN ()
[
 SUB ()
 FUN ()
]
SUB ()
 [PRINTF ("JANB SUB N")]
FUN ()
 [PRINTF ("LIBRARY FUN")]
```

ДИРЕКТИВА # TRANSLATE 'ИМЯ'

ДИРЕКТИВА ВЫПОЛНЯЕТСЯ ПЕРЕД КОМПИЛЯЦИЕЙ, ЧТО ПРИВОДИТ К ФОРМИРОВАНИЮ ПРОГРАММНОГО МОДУЛЯ, РАЗМЕЩАЕМОГО В ФАЙЛЕ 'ИМЯ' И ЗАГРУЖАЕМОГО ИЗ BASIC'A КОМАНДАМИ;

```
LOAD "ИМЯ" CODE
RANDOMIZE USP 25200.
```

## 8. ИСПОЛЬЗОВАНИЕ БИБЛИОТЕК ФУНКЦИЙ

СОВМЕСТНО С КОМПИЛЯТОРОМ ЯЗЫКА ФИРМА NISOFT ПОСТАВЛЯЕТ ДВА МОДУЛЯ STDIO.H И STDIO.LIB. ПЕРВЫЙ СОДЕРЖИТ РЯД ПОЛЕЗНЫХ ОПРЕДЕЛЕНИЙ:

```
DEFINE NULL 0
DEFINE FALSE 0
DEFINE TRUE -1
DEFINE EOF -1
typedef int FILE
extern unsigned strlen () и др.
```

А ТАК ЖЕ ОПРЕДЕЛЕНИЕ ДВУХ ФУНКЦИЙ MIN () И MAX ().  
ПРИ НЕОБХОДИМОСТИ, ВСЕ ЭТИ ОПРЕДЕЛЕНИЯ И ДЕКЛАРАЦИИ МОЖНО ВКЛЮЧИТЬ В ПРОГРАММУ:

```
#include "STDIO.H"
main ()
[printf("%d", max('J','B'))]
```

ИЛИ

```
#include "STDIO.H"
main ()
[printf("%d", strlen("JB"))]
#include ? STDIO.H ?
```

ОПИСАНИЕ СОДЕРЖИМОГО ФАЙЛОВ STDIO.H И STDIO.LIB ПРИВЕДЕНО В ПРИЛОЖЕНИИ С. МОЖНО СОЗДАВАТЬ И СОБСТВЕННЫЕ БИБЛИОТЕКИ, СОЗДАВ В РЕДАКТОРЕ ПОСЛЕДОВАТЕЛЬНОСТЬ ОПИСАНИЙ И ВЫПОЛНИВ ДИРЕКТИВУ РЕДАКТОРА: P1,9999,JB.LIB.

## 9. ВЫПОЛНЕНИЕ ОПЕРАЦИИ ВВОДА/ВЫВОДА

В ПРОЦЕССЕ ОПЕРАЦИЙ ВВОДА/ВЫВОДА ПРОИСХОДИТ ОБМЕН ИНФОРМАЦИЕЙ МЕЖДУ МАССИВАМИ ДАННЫХ И ОБОРУДОВАНИЕМ. СВЯЗЬ ФАЙЛА С МАССИВОМ ИЛИ ОБОРУДОВАНИЕМ ОСУЩЕСТВЛЯЕТСЯ С ПОМОЩЬЮ ФУНКЦИИ FOREN(), А ПРЕКРАЩЕНИЕ ЭТОЙ СВЯЗИ — FCLOSE(). ИДЕНТИФИКАЦИЯ ФАЙЛА ПРОИСХОДИТ С ПОМОЩЬЮ ФАЙЛОВОГО УКАЗАТЕЛЯ, ПОЛУЧАЕМОГО КАК РЕЗУЛЬТАТ ФУНКЦИИ FOREN. ЭТИМ УКАЗАТЕЛЕМ ЯВЛЯЕТСЯ ДАННОЕ ТИПА (FILE \*), ЗАВИСЯЩЕЕ ОТ РЕАЛИЗАЦИИ И ОПРЕДЕЛЕННОЕ В ФАЙЛЕ STDIO.H. В СИСТЕМЕ NISOFT ТИП FILE ОПРЕДЕЛЕН С ПОМОЩЬЮ ДИРЕКТИВЫ:

```
typedef int FILE Т.Е. ОН НЕ ОТЛИЧАЕТСЯ ОТ ТИПА (INT)
```

СПЕЦИАЛЬНЫМ ОБРАЗОМ ТРАКТУЮТСЯ ТРИ ФАЙЛА: СТАНДАРТНЫЙ ФАЙЛ ВВОДА, СТАНДАРТНЫЙ ФАЙЛ ВЫВОДА И СТАНДАРТНЫЙ ФАЙЛ СООБЩЕНИЙ ОБ ОШИБКАХ. В СИСТЕМЕ NISOFT СТАНДАРТНЫЙ ФАЙЛ ВВОДА СВЯЗАН С КЛАВИАТУРОЙ, А ФАЙЛЫ ВЫВОДА И СООБЩЕНИЙ ОБ ОШИБКАХ — С ВЕРХНЕЙ ЧАСТЬЮ

ЭКРАНА. ПРИ ОБМЕНЕ С КЛАВИАТУРОЙ, ЭКРАНОМ И ПРИНТЕРОМ, СЛЕДОВАТЕЛЬНО, НЕ ТРЕБУЕТСЯ ПРЕДВАРИТЕЛЬНОЕ ВЫПОЛНЕНИЕ ФУНКЦИИ `FOREN`. ЕСЛИ ТИП (`STREAM`) ОПРЕДЕЛЕН КАК: `TYPDEF FILE *STREAM` ТО УСТАНОВИТСЯ СЛЕДУЮЩЕЕ СООТВЕТСТВИЕ;

| ОБОРУДОВАНИЕ         | ФАЙЛОВЫЙ УКАЗАТЕЛЬ          |
|----------------------|-----------------------------|
| КЛАВИАТУРА           | <code>CAST(STREAM) 0</code> |
| ВЕРХНЯЯ ЧАСТЬ ЭКРАНА | <code>CAST(STREAM) 0</code> |
| НИЖНЯЯ ЧАСТЬ ЭКРАНА  | <code>CAST(STREAM) 1</code> |
| ВЕРХНЯЯ ЧАСТЬ ЭКРАНА | <code>CAST(STREAM) 2</code> |
| ПРИНТЕР              | <code>CAST(STREAM) 3</code> |

НЕКОТОРЫЕ ФУНКЦИИ, НЕЯВНО СВЯЗАННЫХ СО СТАНДАРТНЫМИ ФАЙЛАМИ (`PRINTF` И ДР.), МОГУТ БЫТЬ ВЫЗВАНЫ БЕЗ УСТАНОВКИ ФАЙЛОВОГО УКАЗАТЕЛЯ.

#### ФУНКЦИЯ `FOREN` (ВСТРОЕННАЯ)

```
FILE *FOREN (NAME, MODE)
CHAR *NAME, *MODE
```

ПРИ ВЫЗОВЕ ФУНКЦИИ ОТКРЫВАЕТСЯ ФАЙЛ, СВЯЗАННЫЙ С МАССИВОМ `NAME`, В РЕЖИМЕ, УСТАНОВЛЕННЫМ АРГУМЕНТОМ `MODE`:

```
MODE = "R" - ФАЙЛ НА ВВОД
MODE = "W" - ФАЙЛ НА ВЫВОД.
```

РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ ФАЙЛОВЫЙ УКАЗАТЕЛЬ, ИДЕНТИФИЦИРУЮЩИЙ ОТКРЫТЫЙ ФАЙЛ. ЕСЛИ ФАЙЛ ОТКРЫТЬ НЕВОЗМОЖНО, ТО РЕЗУЛЬТАТ - ПУСТОЙ УКАЗАТЕЛЬ (`NULL`)

#### ФУНКЦИЯ `FCLOSE` (ВСТРОЕННАЯ)

```
FILE *PTR
```

ВЫПОЛНЕНИЕ ФУНКЦИИ `FCLOSE` ПРИВОДИТ К ЗАКРЫТИЮ ФАЙЛА, ИДЕНТИФИЦИРУЕМОГО УКАЗАТЕЛЕМ `PTR`. ЕСЛИ ФАЙЛ БЫЛ ОТКРЫТ НА ВЫВОД, ТО ЕГО ЗАКРЫТИЕ БУДЕТ СОПРОВОЖДАТЬСЯ ЗНАКОМ 'КОНЕЦ ФАЙЛА' - `EOF`, КОТОРЫМ В `HISOFT` ЯВЛЯЕТСЯ (`SS+1`) ИЛИ `-1`.

```
TYPDEF INT FILE
MAIN ()
{
 FILE *OUT, *FOREN()
 OUT = FOREN("JANB","W")
 PUTC ('J', OUT)
 PUTC ('B', OUT)
 FCLOSE (OUT)
}
```

- ВЫПОЛНЕНИЕ ФУНКЦИИ `FOREN` ПРИВОДИТ К ОТКРЫТИЮ ФАЙЛА, СВЯЗАННОГО С МАССИВОМ `JANB`, НА ВЫВОД. РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ ФАЙЛОВЫЙ УКАЗАТЕЛЬ, ИДЕНТИФИЦИРУЮЩИЙ УПОМЯНУТЫЙ ФАЙЛ. ЭТОТ УКАЗАТЕЛЬ ПРИСВОЕН ФАЙЛОВОЙ ПЕРЕМЕННОЙ `OUT`, К КОТОРОЙ ИМЕЮТСЯ ОБРАЩЕНИЯ В ФУНКЦИЯХ `PUTC` И `FCLOSE`. ТАКИМ ОБРАЗОМ, ВЫПОЛНЕНИЕ ЭТИХ ФУНКЦИЙ В ОБЩЕМ СЛУЧАЕ ОТНОСИТСЯ К МАССИВУ `JANB`. ВЫПОЛНЕНИЕ ФУНКЦИИ `FCLOSE` ПРИВОДИТ К ЗАКРЫТИЮ ФАЙЛА, ДОПОЛНИВ ОНЫИ `EOF`.

ФУНКЦИЯ GETC (ВСТРОЕННАЯ)  
INT GETC (INP)  
FILE \*INP

ВЫВОД ОЧЕРЕДНОГО ЗНАКА ИЗ ФАЙЛА INP. РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ ДАННОЕ ТИПА (INT) ВЕЛИЧИНОЙ, РАВНОЙ КОДУ ВВЕДЕННОГО ЗНАКА

ФУНКЦИЯ UNGETC (ВСТРОЕННАЯ)  
UNGETC (C, INP)  
CHAR C  
FILE \*INP

ВОЗВРАТ В ФАЙЛ INP, СИМВОЛА, ПРЕДСТАВЛЕННОГО C. ВОЗВРАТ ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТО ПРИ ПОСЛЕДУЮЩЕМ ВЫПОЛНЕНИИ ФУНКЦИИ GETC ОСУЩЕСТВЛЯЕТСЯ ДОСТУП К ВОЗВРАЩЕННОМУ СИМВОЛУ.

ФУНКЦИЯ PUTC (ВСТРОЕННАЯ)  
INT PUTC (C, OUT)  
CHAR C  
FILE \*OUT

ВЫВОД В ФАЙЛ OUT ЗНАКА C. РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ ДАННОЕ ТИПА (INT) ВЕЛИЧИНОЙ, РАВНОЙ КОДУ ВВЕДЕННОГО ЗНАКА.

typedef int FILE  
typedef FILE \*STREAM  
# define EOF (-1)  
# define MONITOR CAST(STREAM)0

```
MAIN ()
{
 FILE *INP, *FOPEN ()
 INT CH
 IF (INP=FOPEN("STDIO.H", "R")) [
 WHILE ((CH=GETC(INP))!=EOF)
 PUTC (CH, MONITOR)
 PUTC (' N', MONITOR)
]
}
```

- ВЫВОД ФАЙЛА STDIO.H НА МОНИТОР.

ФУНКЦИЯ GETCHAR (ВСТРОЕННАЯ)  
INT GETCHAR ()

ВВОД ОДНОГО СИМВОЛА ИЗ СТАНДАРТНОГО ФАЙЛА ВВОДА. РЕЗУЛЬТАТ ФУНКЦИИ - ДАННОЕ ТИПА (INT), РАВНОЕ КОДУ ВВЕДЕННОГО СИМВОЛА ИЛИ EOF.

ФУНКЦИЯ PUTCHAR (ВСТРОЕННАЯ)  
INT PUTCHAR (C)  
INT C

ВЫВОД В СТАНДАРТНЫЙ ФАЙЛ ВЫВОДА ОДНОГО ЗНАКА С КОДОМ C. РЕЗУЛЬТАТ ФУНКЦИИ - ДАННОЕ ТИПА (INT), РАВНОЕ КОДУ ВЫВОДИМОГО ЗНАКА.

```
DEFINE EOF (-1)
CHAR FLAG
MAIN ()
[
 INT CH
 IF ((CH = GETCHAR()) != EOF)
 MAIN ()
 IF (!FLAG) [
 FLAG = 1
 PUTCHAR (' N')
]
 PUTCHAR (CH)
]
```

- ВВОД С КЛАВИАТУРЫ ПОСЛЕДОВАТЕЛЬНОСТИ ЗНАКОВ, ОКОНЧИВАЮЩЕЯСЯ EOF, А ЗАТЕМ ВЫВОД ЭТОЙ ПОСЛЕДОВАТЕЛЬНОСТИ В ОБРАТНОМ ПОРЯДКЕ.

ФУНКЦИЯ PRINTF (ВСТРОЕННАЯ)  
 PRINTF (FMT, P1, P2, ...., PN)  
 CHAR \*FMT

ВЫПОЛНЕНИЕ ДАННОЙ ФУНКЦИИ ПРИВОДИТ К ВЫВОДУ В СТАНДАРТНЫЙ ФАЙЛ ВЫВОДА ПОД УПРАВЛЕНИЕМ СПИСКА ФОРМАТОВ FMT ДАННЫХ, ПРЕДСТАВЛЕННЫХ P1 ... PN. ФУНКЦИЯ PRINTF МОЖЕТ БЫТЬ ВЫЗВАНА С ПЕРЕМЕННЫМ ЧИСЛОМ АРГУМЕНТОВ. НА ОДИН БОЛЬШЕ, ЧЕМ ФОРМАТОВ КОНВЕРСИИ, СОДЕРЖАЩИХСЯ В ПОСЛЕДОВАТЕЛЬНОСТИ FMT. КАЖДЫЙ ФОРМАТ, ЕСЛИ ОН СУЩЕСТВУЕТ, СОСТОИТ ИЗ ПОСЛЕДОВАТЕЛЬНОСТИ ЗНАКОВ, НАЧИНАЮЩЕЙСЯ С СИМВОЛА % И ЗАКАНЧИВАЮЩЕЙСЯ ОДНОЙ ИЗ БУКВ: D, O, X, U, C, S. % - НАЧАЛО ФОРМАТА ПРЕОБРАЗОВАНИЯ. ОЗНАЧАЕТ, ЧТО ВЫВОДИМАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ ДОЛЖНА БЫТЬ ВЫРОВНЕНА В ВЫХОДНОМ ПОЛЕ ВЛЕВО

D - ВЫВОД АРГУМЕНТОВ В ВИДЕ ДЕСЯТИЧНЫХ ЧИСЕЛ  
 O - ВЫВОД АРГУМЕНТОВ В ВИДЕ ВОСЬМЕРИЧНЫХ ЧИСЕЛ  
 X - ВЫВОД АРГУМЕНТОВ В ВИДЕ ШЕСТНАДЦАТЕРИЧНЫХ ЧИСЕЛ  
 U - ВЫВОД АРГУМЕНТОВ В ВИДЕ ЧИСЛА БЕЗ ЗНАКА  
 C - ВЫВОД АРГУМЕНТОВ В ВИДЕ ОДНОГО ЗНАКА  
 S - ВЫВОД ПОСЛЕДОВАТЕЛЬНОСТИ ЗНАКОВ, УКАЗАННЫХ АРГУМЕНТОМ.

```
PRINTF("ANY TEXT") ----> ANY TEXT
PRINTF("(%-2D)",5) ----> (5)
PRINTF("(%4.3S)","JANB") ----> (JAN)
PRINTF("%C","B") ----> B
PRINTF("%X", -1) ----> FFFF
```

ФУНКЦИЯ FPRINTF (ВСТРОЕННАЯ)  
 FPRINTF (OUT,FMT, P1, P2, ...., PN)  
 FILE \*OUT  
 CHAR \*FMT

ВЫВОД В ФАЙЛ, ИДЕНТИФИЦИРОВАННЫЙ OUT, В ФОРМАТЕ FMT ПЕРЕМЕННЫХ P1 - PN. АНАЛОГИЧНА PRINTF МОЖЕТ БЫТЬ ВЫЗВАНА С ПЕРЕМЕННЫМ ЧИСЛОМ АРГУМЕНТОВ.

ФУНКЦИЯ SPRINTF (ВСТРОЕННАЯ)  
 SPRINTF (STR,FMT, P1, P2, ...., PN)  
 CHAR \*STR  
 CHAR \*FMT

ПРИ ВЫПОЛНЕНИИ ЭТОЙ ФУНКЦИИ ДАННЫЕ P1 - PN РАЗМЕЩАЮТСЯ В ПОЛЕ ПАМЯТИ STR ПОД УПРАВЛЕНИЕМ FMT. ФУНКЦИЯ МОЖЕТ БЫТЬ ВЫЗВАНА С ПЕРЕМЕННЫМ ЧИСЛОМ АРГУМЕНТОВ.



```
INCLUDE "STDIO.H"
FILE *INP, *FOPEN ()
CHAR NUMBER[7]
INT IND, CH, COUNT
MAIN ()
[
 IF (INP=FOPEN("JANEK", "r")) [
 WHILE (CH = GETC(INP)) != EOF)
 COUNT ++
 SPRINTF (NUMBER, "%u%c", COUNT, '0')
 WHILE (NUMBER[IND] == ' ')
 NUMBER[IND++] = '*'
 PRINTF ("JANEK ZAWIERA %S ZNAKOW", NUMBER)
]
]
```

- ПОДСЧЕТ ЗНАКОВ В ФАЙЛЕ JANEK И ВЫВОД В ВИДЕ;  
JANEK ZAWIERA \* \* \* 13 ZNAKOW

ФУНКЦИЯ SCANF (ВСТРОЕННАЯ)  
INT SCANF (FMT, P1, P2, ..., PN)  
CHAR \*FMT

ВЫПОЛНЕНИЕ ДАННОЙ ФУНКЦИИ ПРИВОДИТ К ВВОДУ ИЗ СТАНДАРТНОГО ФАЙЛА ВВОДА ПОД УПРАВЛЕНИЕМ ФОРМАТОВ FMT УСТАНОВЛЕННОГО ЧИСЛА ЗНАКОВ, ТРАКТОВКУ ИХ КАК ПОСЛЕДОВАТЕЛЬНОСТЬ ВЕЛИЧИН ПЕРЕМЕННЫХ И ПРИСВОЕНИЕ ЭТИХ ВЕЛИЧИН АРГУМЕНТАМ P1 ... PN. ФУНКЦИЯ МОЖЕТ БЫТЬ ВЫЗВАНА С ПЕРЕМЕННЫМ ЧИСЛОМ АРГУМЕНТОВ, ЗАВИСЯЩИМ ОТ ЧИСЛА ФОРМАТОВ. КАЖДЫЙ ФОРМАТ, ЕСЛИ ОН ИМЕЕТСЯ, СОСТОИТ ИЗ ПОСЛЕДОВАТЕЛЬНОСТИ ЗНАКОВ, НАЧИНАЮЩЕЙСЯ С СИМВОЛА % И КОНЧАЮЩЕЙСЯ ОДНОЙ ИЗ БУКВ: D, O, X, C, S. ПОСЛЕДОВАТЕЛЬНОСТЬ ЗНАКОВ, УКАЗАННЫХ FMT, В ОБЩЕМ СЛУЧАЕ СОСТОИТ ИЗ ПРОБЕЛА, ЗНАКОВ, ОТЛИЧНЫХ ОТ ПРОБЕЛА, И ФОРМАТОВ ПРЕОБРАЗОВАНИЯ. ЕСЛИ В ТАКОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИСУТСТВУЕТ ПРОБЕЛ, ВО ВХОДНОМ ФАЙЛЕ ЕМУ ДОЛЖНА СООТВЕТСТВОВАТЬ ПОСЛЕДОВАТЕЛЬНОСТЬ ОТСТУПОВ (ПРОБЕЛЫ, ЗНАКИ ТАБУЛЯЦИИ И ЗНАКИ НОВОЙ СТРОКИ) ЕСЛИ В РАССМАТРИВАЕМОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ПРИСУТСТВУЕТ ЗНАК, ОТЛИЧНЫЙ ОТ ПРОБЕЛА, ТО ВО ВХОДНОМ ФАЙЛЕ ДОЛЖЕН ПРИСУТСТВОВАТЬ ТАКОЙ ЖЕ ЗНАК. ЕСЛИ ЖЕ ИМЕЕТ МЕСТО БЫТЬ УКАЗАТЕЛЬ ПРЕОБРАЗОВАНИЯ, ТО ОН ОПРЕДЕЛЯЕТ РАЗМЕР И СПОСОБ ИНТЕРПРЕТАЦИИ СЛЕДУЮЩЕГО ПОЛЯ ВХОДНОГО ФАЙЛА. ТАКОЕ ПОЛЕ ОГРАНИЧИВАЕТСЯ БЛИЖАЙШИМ ОТСТУПОМ, НО ЕСЛИ В ФОРМАТЕ УКАЗАНА ВЕЛИЧИНА ПОЛЯ, ТО ПОЛЕ ИМЕЕТ НЕ БОЛЬШЕ ЗНАКОВ, ЧЕМ УКАЗАНО В ФОРМАТЕ. ЕСЛИ ПОСЛЕ НАКА % СТОИТ ЗНАК \*, ТО БЛИЖАЙШЕЕ ПОЛЕ ВХОДНОГО ФАЙЛА ИГНОРИРУЕТСЯ.

ОЧЕНЬ ВАЖНО, ЧТОБЫ ТИП ПЕРЕМЕННЫХ, УКАЗЫВАЕМЫХ АРГУМЕНТАМИ ФУНКЦИИ SCANF, СООТВЕТСТВОВАЛ ФОРМАТАМ. К ПРИМЕРУ, ЕСЛИ БЫЛ ИСПОЛЬЗОВАН ФОРМАТ %D, ТО ПЕРЕМЕННАЯ, УКАЗАННАЯ АРГУМЕНТОМ, МОЖЕТ БЫТЬ ТИПА (INT), НО НЕ МОЖЕТ БЫТЬ ТИПА (CHAR).

РЕЗУЛЬТАТОМ ФУНКЦИИ SCANF ЯВЛЯЕТСЯ ДАННОЕ ТИПА (INT), РАВНОЕ ЧИСЛУ ПЕРЕМЕННЫХ, КОТОРЫМ БЫЛИ ПРИСВОЕНЫ ДАННЫЕ ИЛИ EOF.

ФУНКЦИЯ FSCANF (ВСТРОЕННАЯ)  
INT FSCANF (INP, FMT, P1, P2, ..., PN)  
FILE \*INP  
CHAR \*FMT

ВВОД ИЗ ФАЙЛА, УКАЗАННОГО INP, ПОД УПРАВЛЕНИЕМ FMT ....  
АНАЛОГИЧНО SCANF.

ФУНКЦИЯ SSCANF (ВСТРОЕННАЯ)  
 INT SSCANF (STR, FMT, P1, P2, ..., PN)  
 CHAR \*STR  
 CHAR \*FMT

ВВОД ИЗ ПОЛЯ ПАМЯТИ, УКАЗАННОГО STR, ПОД УПРАВЛЕНИЕМ FMT  
 УСТАНОВЛЕННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ЗНАКОВ И Т.П. АНАЛОГИЧНО SCANF.

```

INT MYAGE
CHAR SEP
CHAR MYNAME[20]
MAIN ()
[
 SSCANF("AGE=44 JANB=JANEK",
 "AGE=%D%С%*5S%3S",
 &MYAGE, &SEP, MYNAME)
 PRINTF ("%SIS%DCNOW", MYNAME, MYAGE, SEP)
]
```

- ПЕЧАТЬ JAN IS 44 NOW.

ФУНКЦИЯ RAW (ВСТРОЕННАЯ)  
 RAW ()

ВВОД С КЛАВИАТУРЫ ОДНОГО СИМВОЛА БЕЗ ВЫВОДА ЕГО НА ЭКРАН И БЕЗ  
 ПЕРЕДВИЖЕНИЯ КУРСОРА. РЕЗУЛЬТАТ ФУНКЦИИ - ПЕРЕМЕННАЯ ТИПА (INT),  
 РАВНАЯ КОДУ ВВЕДЕННОГО СИМВОЛА.

ФУНКЦИЯ KEYNIT (ВСТРОЕННАЯ)  
 INT KEYNIT ()

ПРОВЕРКА БУФЕРА КЛАВИАТУРЫ НА ПРЕДМЕТ НАЛИЧИЯ НЕВВЕДЕННОГО  
 СИМВОЛА. РЕЗУЛЬТАТ ФУНКЦИИ - ПЕРЕМЕННАЯ ТИПА (INT), ИМЕЮЩАЯ ВЕ-  
 ЛИЧИНУ 0, ЕСЛИ БУФЕР ЧИСТ И 1 (ИСТИНА), ЕСЛИ В БУФЕРЕ КЛАВИАТУРЫ  
 ИМЕЕТСЯ НЕВВЕДЕННЫЙ СИМВОЛ.

## 10. РАСШИРЕНИЕ И ПРИМЕРЫ

С КАЖДОЙ ПЕРЕМЕННОЙ СВЯЗАН ЕЕ КЛАСС. С НЕКОТОРЫМИ УПРОЩЕНИЯ  
 МИ МОЖНО СКАЗАТЬ, ЧТО ЕСЛИ В ДЕКЛАРАЦИИ НЕ УКАЗАН КЛАСС, ТО ПЕРЕ-  
 МЕННАЯ, ОБ ЯВЛЕННАЯ ВНУТРИ ФУНКЦИИ, ИМЕЕТ КЛАСС AUTO, А ПЕРЕМЕН-  
 НАЯ, ОБ ЯВЛЕННАЯ ВНЕ ФУНКЦИИ - КЛАСС EXTERN. РАЗНИЦА МЕЖДУ НИМИ  
 ЗАКЛЮЧАЕТСЯ В ТОМ, ЧТО ПЕРЕМЕННАЯ КЛАССА AUTO СОЗДАЕТСЯ ЗАНОВО  
 ПЕРЕД КАЖДЫМ ВЫЗОВОМ ФУНКЦИИ И ПЕРЕСТАТ СУЩЕСТВОВАТЬ ПОСЛЕ ОКО-  
 НЧАНИЯ ВЫПОЛНЕНИЯ ФУНКЦИИ, В ТО ВРЕМЯ, КАК ПЕРЕМЕННАЯ КЛАССА EX-  
 TERN СУЩЕСТВУЕТ ВСЕ ВРЕМЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ. ДРУГОЙ ВАЖНОЙ  
 ОСОБЕННОСТЬЮ ПЕРЕМЕННЫХ КЛАССА EXTERN ЯВЛЯЕТСЯ ТО, ЧТО, ЕСЛИ В  
 ПРОГРАММЕ СУЩЕСТВУЕТ НЕСКОЛЬКО ОБ ЯВЛЕНИИ НЕКОТОРОЙ ПЕРЕМЕННОЙ С  
 АТТРИБУТОМ EXTERN И ТОЛЬКО ОДНО ОБ ЯВЛЕНИЕ ЭТОЙ ПЕРЕМЕННОЙ БЕЗ  
 АТТРИБУТА EXTERN, НАХОДЯЩЕЕСЯ ВНЕ ФУНКЦИИ, ТО ВСЕ ЭТИ ОБ ЯВЛЕНИЯ  
 ОТНОСЯТСЯ К ЭТОЙ ПЕРЕМЕННОЙ.

ПЕРЕМЕННЫЕ КЛАССА STATIC СУЩЕСТВУЮТ ВСЕ ВРЕМЯ ВЫПОЛНЕНИЯ  
 ПРОГРАММЫ, НО ЕСЛИ ОНИ ОБ ЯВЛЕНИИ ВНУТРИ ФУНКЦИИ, ТО ОНИ СТАНОВЯ-  
 ТСЯ ДОСТУПНЫ ТОЛЬКО ПРИ ВЫПОЛНЕНИИ ЭТОЙ ФУНКЦИИ. В ДЕКЛАРАЦИЯХ  
 КЛАССА STATIC И В ТЕХ ДЕКЛАРАЦИЯХ ПЕРЕМЕННЫХ КЛАССА EXTERN, В  
 КОТОРЫХ НЕТ СЛОВА EXTERN, МОЖНО ПРИСВАИВАТЬ НАЧАЛЬНЫЕ ЗНАЧЕНИЯ.

-1 ФУНКЦИИ С ПЕРЕМЕННЫМ ЧИСЛОМ АРГУМЕНТОВ -0

ФУНКЦИЮ С ПЕРЕМЕННЫМ ЧИСЛОМ АРГУМЕНТОВ МОЖНО ОБЪЯВИТЬ ПЕРЕД ПЕРВЫМ ВЫЗОВОМ ЭТОЙ ФУНКЦИИ, А В ЗАГОЛОВКЕ ОПРЕДЕЛЕНИЯ, НЕПОСРЕДСТВЕННО ПОСЛЕ СКОБКИ, ЗАМЫКАЮЩЕЙ СПИСОК ПАРАМЕТРОВ, ДОЛЖНО СТОЯТЬ СЛОВО AUTO. ЕСЛИ ЭТИ УСЛОВИЯ СОБЛЕДЕННЫ, ТО КАЖДЫЙ ВЫЗОВ ФУНКЦИИ БУДЕТ НЕЯВНО ДОПОЛНЕН АРГУМЕНТОМ ТИПА (INT), УКАЗЫВАЮЩИМ, СКОЛЬКО БАЙТ ЗАНЯЛИ ВСЕ АРГУМЕНТЫ ФУНКЦИИ, ВКЛЮЧАЯ И НОВЫЙ АРГУМЕНТ. СЛЕДУЕТ ОТМЕТИТЬ, ЧТО АРГУМЕНТЫ ВЫЧИСЛЯЮТСЯ В ПОРЯДКЕ ИХ ВЫСТУПЛЕНИЯ В ВЫЗОВЕ И ЗАПИСЫВАЮТСЯ В ПАМЯТЬ В ПОРЯДКЕ УБЫВАНИЯ АДРЕСОВ. В ЧАСТНОСТИ, РЕЗУЛЬТАТОМ ФУНКЦИИ MAX ЯВЛЯЕТСЯ МАКСИМУМ ПЕРЕДАННЫХ ЕЙ АРГУМЕНТОВ ТИПА (INT):

```
INT
MAX (COUNT) AUTO
 INT COUNT
[
 INT ARGC, *ARGV, MAX
 ARGC = (COUNT >> 1) - 1
 ARGV = COUNT + ARGC
 MAX = -32768
 WHILE (ARGC--)
 IF (*ARGV-- > MAX)
 MAX = ARGV[1]
 RETURN MAX
]
```

-1UWAGA -0: ЭТОГО НЕТ В БАЗОВОМ ЯЗЫКЕ.  
-1КОМАНДА INLINE -0

В СИСТЕМЕ HISOFT ВЫ ИМЕЕТЕ ВОЗМОЖНОСТЬ ВСТАВЛЯТЬ В СВОЮ ПРОГРАММУ НА СИ ФРАГМЕНТЫ В МАШИННЫХ КОДАХ С ПОМОЩЬЮ КОМАНДЫ INLINE АРГУМЕНТАМИ ЭТОЙ КОМАНДЫ МОГУТ БЫТЬ ПРОИЗВОЛЬНЫЕ ВЫРАЖЕНИЯ, В ЧАСТНОСТИ, 16-РИЧНЫЕ ЛИТЕРАЛЫ 0XDD. ПРИНЯТО, ЧТО ПЕРВЫЙ АРГУМЕНТ В ПРЕДЕЛАХ 0-255 ГЕНЕРИРУЕТ ОДИН БАЙТ КОДА, А КАЖДЫЙ ПОСЛЕДУЮЩИЙ - ДВА БАЙТА КОДА. В ФАЙЛЕ STDIO.LIB КОМАНДА INLINE ИСПОЛЬЗУЕТСЯ ДЛЯ ОПРЕДЕЛЕНИЯ РЯДА ГРАФИЧЕСКИХ ФУНКЦИЙ:

```
CLS ()
[
 INLINE (0XC0, 0XD6B)
]
```

-1UWAGA -0: ЭТОГО НЕТ В БАЗОВОМ ЯЗЫКЕ.

-1ПРИМЕРЫ -0

1. ОПРЕДЕЛЕНИЕ ФУНКЦИИ ДЛЯ ВЫЧИСЛЕНИЯ АБСОЛЮТНОЙ ВЕЛИЧИНЫ АРГУМЕНТА ТИПА (INT).

```
INT
ABS (PAR)
 INT PAR
[RETURN PAR < 0 ? -PAR: PAR]
```

2. ОПРЕДЕЛЕНИЕ ФУНКЦИИ ДЛЯ ВЫЧИСЛЕНИЯ МАКСИМАЛЬНОГО ЭЛЕМЕНТА МАССИВА С ЭЛЕМЕНТАМИ ТИПА (INT):

```
INT
MAX (ARR, LEN)
 INT LEN, ARR[]
```

```

[
 INT VAL
 VAL = -32768
 WHILE (LEN--)
 IF (ARR[LEN]>VAL)
 VAL = ARR[LEN]
 RETURN VAL
]

```

3. ОПРЕДЕЛЕНИЕ ФУНКЦИИ ДЛЯ ПОДСЧЕТА ЧИСЛА ЕДИНИЧНЫХ БИТ В АРГУМЕНТЕ ТИПА (INT):

```

INT
COUNT (BITS)
 UNSIGNED BITS
[
 INT TALLY
 FOR (TALLY = 0 BITS BITS >>=1)
 TALLY+ = BITS&1
 RETURN TALLY
]

```

4. ОПРЕДЕЛЕНИЕ ФУНКЦИИ, АРГУМЕНТОМ КОТОРОЙ ЯВЛЯЕТСЯ АДРЕС ПАМЯТИ, А РЕЗУЛЬТАТОМ - ДАННОЕ ТИПА (CHAR), НАХОДЯЩЕЕСЯ ПО ЭТОМУ АДРЕСУ:

```

CHAR
PEEK (ADDR)
 UNSIGNED ADDR
[
 TYPEDEF CHAR *CHRPTR
 RETURN *CAST(CHRPTR)ADDR
]

```

5. ОПРЕДЕЛЕНИЕ ФУНКЦИИ. РОКЕ ДЛЯ ЗАПИСИ ДАННЫХ ТИПА (CHAR) ПО АДРЕСУ ADDR:

```

POKE (ADDR, DATA)
 UNSIGNED ADDR
 CHAR DATA
[
 TYPEDEF CHAR *CHRPTR
 *CAST(CHRPTR)ADDR = DATA
]

```

6. ОПРЕДЕЛЕНИЕ ФУНКЦИИ СРАВНЕНИЯ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ЗНАКОВ:

```

INT
STRCMP (SRC, TRG)
 CHAR *SRC, *TRG
[
 WHILE (*SRC == *TRG++)
 IF (!*SRC++) RETURN 0
 RETURN *SRC > TRG[-1] ? 1: -1
]

```

7. ОПРЕДЕЛЕНИЕ ФУНКЦИИ ДЛЯ СОЗДАНИЯ ДАННОГО ТИПА (INT) ИЗ ПОСЛЕДОВАТЕЛЬНОСТИ ЦИФР:

```

INT
atoi (SRC)
 CHAR *SRC
[
 INT CHR, VAL
 VAL = 0
 WHILE ((CH = *SRC++) == ' ')
 VAL = 10*VAL+CHR - '0'

```

RETURN VAL

]

8. ИСПОЛЬЗУЯ ФУНКЦИИ ДИНАМИЧЕСКОГО УПРАВЛЕНИЯ ПАМЯТЬЮ (CALLOC И FREE), ОПИСАННЫЕ В ПРИЛОЖЕНИИ С, СОЗДАТЬ СТРОКОВЫЙ РЕДАКТОР, ПОЗВОЛЯЮЩИЙ СОЗДАВАТЬ ТЕКСТ, СТРОКИ КОТОРОГО НУМЕРУЮТСЯ ЧИСЛАМИ БЕЗЗНАКА. ПРИНЯТЬ, ЧТО ВВОД СТРОКИ ТРЕБУЕТ ВВОДА ЧИСЛА И ТЕКСТА СТРОКИ. ЕСЛИ ТЕКСТ СТРОКИ ОТСУТСТВУЕТ (ПУСТО), ТО СТРОКА ИГНОРИРУЕТСЯ. ЕСЛИ ОТСУТСТВУЕТ И ЧИСЛО И ТЕКСТ, ТО НА МОНИТОР ВЫВОДЯТСЯ ВСЕ СТРОКИ ТЕКСТА. КАЖДАЯ ВВЕДЕННАЯ СТРОКА ДОЛЖНА ОКОНЧИВАТЬСЯ ЗНАКОМ ENTER, А ТЕКСТ - ЗНАКОМ EOF (SS + I).

```
#INCLUDE "STDIO.H"
typedef struct STR{
 struct STR *NEXT
 int LAB
}LIST
typedef LIST *LISTPTR
typedef char *CHARPTR
LIST HEAD = [NULL]

MAIN ()
[
 EXTERN LIST *FIND()
 int NUM, DEL
 char LINE[81]
 LIST *POS
 NUM = 0
 DO [:
 IF (INUM)
 DISPLAY()
 ELSE [
 POS = FIND(NUM)
 IF (POS->NEXT && POS->NEXT->LAB == NUM)
 DELETE (POS)
 IF (DEL != 'N') [
 GETLINE (LINE, 81)
 INSERT (POS, NUM, LINE)
]
]
 GETNUMBER (&NUM, &DEL)
]

LIST*
FIND (NUM)
 int NUM
[
 LIST *REF
 REF = &HEAD
 WHILE (REF->NEXT && REF->NEXT->LAB < NUM)
 REF = REF->NEXT
 RETURN REF
[
DELETE (POS)
 LIST *POS
[
 LIST *PTR
 PTR = POS->NEXT
```



```
 POS->NEXT = PTR->NEXT
 FREE(CAST(CHARPTR)PTR)
]

GETLINE (PTR, MAX)
 CHAR *PTR
 INT MAX
[
 INT I, C
 FOR (I=0
 I < MAX-1&&(C=GETCHAR()) != ' N'
 I++) *PTR++ = C
 IF (C != ' N')
 WHILE (GETCHAR() != ' N')
 *PTR = ' 0'
]

INT
LENGTH (LINE)
 CHAR *LINE
[
 INT LEN
 LEN=0
 WHILE (*LINE++)
 LEN++
 RETURN LEN
]

INSERT (POS, NUM, LINE)
 LIST *POS
 INT NUM
 CHAR LINE[]
[
 LIST *PTR, *REF
 INT SIZE
 SIZE=SIZEOF(LISTPTR)+SIZEOF(INT)+LENGTH(LINE)+1
 IF (PTR=CAST(LISTPTR)CALLOF(1,SIZE)) [
 REF=POS->NEXT
 POS->NEXT=PTR
 PTR->NEXT=REF
 PTR->LAB=NUM

 FILL(CAST(CHARPTR)(PTR+1),LINE)
] ELSE
 PRINTF ("ZIGNOROWANO:-BRAK PAMIECI")
]

DISPLAY ()
[
 LIST *REF
 REF=HEAD.NEXT
 WHILE (REF) [
 PRINTF ("%3D%S N", REF->LAB, CAST(CHARPTR)(REF+1))
 REF=REF->NEXT
]
]

FILL (REF, LINE)
 CHAR *REF, *LINE
```

```
[
 WHILE (*REF++==*LINE++)
 *REF='0'
]

GETNUMBER (NUM, DEL)
 INT *NUM, *DEL
[
 INT C, VAL
 VAL=0
 WHILE ((C=GETCHAR())>='0'&&C<='9')
 VAL=10*VAL+C-'0'
 *NUM=VAL
 *DEL=C
]
```

#INCLUDE ? STDIO.LIB ?

9. СОЗДАТЬ ПРОГРАММУ ДЛЯ ВЫВОДА НА ЭКРАН СОДЕРЖИМОГО ФАЙЛА TEXT.DOC:

```
#DEFINE EOF (-1)
TYPEDEF INT FILE
MAIN ()
[
 INT CHR
 FILE *INPUT, *FOPEN()
 IF ((INPUT=FOPEN("TEXT.DOC", "R"))!=NULL)
 WHILE (C=GETC(INPUT) != EOF)
 PUTCHAR (CHR)
]
```

10. ПРОГРАММА КОПИРОВАНИЯ СТРОКИ С КЛАВИАТУРЫ НА ПРИНТЕР:

```
TYPEDEF INT FILE
#DEFINE EOF (-1)
MAIN ()
[
 FILE *STREAM
 INT CHR
 WHILE ((CHR=GETCHAR() != EOF)
 PUTC (CHR, CAST(STREAM)3)
]
```

## ПРИЛОЖЕНИЕ В. РЕДАКТОР HISOFT

ДИРЕКТИВЫ РЕДАКТОРА ИМЕЮТ ВИД:

DN1, N2, T1, T2

- ГДЕ D - ОДНОБУКВЕННАЯ ДИРЕКТИВА, N1 И N2 - ЦЕЛЫЕ ЧИСЛА 1 ... 32767, T1 И T2 - ЗАПИСИ НЕ БОЛЕЕ 20 СИМВОЛОВ КАЖДАЯ. НЕКОТОРЫЕ ПАРАМЕТРЫ МОГУТ БЫТЬ ОПУЩЕНЫ. В ЭТОМ СЛУЧАЕ ОНИ ПРИНИМАЮТ ЗНАЧЕНИЕ СООТВЕТСТВУЮЩИХ ПАРАМЕТРОВ ПРЕДЫДУЩЕЙ ДИРЕКТИВЫ.

ЕСЛИ СИСТЕМА HISOFT НАХОДИТСЯ В РЕЖИМЕ КОМПИЛЯЦИИ, ТО ДЛЯ ВЫЗОВА РЕДАКТОРА НЕОБХОДИМО НАЖАТЬ EDIT (CS+1), А ЗАТЕМ КЛАВИШУ ENTER. ДЛЯ ОКОНЧАНИЯ РЕДАКТИРОВАНИЯ И ВЫЗОВ КОМПИЛЯТОРА ИСПОЛЬЗУЕТСЯ ДИРЕКТИВА C, А ДЛЯ ВЫХОДА В BASIC - ДИРЕКТИВА B.

-1ДИРЕКТИВА I (INSERT) IM,N -0  
ДИРЕКТИВА СЛУЖИТ ДЛЯ ИНИЦИИИ АВТОМАТИЧЕСКОЙ НУМЕРАЦИИ СТРОК ТЕКСТА. НЕПОСРЕДСТВЕННО ПОСЛЕ ЕЕ ВЫПОЛНЕНИЯ НА ЭКРАНЕ ПОЯВИТСЯ ЧИСЛО M. ПОСЛЕ НАЖАТИЯ ENTER ВВОД СТРОКИ С ЭТИМ НОМЕРОМ ЗАКАНЧИВАЕТСЯ, А НА ЭКРАНЕ В НОВОЙ СТРОКЕ ПОЯВИТСЯ ЧИСЛО M+N, СООТВЕТСТВУЮЩЕЕ НОМЕРУ СЛЕДУЮЩЕЙ СТРОКИ. ДЛЯ ПРЕКРАЩЕНИЯ ДЕЙСТВИЯ ДИРЕКТИВЫ I НАЖМИТЕ EDIT.

-1ДИРЕКТИВА L (LIST) LM,N -0  
ДИРЕКТИВА СЛУЖИТ ДЛЯ ВЫВОДА НА ЭКРАН СТРОК ТЕКСТА N ... M. ПО УМОЛЧАНИЮ M=1, А N=32767. ЕСЛИ ВСЬ ТЕКСТ НЕ УМЕЩАЕТСЯ НА ЭКРАНЕ, ТО ОН ВЫВОДИТСЯ ПОРЦИЯМИ ПО 10 СТРОК. РАЗМЕР ПОРЦИИ ВЫВОДА МОЖЕТ БЫТЬ ИЗМЕНЕН ДИРЕКТИВОЙ K. ДЛЯ ВЫХОДА ИЗ ДИРЕКТИВЫ НЕОБХОДИМО НАЖАТЬ EDIT.

-1ДИРЕКТИВА K (CHUNK) KM -0  
ДИРЕКТИВА УСТАНОВЛИВАЕТ ПОРЦИЮ ВЫВОДА ИЗ M СТРОК ДЛЯ ДИРЕКТИВЫ L.

-1ДИРЕКТИВА W (WRITE) WM,N -0  
ВЫВОД НА ПРИНТЕР. ДЛЯ ПРЕРЫВАНИЯ ПЕЧАТИ - BREAK.

-1ДИРЕКТИВА S (SET) S,,D -0  
ИЗМЕНЕНИЕ СЕПАРАТОРА АРГУМЕНТОВ ДИРЕКТИВ. ПО УМОЛЧАНИЮ СЕПАРАТОРОМ ЯВЛЯЕТСЯ ЗАПЯТАЯ, НО МОЖНО УСТАНОВИТЬ ЛЮБОЙ ДРУГОЙ ЗНАК.

-1ДИРЕКТИВА V (VIEW) LV -0  
ВЫВОД НА ЭКРАН АРГУМЕНТОВ N1, N2, T1, T2. ИСПОЛЬЗОВАННЫХ В ПОСЛЕДНЕЙ ВЫПОЛНЕННОЙ ДИРЕКТИВЕ ВЫВОД АДРЕСОВ НАЧАЛА И КОНЦА БУФЕРА ТЕКСТА.

-1ДИРЕКТИВА D (DELETE) DM,N -0  
УДАЛЕНИЕ СТРОК ОТ M ДО N. D10,10 ИЛИ D10 - УДАЛЕНИЕ ТОЛЬКО 10-ОЙ СТРОКИ.

-1ДИРЕКТИВА N (RENUMBER) NM,N -0  
ПЕРЕНОМЕРАЦИЯ: ПЕРВАЯ СТРОКА ИМЕЕТ НОМЕР M, ПОСЛЕДУЮЩИЕ N+M

-1ДИРЕКТИВА P (PUT) PM,N,T -0  
ЗАПИСЬ ВО ВНЕШНЮЮ ПАМЯТЬ, КАК В ФАЙЛ С ИМЕНЕМ T, СТРОК ПРОГРАММЫ ОТ M ДО N.

-1ДИРЕКТИВА G (GET) G,,T -0  
ВВОД ИЗ ВНЕШНЕЙ ПАМЯТИ КАК ИЗ ФАЙЛА С ИМЕНЕМ T ТЕКСТА ПРОГРАММЫ. ВВЕДЕННЫЙ ТЕКСТ РАЗМЕЩАЕТСЯ НЕПОСРЕДСТВЕННО ЗА УЖЕ ОТРЕДАКТИРОВАННЫМ, А ЕГО СТРОКИ АВТОМАТИЧЕСКИ НУМЕРУЮТСЯ С ДИСКРЕТНОСТЬЮ 10.

-1ДИРЕКТИВА C (COMPILER) C -0

-1ДИРЕКТИВА B (BASIC) B -0  
ДЛЯ ВОЗВРАТА В СИ - PRINT USR 25200.

-1ДИРЕКТИВА F (FIND) FM,N,F,S -0  
ПОИСК В ОТРЕЗКЕ M...N СТРОКИ С НАИМЕНЬШИМ НОМЕРОМ, СОДЕРЖАЩЕЙ F. ПОСЛЕ ЭТОГО ВЫПОЛНЯЕТСЯ ДИРЕКТИВА E. НАЙДЕННЫЙ ТЕКСТ МОЖЕТ БЫТЬ ЗАМЕНЕН НА ТЕКСТ S, ЕСЛИ ИСПОЛЬЗОВАТЬ ПОДДИРЕКТИВУ S.

-1 ДИРЕКТИВА E (EDIT) ЕМ -2

НЕЯВНОЕ ВЫПОЛНЕНИЕ ЭТОЙ ДИРЕКТИВЫ ПРОИСХОДИТ ПРИ КОМПИЛЯЦИИ ПРОГРАММЫ, СОДЕРЖАЩЕЙ ОШИБОЧНУЮ СТРОКУ M. В ЭТОМ СЛУЧАЕ НА ЭКРАН ВЫВОДИТСЯ СООБЩЕНИЕ ОБ ОШИБКЕ, А ПО НАЖАТИЕ EDIT НЕЯВНО ВЫПОЛНЯЕТСЯ ДИРЕКТИВА ЕМ. ПРИ НАЖАТИИ ЛЮБОЙ ДРУГОЙ КЛАВИШИ ВЫЗЫВАЕТСЯ КОМПИЛЯТОР. РЕДАКТИРОВАНИЕ СТРОКИ ОСУЩЕСТВЛЯЕТСЯ С ПОМОЩЬЮ ОДНОЗНАЧНЫХ ПОДДИРЕКТИВ.

\* ПОДДИРЕКТИВА => (CS+8) - ЕСЛИ К МОМЕНТУ ИСПОЛЬЗОВАНИЯ ЭТОЙ ПОД ДИРЕКТИВЫ ВСЯ ИТОГОВАЯ СТРОКА БЫЛА ВЫВЕДЕНА НА ЭКРАН, ТО К СТРОКЕ ДОБАВЛЯЕТСЯ ВВЕДЕННЫЙ СИМВОЛ, В ПРОТИВНОМ СЛУЧАЕ ВВОДИТСЯ СЛЕДУЮЩИЙ ЗНАК ИТОГОВОЙ СТРОКИ

\* ПОДДИРЕКТИВА <= (CS+5) - СТИРАНИЕ НА ЭКРАНЕ ПОСЛЕДНЕГО ЗНАКА ИТОГОВОЙ СТРОКИ И СДВИГ КУРСОРА ВПРАВО

\* ПОДДИРЕКТИВА ENTER - ПЕРЕНОС В ИТОГОВУЮ СТРОКУ ОСТАВШИХСЯ СИМВОЛОВ ИСХОДНОЙ, ОКОНЧАНИЕ РЕДАКТИРОВАНИЯ И ЗАМЕНА В ТЕКСТЕ ИСХОДНОЙ СТРОКИ НА ИТОГОВУЮ

\* ПОДДИРЕКТИВА Q (QUIT) - ОКОНЧАНИЕ РЕДАКТИРОВАНИЯ БЕЗ ИЗМЕНЕНИЯ ИСХОДНОЙ СТРОКИ

\* ПОДДИРЕКТИВА R (REPLACE) - ИГНОРИРОВАНИЕ ИТОГОВОЙ СТРОКИ И ПОВТОРЕНИЕ РЕДАКТИРОВАНИЯ

\* ПОДДИРЕКТИВА L (LIST) - ПЕРЕНОС В ИТОГОВУЮ СТРОКУ ОСТАВШИХСЯ СИМВОЛОВ ИСХОДНОЙ И УСТАНОВКА КУРСОРА НА НАЧАЛО ИТОГОВОЙ СТРОКИ, КАК-БУДТО ИТОГОВАЯ СТАНОВИТСЯ ИСХОДНОЙ, А НОВАЯ ИТОГОВАЯ - ПУСТОЙ

\* ПОДДИРЕКТИВА K (KILL) - ЕСЛИ НА ЭКРАН ВЫВЕДЕНА ВСЯ ИТОГОВАЯ СТРОКА, ТО ПРИ ВЫПОЛНЕНИИ ПОДДИРЕКТИВЫ УДАЛЯЕТСЯ БЛИЖАЙШИЙ СИМВОЛ ИСХОДНОЙ СТРОКИ, В ПРОТИВНОМ СЛУЧАЕ УДАЛЯЕТСЯ ЗНАК, ПОД КОТОРЫМ НАХОДИТСЯ КУРСОР В ИТОГОВОЙ СТРОКЕ

\* ПОДДИРЕКТИВА Z (ZEROIZE) - ПРОПУСК ВСЕХ СИМВОЛОВ ИТОГОВОЙ СТРОКИ, НАЧИНАЯ ОТ СИМВОЛА ПОД КУРСОРОМ И ВСЕХ ПОСЛЕДУЮЩИХ СИМВОЛОВ - ЕЩЕ НЕ РАССМОТРЕННЫХ - ИСХОДНОЙ

\* ПОДДИРЕКТИВА I (INSERT) - ИЗМЕНЕНИЕ ВИДА КУРСОРА С L НА \* И ПЕРЕХОД В РЕЖИМ, ПРИ КОТОРОМ СИМВОЛЫ, ВВЕДЕННЫЕ С КЛАВИАТУРЫ, ВВОДЯТСЯ НЕПОСРЕДСТВЕННО В ИТОГОВУЮ СТРОКУ. В ЭТОМ РЕЖИМЕ ОШИБКИ УСТРАНЯЮТСЯ DELETE. ВЫХОД ИЗ РЕЖИМА ПО НАЖАТИЮ ENTER.

\* ПОДДИРЕКТИВА X (EXTERNAL) - ПЕРЕНОС В ИТОГОВУЮ СТРОКУ ОСТАВШИХСЯ СИМВОЛОВ ИСХОДНОЙ, ВЫВОД НА ЭКРАН ВСЕЙ ИТОГОВОЙ СТРОКИ И НЕЯВНОЕ ВЫПОЛНЕНИЕ ПОДДИРЕКТИВЫ I

\* ПОДДИРЕКТИВА C (CHANG) - ИЗМЕНЕНИЕ ВИДА КУРСОРА С L НА + И ПЕРЕХОД В РЕЖИМ, ПРИ КОТОРОМ СИМВОЛЫ, ВВЕДЕННЫЕ С КЛАВИАТУРЫ, НЕПОСРЕДСТВЕННО ВВОДЯТСЯ В ИТОГОВУЮ СТРОКУ. ЕСЛИ В МОМЕНТ ВВОДА ТАКОГО СИМВОЛА НА ЭКРАН ВЫВЕДЕНА ВСЯ ИТОГОВАЯ СТРОКА, ТО ВЕДЕННЫЕ СИМВОЛЫ РАЗМЕЩАЮТСЯ В КОНЦЕ СТРОКИ, А СЛЕДУЮЩИЙ ЗНАК ИСХОДНОЙ СТРОКИ ОПУСКАЕТСЯ. В ПРОТИВНОМ СЛУЧАЕ ВВОДИМЫЙ СИМВОЛ ЗАМЕЩАЕТ НАХОДЯЩИЙСЯ ПОД КУРСОРОМ СИМВОЛ ИСХОДНОЙ СТРОКИ И НЕЯВНО ВЫПОЛНЯЕТСЯ ПОДДИРЕКТИВА =>. ВЫХОД ИЗ ЭТОГО РЕЖИМА ПО ENTER.

\* ПОДДИРЕКТИВА F (FINAL) - ПОДДИРЕКТИВА ВЫПОЛНЯЕТСЯ ПОСЛЕ НЕЯВНОГО ВЫПОЛНЕНИЯ ДИРЕКТИВЫ E, ВЫЗВАННОГО ВЫПОЛНЕНИЕМ ДИРЕКТИВЫ F. ПРИ ЭТОМ ОРГАНИЗУЕТСЯ ПОИСК СЛЕДУЮЩЕГО УЧАСТКА, СОДЕРЖАЩЕГО ТЕКСТ F, КОТОРЫЙ БЫЛ УКАЗАН В ПРЕДВАРИТЕЛЬНО ВЫПОЛНЕННОЙ ДИРЕКТИВЕ F. СЛЕДУЕТ ПОМНИТЬ, ЧТО ПОИСК ТЕКСТА ОСУЩЕСТВЛЯЕТСЯ В ГРАНИЦАХ, ОПРЕДЕЛЕННЫХ В ДИРЕКТИВЕ F

\* ПОДДИРЕКТИВА S (SUBSTITUTE)

ЗАМЕНА НАЙДЕННОГО ТЕКСТА F ТЕКСТОМ S И НЕЯВНОЕ ВЫПОЛНЕНИЕ ПОДДИРЕКТИВЫ F. ПОИСК ТЕКСТА ДОЛЖЕН БЫТЬ ИНИЦИИРОВАН ВЫПОЛНЕНИЕМ ДИРЕКТИВЫ ИЛИ ПОДДИРЕКТИВЫ F.

ПРИЛОЖЕНИЕ С. НЕКОТОРЫЕ БИБЛИОТЕЧНЫЕ ФУНКЦИИ  
АРИФМЕТИКА, ЗНАКОВЫЕ ОПЕРАЦИИ И КОНВЕРСИИ

ФУНКЦИЯ MAX (STDIO.H)  
INT MAX (COUNT) AUTO  
INT COUNT

ФУНКЦИЯ MIN (STDIO.H)  
INT MIN (COUNT) AUTO  
INT COUNT

ФУНКЦИЯ ABS (STDIO.LIB)  
INT ABS(P)  
INT P

ФУНКЦИЯ SIGN (STDIO.H)  
INT SIGN (P)  
INT P

РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ ДАННОЕ ВЕЛИЧИНОЙ -1, 0 ИЛИ 1 В ЗАВИСИМОСТИ ОТ ЗНАКА АРГУМЕНТА

ФУНКЦИЯ PEEK (STDIO.LIB)  
CHAR PEEK (ADDR)  
UNSIGNED ADDR

ВОЗВРАЩАЕТ ДАННОЕ ТИПА (CHAR), ХРАНЯЩЕЕСЯ ПО АДРЕСУ ADDR.

ФУНКЦИЯ POKE (STDIO.LIB)  
POKE (ADDR, CH)  
UNSIGNED ADDR  
INT CH

ФУНКЦИЯ TOLOWER (ВСТРОЕННАЯ)  
CHAR TOLOWER (C)  
CHAR C

РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ ДАННОЕ ТИПА (CHAR). ЕСЛИ C ЗАГЛАВНАЯ БУКВА, ТО РЕЗУЛЬТАТ - СООТВЕТСТВУЮЩАЯ МАЛАЯ. В ОСТАЛЬНЫХ СЛУЧАЯХ - СИМВОЛ, РАВНЫЙ C.

ФУНКЦИЯ TOUPPER (ВСТРОЕННАЯ)  
CHAR TOUPPER (C)  
CHAR C

ФУНКЦИЯ ОБРАТНАЯ TOLOWER.

ФУНКЦИЯ ATOI (STDIO.LIB)  
INT ATOI (STR)  
CHAR \*STR

ПРЕОБРАЗОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ, УКАЗАННЫХ STR В ДАННЫЕ ТИПА (INT). ЕСЛИ ПОСЛЕДОВАТЕЛЬНОСТЬ НАЧИНАЕТСЯ С ЦЕЛОГО ЧИСЛА, ТО РЕЗУЛЬТАТОМ ЯВЛЯЕТСЯ ЭТО ЧИСЛО, В ПРОТИВНОМ СЛУЧАЕ - 0.

СОРТИРОВКА

ФУНКЦИЯ QSORT (STDIO.LIB)  
QSORT (PTR, CNT, SIZE, FUN)  
CHAR \*PTR  
INT CNT, SIZE  
INT (\*FUN)()

СОРТИРОВКА ДАННЫХ, ЯВЛЯЮЩИХСЯ МАССИВОМ ИЗ CNT ЭЛЕМЕНТОВ, ПЕРВЫЙ БАЙТ КОТОРОГО УКАЗАН STR. РАЗМЕР ЭЛЕМЕНТОВ ОПРЕДЕЛЕН SIZE, А СРАВНИВАЮЩАЯ ФУНКЦИЯ - FUN. ФУНКЦИЯ, СВЯЗАННАЯ С ПАРАМЕТРОМ FUN, ДОЛЖНА БЫТЬ ПОДОБРАНА ТАК, ЧТОБЫ ДЛЯ ЕЕ ЭЛЕМЕНТОВ ELM1 И ELM2. РЕЗУЛЬТАТОМ ВЫЗОВА:

(\*FUN)(SELM1,SELM2)

БЫЛО ДАННОЕ ТИПА (INT) ВЕЛИЧИНОЙ -1,0 ИЛИ 1 В СООТВЕТСТ-  
ВИИ С:

-1 ЕСЛИ ELM1<ELM2  
0 ЕСЛИ ELM1=ELM2  
1 ЕСЛИ ELM1>ELM2

# ПРЕОБРАЗОВАНИЕ СИМВОЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ

ФУНКЦИЯ STRCAT (STDIO.LIB)

CHAR \*STRCAT (FIRST, SECOND)

CHAR \*FIRST, \*SECOND

УДЛИНЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ, УКАЗАННЫХ FIRST, НА ДЛИНУ ПОСЛЕДОВАТЕЛЬНОСТИ SECOND (Т.Е. ПРИСОЕДИНЕНИЕ SECOND К FIRST). РЕЗУЛЬТАТ ФУНКЦИИ - ДАННОЕ ТИПА (\*CHAR), УКАЗЫВАЮЩЕЕ НА ПЕРВЫЙ ЗНАК ПОСЛЕДОВАТЕЛЬНОСТИ FIRST.

ФУНКЦИЯ STRCMP (STDIO.LIB)

INT STRCMP (SRC, TRG)

CHAR \*SRC, \*TRG

СРАВНЕНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ЗНАКОВ, УКАЗАННЫХ SRC И TRG, И ДОСТУП К ДАННОМУ ТИПА (INT), ОПРЕДЕЛЯЮЩЕМУ РЕЗУЛЬТАТ СРАВНЕНИЯ. ДАННОЕ ИМЕЕТ ВЕЛИЧИНУ 0 -ЕСЛИ ПОСЛЕДОВАТЕЛЬНОСТИ РАВНЫ, ПОЛОЖИТЕЛЬНУЮ - ЕСЛИ ПЕРВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ БОЛЬШЕ ВТОРОЙ ИЛИ ОТРИЦАТЕЛЬНУЮ, ЕСЛИ ПЕРВАЯ МЕНЬШЕ ВТОРОЙ. СРАВНЕНИЕ ИДЕТ ДО КОНЦА ПОСЛЕДОВАТЕЛЬНОСТИ ИЛИ ДО ПЕРВОГО НЕСРАВНЕНИЯ.

ФУНКЦИЯ STRCPY (STDIO.LIB)

CHAR \*STRCPY (DST, SRC)

CHAR \*DST, \*SRC

КОПИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ, УКАЗАННОЙ SRC, В ОБЛАСТЬ ПАМЯТИ, УКАЗАННУЮ DST.

ФУНКЦИЯ STRLEN (STDIO.LIB)

UNSIGNED STRLEN (STR)

CHAR \*STR

ДОСТУП К ДАННОМУ ТИПА (INT) РАВНОМУ ЧИСЛУ ЗНАКОВ ПОСЛЕДОВАТЕЛЬНОСТИ, УКАЗАННОЙ STR.

# УПРАВЛЕНИЕ ОПЕРАТИВНОЙ ПАМЯТЬЮ

ФУНКЦИЯ CALLOC (STDIO.LIB)

CHAR \*CALLOC (N, SIZE)

UNSIGNED N, SIZE

ВЫДЕЛЕНИЕ ОБЛАСТИ ПАМЯТИ, ПРЕДНАЗНАЧЕННОЙ ДЛЯ РАЗМЕЩЕНИЯ N ОБЪЕКТОВ РАЗМЕРОМ SIZE КАЖДЫЙ И ДОСТУП К УКАЗАТЕЛЮ НА ЭТУ ОБЛАСТЬ. ЕСЛИ ВЫДЕЛЕНИЕ ОБЛАСТИ НЕВОЗМОЖНО, ТО РЕЗУЛЬТАТ - NULL.

ФУНКЦИЯ FREE (STDIO.LIB)

FREE (PTR)

CHAR \*PTR

ОСВОБОЖДЕНИЕ ОБЛАСТИ ПАМЯТИ, ВЫДЕЛЕННОЙ ФУНКЦИЕЙ CALLOC. АРГУМЕНТ ФУНКЦИИ ПРЕДСТАВЛЯЕТ СОБОЙ УКАЗАТЕЛЬ НА ЭТУ ОБЛАСТЬ.

ФУНКЦИЯ SWAP (ВСТРОЕННАЯ)

SWAP (PAREA, QAREA, LEN)

CHAR \*PAREA, \*QAREA

UNSIGNED LEN



ОБМЕН МЕСТАМИ СОДЕРЖИМОГО ДВУХ ОБЛАСТЕЙ ПАМЯТИ ДЛИНОЙ LEN БАЙТ, УКАЗАННЫХ PAREA И QAREA.

ФУНКЦИЯ MOVE (ВСТРОЕННАЯ)  
MOVE (DST, SRC, LEN)  
CHAR \*DST, \*SRC  
UNSIGNED LEN

КОПИРОВАНИЕ ОБЛАСТИ ПАМЯТИ ДЛИНОЙ LEN, УКАЗАННОЙ SRC, В ОБЛАСТЬ, УКАЗАННУЮ DST.

#### ОПЕРАЦИИ ВВОДА/ВЫВОДА

ФУНКЦИЯ GETS (STDIO.LIB)  
CHAR GETS (STR)  
CHAR \*STR

РАЗМЕЩЕНИЕ В ОБЛАСТИ ПАМЯТИ, УКАЗАННОЙ STR, ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ, ПОЛУЧАЕМОЙ ИЗ СТАНДАРТНОГО ФАЙЛА ВВОДА. ВЫПОЛНЕНИЕ ФУНКЦИИ ЗАКАНЧИВАЕТСЯ ПРИ ПОЛУЧЕНИИ ЗНАКА 'N' (ENTER). ВМЕСТО НЕГО В ЗУ ЗАПИСЫВАЕТСЯ ЗНАК С КОДОМ 0. РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ УКАЗАТЕЛЬ НА ПЕРВЫЙ ЗНАК ПОЛЯ ПАМЯТИ.

ФУНКЦИЯ PUTS (STDIO.LIB)  
PUTS (STR)  
CHAR \*STR

ВЫВОД В СТАНДАРТНЫЙ ФАЙЛ ВЫВОДА ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ, УКАЗАННЫХ STR. НЕОБХОДИМО, ЧТОБЫ ПОСЛЕДНИМ ЗНАКОМ БЫЛ ЗНАК С КОДОМ 0.

ФУНКЦИЯ FGETS (STDIO.LIB)  
CHAR \*FGETS (STR, N, INP)  
CHAR \*STR  
INT N  
FILE \*INP

РАЗМЕЩЕНИЕ В ПОЛЕ ПАМЯТИ, УКАЗАННОМ STR, НЕ БОЛЕЕ ЧЕМ N ЗНАКОВ, ПОЛУЧАЕМЫХ ИЗ ФАЙЛА, УКАЗАННОГО INP. ВЫПОЛНЕНИЕ ФУНКЦИИ ЗАКАНЧИВАЕТСЯ ПРИ РАЗМЕЩЕНИИ N-1 ЗНАКА ИЛИ ПРИ ПОЛУЧЕНИИ ЗНАКА 'N'. ПОСЛЕ ЭТОГО В ПОЛЕ РАЗМЕЩАЕТСЯ ДОПОЛНИТЕЛЬНЫЙ ЗНАК С КОДОМ 0. РЕЗУЛЬТАТОМ ФУНКЦИИ ЯВЛЯЕТСЯ УКАЗАТЕЛЬ НА ПЕРВЫЙ ЗНАК ПОЛЯ ИЛИ ЗНАК NULL, ЕСЛИ ФАЙЛ ПУСТОЙ.

ФУНКЦИЯ FPUTS (STDIO.LIB)  
FPUTS (STR, OUT)  
CHAR \*STR  
FILE \*OUT

ВЫВОД В ФАЙЛ, УКАЗАННЫЙ OUT, ПОСЛЕДОВАТЕЛЬНОСТИ ЗНАКОВ, УКАЗАННЫХ STR, ОКАНЧИВАЕМОЙ СИМВОЛОМ С КОДОМ 0.

#### ГЕНЕРАЦИЯ СЛУЧАЙНЫХ ЧИСЕЛ

ФУНКЦИЯ RAND (STDIO.LIB)  
INT RAND()  
ФУНКЦИЯ SRAND (S) (STDIO.LIB)  
INT S

ПЕРЕУСТАНОВКА ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ.

# ГРАФИКА И ЗВУК

ФУНКЦИЯ PLOT (STDIO.LIB)

PLOT (ON, X, Y)

INT ON, X, Y

ИНДИКАЦИЯ НА ЭКРАНЕ ПИКСЕЛЯ С КООРДИНАТАМИ X, Y И С ЦВЕТОМ PAPER  
ЕСЛИ ON=0, ИЛИ INK, ЕСЛИ ON=1.

ФУНКЦИЯ LINE (STDIO.LIB)

LINE (ON, DX, DY)

INT ON, DX, DY

ВЫЧЕРЧИВАНИЕ ОТРЕЗКА ПРЯМОЙ ОТ ТЕКУЩЕЙ ПОЗИЦИИ, УСТАНОВЛЕННОЙ  
ФУНКЦИЕЙ PLOT, ДО ТОЧКИ, УДАЛЕННОЙ НА DX, DY. ON ОПРЕДЕЛЯЕТ ЦВЕТ.

ФУНКЦИЯ PAPER (STDIO.LIB)

PAPER (COLOR)

INT COLOR

ФУНКЦИЯ INK (STDIO.LIB)

INK (COLOR)

INT COLOR

ФУНКЦИЯ CLS (STDIO.LIB)

CLS()

ФУНКЦИЯ BEEP (STDIO.LIB)

BEEP (TIME, TONE)

INT TIME, TONE

TONE - ЧАСТОТА В HZ, TIME - ДЛИТЕЛЬНОСТЬ 1/10 S.

## ПРИЛОЖЕНИЕ D. СООБЩЕНИЯ ОБ ОШИБКАХ

|          |                                             |                                                            |
|----------|---------------------------------------------|------------------------------------------------------------|
| ERROR-0  | MISSING 'X'                                 | ОТСУТСТВИЕ ОЖИДАЕМОГО ЗНАКА                                |
| ERROR-1  | RESTRICTION: NOT IMPLEMENTED                | НЕИЗВЕСТНЫЙ ТИП ДАННЫХ                                     |
| ERROR-2  | BAD CHARACTER CONSTANT                      | ОШИБКА В СИМВОЛЬНОМ ЛИТЕРАЛЕ                               |
| ERROR-3  | NOT A PREPROCESSOR COMMAND                  | ОШИБОЧНАЯ ДИРЕКТИВА ПРЕПРОЦЕССОРА                          |
| ERROR-4  | LIMIT: MACRO BUFFER FULL                    | ОПРЕДЕЛЕНИЕ ОТМЕНЯЕТ САМО СЕБЯ                             |
| ERROR-5  | CAN ONLY DEFINE IDENTIFIERS AS MACROS       | ОШИБОЧНАЯ ДИРЕКТИВА ПРЕПРОЦЕССОРА                          |
| ERROR-6  | RESTRICTION: MACROS MAY NOT HAVE PARAMETERS | ОШИБОЧНАЯ ДИРЕКТИВА ПРЕПРОЦЕССОРА                          |
| ERROR-7  | CANNOT OPEN FILE                            | ОТСУТСТВИЕ ФАЙЛА С ДАННЫМ ИМЕНЕМ                           |
| ERROR-8  | RESTRICTION: CANNOT NEST INCLUDES           | ВКЛЮЧЕННЫЙ ФАЙЛ СОДЕРЖИТ ДИРЕКТИВУ ВКЛЮЧЕНИЯ ДРУГОГО ФАЙЛА |
| ERROR-9  | MISSING WHILE                               | ОШИБКА В КОМАНДЕ DO                                        |
| ERROR-10 | NOT IN LOOP OR SWITCH                       | НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ КОМАНДЫ BREAK                   |
| ERROR-11 | NOT IN LOOP                                 | НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ КОМАНДЫ CONTINUE                |

|          |                                                                                             |
|----------|---------------------------------------------------------------------------------------------|
| ERROR-12 | NOT IN SWITCH<br>КОМАНДЫ CASE ИЛИ DEFAULT НЕПРАВИЛЬНО ИСПОЛЬЗОВАНЫ                          |
| ERROR-13 | LIMIT: TOO MANY CASE STATEMENTS<br>СЛИШКОМ МНОГО CASE                                       |
| ERROR-14 | MULTIPLE DEFAULT STATEMENTS<br>БОЛЕЕ ЧЕМ ОДИН DEFAULT                                       |
| ERROR-15 | GOTO NEEDS A LABEL<br>ОТСУТСТВИЕ МЕТКИ В КОМАНДЕ GOTO                                       |
| ERROR-16 | MULTIPLE USE OF IDENTIFIER<br>СОВПАДЕНИЕ МЕТКИ И ИДЕНТИФИКАТОРА                             |
| ERROR-17 | DIRECT EXECUTION NOT POSSIBLE<br>НЕКОРРЕКТНОЕ ИСПОЛЬЗОВАНИЕ ДИРЕКТИВЫ ПРЕПРОЦЕССОРА         |
| ERROR-18 | LIMIT: NAME TABLE FULL<br>СЛИШКОМ МНОГО ТИПОВ ДАННЫХ                                        |
| ERROR-19 | LIMIT: TOO MANY TYPES<br>СЛИШКОМ МНОГО ТИПОВ ДАННЫХ                                         |
| ERROR-20 | DUPLICATE DECLARATION - TYPE<br>ДВОЙНОЕ ОБЪЯВЛЕНИЕ                                          |
| ERROR-21 | DUPLICATE DECLARATION - STORAGE CLASS<br>ДВОЙНОЕ ОБЪЯВЛЕНИЕ                                 |
| ERROR-22 | LIMIT: GLOBAL SYMBOL TABLE FULL                                                             |
| ERROR-23 | LIMIT: TOO MUCH GLOBAL DATA<br>СЛИШКОМ МНОГО ПЕРЕМЕННЫХ ВНЕ ОБЪЯВЛЕНИЯ ФУНКЦИИ              |
| ERROR-24 | DUPLICATE DECLARATION<br>ДВОЙНОЕ ОБЪЯВЛЕНИЕ                                                 |
| ERROR-25 | LIMIT: LOCAL SYMBOL TABLE FULL<br>СЛИШКОМ МНОГО ПЕРЕМЕННЫХ В ФУНКЦИИ                        |
| ERROR-26 | THIS VARIABLE WAS NOT IN PARAMETER LIST<br>ПЕРЕМЕННАЯ НЕ БЫЛА УПОМЯНУТА В СПИСКЕ ПАРАМЕТРОВ |
| ERROR-27 | UNDEFINED VARIABLE                                                                          |
| ERROR-28 | BAD FUNCTION RETURN TYPE<br>ОШИБОЧНОЕ ВЫРАЖЕНИЕ В КОМАНДЕ RETURN                            |
| ERROR-29 | NO ARRAYS OF FUNCTIONS<br>ИСПОЛЬЗОВАНЫ МАССИВЫ ФУНКЦИЙ                                      |
| ERROR-30 | LIMIT: EXPRESSION TOO COMPLICATED                                                           |
| ERROR-31 | LIMIT: EXPRESSION TOO COMPLICATED<br>СЛИШКОМ СЛОЖНОЕ ВЫРАЖЕНИЕ                              |
| ERROR-32 | BAD TYPE COMBINATION<br>НЕКОРРЕКТНЫЙ АРГУМЕНТ ОПЕРАЦИИ                                      |
| ERROR-33 | BAD OPERAND TYPE<br>НЕКОРРЕКТНЫЙ АРГУМЕНТ ОПЕРАЦИИ                                          |
| ERROR-34 | NEED AN LVALUE<br>ИСПОЛЬЗОВАНО НЕ L-ВЫРАЖЕНИЕ                                               |
| ERROR-35 | NOT A DEFINED MEMBER OF A STRUCTURE<br>ИМЯ НЕ ИДЕНТИФИЦИРУЕТ ПОЛЯ                           |
| ERROR-36 | EXPECTED A PRIMARY HERE<br>ИСПОЛЬЗОВАНО НЕ ПЕРВИЧНОЕ ВЫРАЖЕНИЕ                              |
| ERROR-37 | UNDEFINED VARIABLE<br>ОБРАЩЕНИЕ К НЕ ОБЪЯВЛЕННОЙ ПЕРЕМЕННОЙ                                 |
| ERROR-38 | NEED A TYPE NAME<br>НЕ ИСПОЛЬЗОВАНО ИМЯ ТИПА                                                |
| ERROR-39 | NEED A CONSTANT EXPRESSION<br>ИСПОЛЬЗОВАНО НЕ КОНСТАНТНОЕ ВЫРАЖЕНИЕ                         |
| ERROR-40 | CAN ONLY CALL FUNCTIONS<br>В ВЫЗОВЕ НЕ ИСПОЛЬЗОВАНО ИМЯ ФУНКЦИИ                             |
| ERROR-41 | DOES NOT FOLLOW ? PROPERLY<br>НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ 3Х-АРГУМЕНТНОГО ОПЕРАТОРА          |
| ERROR-42 | DESTINATION MUST BE AN LVALUE<br>ИСПОЛЬЗОВАНО НЕ L-ВЫРАЖЕНИЕ                                |

|             |                                                                                       |
|-------------|---------------------------------------------------------------------------------------|
| ERROR-43    | NEED A:TO FOLLOW ?<br>НЕПРАВИЛЬНОЕ ИСПОЛЬЗОВАНИЕ 3Х-АРГУМЕНТНОГО ОПЕРАТОРА            |
| ERROR-44    | NEED A POINTER<br>НЕ ИСПОЛЬЗОВАН УКАЗАТЕЛЬ                                            |
| ERROR-45    | ILLEGAL PARAMETR TYPE<br>НЕКОРРЕКТНЫЙ ПАРАМЕТР ФУНКЦИИ                                |
| ERROR-46    | RESTRICTION: NOT IMPLEMENTED<br>НЕИЗВЕСТНЫЙ ТИП ДАННЫХ                                |
| ERROR-47    | CANNOT USE THIS OPERATOR<br>НЕИЗВЕСТНЫЙ ТИП ДАННЫХ                                    |
| ERROR-48    | BAD DECLARATION<br>ОШИБКА В ОБ ЯВЛЕНИИ                                                |
| ERROR-49    | STORAGE CLASS NOT VALID<br>ОШИБКА В ОБ ЯВЛЕНИИ                                        |
| ERROR-50    |                                                                                       |
| ERROR-51    | DUPLICATE DECLARATION OF STRUCTURE TAG<br>ДВОЙНАЯ ДЕКЛАРАЦИЯ ПРИЗНАКА                 |
| ERROR-52    | USE A PREDECLARED STRUCTURE FOR PARAMETERS<br>ОБ ЯВИ СТРУКТУРНЫЙ ТИП ПОСЛЕ ФУНКЦИИ    |
| ERROR-53    | STRUCTURE CANNOT CONTAIN ITSELF<br>СТРУКТУРА НЕ МОЖЕТ СОДЕРЖАТЬ САМУ СЮБЯ             |
| ERROR-54    | BAD DECLARATOR<br>ОШИБКА В ОБ ЯВЛЕНИИ                                                 |
| ERROR-55    | MISSING ) IN FUNCTION DECLARATION<br>ОТСУТСТВИЕ ( ) В ОБ ЯВЛЕНИИ ФУНКЦИИ              |
| ERROR-56    | BAD FORMAL PARAMETER LIST<br>ОШИБКА В ОБ ЯВЛЕНИИ ФУНКЦИИ                              |
| ERROR-57    | TYPE SHOULD BE FUNCTION<br>НЕИЗВЕСТНЫЙ ТИП РЕЗУЛЬТАТА ФУНКЦИИ                         |
| ERROR-58    |                                                                                       |
| ERROR-59    |                                                                                       |
| ERROR-60    | LIMIT: NO MORE MEMORY                                                                 |
| ERROR-61    | RESTRICTION: USE ASSIGNMENT TO INITIALIZE<br>НЕКОРРЕКТНОЕ ПРИСВОЕНИЕ НАЧАЛЬНЫХ ДАННЫХ |
| ERROR-62-63 | CANNOT INITIALIZE THIS<br>НЕКОРРЕКТНОЕ ПРИСВОЕНИЕ НАЧАЛЬНЫХ ДАННЫХ                    |
| ERROR-64    | TOO MUCH INITIALIZATION DATA<br>СЛИШКОМ МНОГО НАЧАЛЬНЫХ ДАННЫХ.                       |

КОДИРОВКА СИМВОЛОВ И КОМАНД

КОДИРОВКА СИМВОЛОВ И КОМАНД



| КОД | СИМВОЛ                     | HEX | ОДНОБАЙТОВАЯ | КОМАНДА Z80 | ПОСЛЕ СВН | ПОСЛЕ EDH |
|-----|----------------------------|-----|--------------|-------------|-----------|-----------|
| 0   |                            | 00  | NOP          | RLC B       |           |           |
| 1   |                            | 01  | LD BC, NN    | RLC C       |           |           |
| 2   |                            | 02  | LD (BC), A   | RLC D       |           |           |
| 3   |                            | 03  | INC BC       | RLC E       |           |           |
| 4   |                            | 04  | INC B        | RLC H       |           |           |
| 5   |                            | 05  | DEC B        | RLC L       |           |           |
| 6   | PRINT<br>PRZECINEK<br>EDIT | 06  | LD B, N      | RLC (HL)    |           |           |
| 7   |                            | 07  | RLC A        | RLC A       |           |           |
| 8   |                            | 08  | EX AF, AF'   | RRC B       |           |           |
| 9   |                            | 09  | ADD HL, BC   | RRC C       |           |           |
| 10  |                            | 0A  | LD A, (BC)   | RRC D       |           |           |
| 11  |                            | 0B  | DEC BC       | RRC E       |           |           |
| 12  | DELETE                     | 0C  | INC C        | RRC H       |           |           |
| 13  | ENTER                      | 0D  | DEC C        | RRC L       |           |           |
| 14  |                            | 0E  | LD C, N      | RRC (HL)    |           |           |
| 15  |                            | 0F  | RRC A        | RRC A       |           |           |
| 16  | INK                        | 10  | DJNZ DIS     | RL B        |           |           |
| 17  | PAPER                      | 11  | LD DE, NN    | RL C        |           |           |
| 18  | FLASH                      | 12  | LD (DE), A   | RL D        |           |           |
| 19  | BRIGHT                     | 13  | INC DE       | RL E        |           |           |
| 20  | INVERSE                    | 14  | INC D        | RL H        |           |           |
| 21  | OVER                       | 15  | DEC D        | RL L        |           |           |
| 22  | AT                         | 16  | LD D, N      | RL (HL)     |           |           |
| 23  | TAB                        | 17  | RLA          | RL A        |           |           |
| 24  |                            | 18  | JR DIS       | RR B        |           |           |
| 25  |                            | 19  | ADD HL, DE   | RR C        |           |           |
| 26  |                            | 1A  | LD A, (DE)   | RR D        |           |           |
| 27  |                            | 1B  | DEC DE       | RR E        |           |           |
| 28  |                            | 1C  | INC E        | RR H        |           |           |
| 29  |                            | 1D  | DEC E        | RR L        |           |           |
| 30  |                            | 1E  | LD E, N      | RR (HL)     |           |           |
| 31  |                            | 1F  | RRA          | RR A        |           |           |
| 32  | SPACJA                     | 20  | JR NZ, DIS   | SLA B       |           |           |
| 33  |                            | 21  | LD HL, NN    | SLA C       |           |           |
| 34  | "                          | 22  | LD (NN), HL  | SLA D       |           |           |
| 35  | #                          | 23  | INC HL       | SLA E       |           |           |
| 36  | x                          | 24  | INC H        | SLA H       |           |           |
| 37  | %                          | 25  | DEC H        | SLA L       |           |           |
| 38  | &                          | 26  | LD H, N      | SLA (HL)    |           |           |
| 39  | '                          | 27  | DAA          | SLA A       |           |           |
| 40  | (                          | 28  | JR Z, DIS    | SRA B       |           |           |
| 41  | )                          | 29  | ADD HL, HL   | SRA C       |           |           |
| 42  | *                          | 2A  | LD HL, (NN)  | SRA D       |           |           |
| 43  | +                          | 2B  | DEC NL       | SRA E       |           |           |
| 44  | ,                          | 2C  | INC L        | SRA H       |           |           |
| 45  | -                          | 2D  | DEC L        | SRA L       |           |           |
| 46  | .                          | 2E  | LD L, N      | SRA (HL)    |           |           |
| 47  | /                          | 2F  | CPL          | SRA A       |           |           |
| 48  | 0                          | 30  | JR NC, DIS   |             |           |           |
| 49  | 1                          | 31  | LD SP, NN    |             |           |           |
| 50  | 2                          | 32  | LD (NN), A   |             |           |           |
| 51  | 3                          | 33  | INC SP       |             |           |           |

|     |   |    |           |            |            |
|-----|---|----|-----------|------------|------------|
| 52  | 4 | 34 | INC (HL)  |            |            |
| 53  | 5 | 35 | DEC (HL)  |            |            |
| 54  | 6 | 36 | LD (HL),N |            |            |
| 55  | 7 | 37 | SCF       |            |            |
| 56  | 8 | 38 | JR C,DIS  | SRL B      |            |
| 57  | 9 | 39 | ADD HL,SP | SRL C      |            |
| 58  | : | 3A | LD A,(NN) | SRL D      |            |
| 59  |   | 3B | DEC SP    | SRL E      |            |
| 60  | < | 3C | INC A     | SRL H      |            |
| 61  | = | 3D | DEC A     | SRL L      |            |
| 62  | > | 3E | LD A,N    | SRL (HL)   |            |
| 63  | ? | 3F | CCF       | SRL A      |            |
| 64  | @ | 40 | LD B,B    | BIT 0,B    | IN B,(C)   |
| 65  | A | 41 | LD B,C    | BIT 0,C    | OUT (C),B  |
| 66  | B | 42 | LD B,D    | BIT 0,D    | SBC HL,BC  |
| 67  | C | 43 | LD B,E    | BIT 0,E    | LD (NN),BC |
| 68  | D | 44 | LD B,H    | BIT 0,H    | NEG        |
| 69  | E | 45 | LD B,L    | BIT 0,L    | RETN       |
| 70  | F | 46 | LD B,(HL) | BIT 0,(HL) | IM 0       |
| 71  | G | 47 | LD B,A    | BIT 0,A    | LD I,A     |
| 72  | H | 48 | LD C,B    | BIT 1,B    | IN C,(C)   |
| 73  | I | 49 | LD C,C    | BIT 1,C    | OUT (C),C  |
| 74  | J | 4A | LD C,D    | BIT 1,D    | ADC HL,BC  |
| 75  | K | 4B | LD C,E    | BIT 1,E    | LD BC,(NN) |
| 76  | L | 4C | LD C,H    | BIT 1,H    |            |
| 77  | M | 4D | LD C,L    | BIT 1,L    | RETI       |
| 78  | N | 4E | LD C,(HL) | BIT 1,(HL) |            |
| 79  | O | 4F | LD C,A    | BIT 1,A    | LD R,A     |
| 80  | P | 50 | LD D,B    | BIT 2,B    | IN D,(C)   |
| 81  | Q | 51 | LD D,C    | BIT 2,C    | OUT (C),D  |
| 82  | R | 52 | LD D,D    | BIT 2,D    | SBC HL,DE  |
| 83  | S | 53 | LD D,E    | BIT 2,E    | LD (NN),DE |
| 84  | T | 54 | LD D,H    | BIT 2,H    |            |
| 85  | U | 55 | LD D,L    | BIT 2,L    |            |
| 86  | V | 56 | LD D,(HL) | BIT 2,(HL) | IM 1       |
| 87  | W | 57 | LD D,A    | BIT 2,A    | LD A,I     |
| 88  | X | 58 | LD E,B    | BIT 3,B    | IN E,(C)   |
| 89  | Y | 59 | LD E,C    | BIT 3,C    | OUT (C),E  |
| 90  | Z | 5A | LD E,D    | BIT 3,D    | ADC HL,DE  |
| 91  | [ | 5B | LD E,E    | BIT 3,E    | LD DE,(NN) |
| 92  | / | 5C | LD E,H    | BIT 3,H    |            |
| 93  | ] | 5D | LD E,L    | BIT 3,L    |            |
| 94  |   | 5E | LD E,(HL) | BIT 3,(HL) | IM 2       |
| 95  | - | 5F | LD E,A    | BIT 3,A    | LD A,R     |
| 96  |   | 60 | LD H,B    | BIT 4,B    | IN H,(C)   |
| 97  | A | 61 | LD H,C    | BIT 4,C    | OUT (C),H  |
| 98  | B | 62 | LD H,D    | BIT 4,D    | SBC HL,HL  |
| 99  | C | 63 | LD H,E    | BIT 4,E    | LD (NN),HL |
| 100 | D | 64 | LD H,H    | BIT 4,H    |            |
| 101 | E | 65 | LD H,L    | BIT 4,L    |            |
| 102 | F | 66 | LD H,(HL) | BIT 4,(HL) |            |
| 103 | G | 67 | LD H,A    | BIT 4,A    | RRD        |
| 104 | H | 68 | LD L,B    | BIT 5,B    | IN L,(C)   |
| 105 | I | 69 | LD L,C    | BIT 5,C    | OUT (C),L  |
| 106 | J | 6A | LD L,D    | BIT 5,D    | ADC HL,HL  |
| 107 | K | 6B | LD L,E    | BIT 5,E    | LD HL,(NN) |
| 108 | L | 6C | LD L,H    | BIT 5,H    |            |

|     |     |    |            |            |            |
|-----|-----|----|------------|------------|------------|
| 109 | M   | 6D | LD L,L     | BIT 5,L    |            |
| 110 | N   | 6E | LD L,(HL)  | BIT 5,(HL) |            |
| 111 | O   | 6F | LD L,A     | BIT 5,A    | RLD        |
| 112 | P   | 70 | LD (HL),B  | BIT 6,B    | IN F,(C)   |
| 113 | Q   | 71 | LD (HL),C  | BIT 6,C    |            |
| 114 | R   | 72 | LD (HL),D  | BIT 6,D    | SBC HL,SP  |
| 115 | S   | 73 | LD (HL),E  | BIT 6,E    | LD (NN),SP |
| 116 | T   | 74 | LD (HL),H  | BIT 6,H    |            |
| 117 | U   | 75 | LD (HL),L  | BIT 6,L    |            |
| 118 | V   | 76 | HALT       | BIT 6,(HL) |            |
| 119 | W   | 77 | LD (HL),A  | BIT 6,A    |            |
| 120 | X   | 78 | LD A,B     | BIT 7,B    | IN A,(C)   |
| 121 | Y   | 79 | LD A,C     | BIT 7,C    | OUT (C),A  |
| 122 | Z   | 7A | LD A,D     | BIT 7,D    | ADC HL,SP  |
| 123 | [   | 7B | LD A,E     | BIT 7,E    | LD SP,(NN) |
| 124 |     | 7C | LD A,H     | BIT 7,H    |            |
| 125 | ]   | 7D | LD A,L     | BIT 7,L    |            |
| 126 |     | 7E | LD A,(HL)  | BIT 7,(HL) |            |
| 127 |     | 7F | LD A,A     | BIT 7,A    |            |
| 128 |     | 80 | ADD A,B    | RES 0,B    |            |
| 129 |     | 81 | ADD A,C    | RES 0,C    |            |
| 130 |     | 82 | ADD A,D    | RES 0,D    |            |
| 131 |     | 83 | ADD A,E    | RES 0,E    |            |
| 132 |     | 84 | ADD A,H    | RES 0,H    |            |
| 133 |     | 85 | ADD A,L    | RES 0,L    |            |
| 134 |     | 86 | ADD A,(HL) | RES 0,(HL) |            |
| 135 |     | 87 | ADD A,A    | RES 0,A    |            |
| 136 |     | 88 | ADC A,B    | RES 1,B    |            |
| 137 |     | 89 | ADC A,C    | RES 1,C    |            |
| 138 |     | 8A | ADC A,D    | RES 1,D    |            |
| 139 |     | 8B | ADC A,E    | RES 1,E    |            |
| 140 |     | 8C | ADC A,H    | RES 1,H    |            |
| 141 |     | 8D | ADC A,L    | RES 1,L    |            |
| 142 |     | 8E | ADC A,(HL) | RES 1,(HL) |            |
| 143 |     | 8F | ADC A,A    | RES 1,A    |            |
| 144 | (A) | 90 | SUB B      | RES 2,B    |            |
| 145 | (B) | 91 | SUB C      | RES 2,C    |            |
| 146 | (C) | 92 | SUB D      | RES 2,D    |            |
| 147 | (D) | 93 | SUB E      | RES 2,E    |            |
| 148 | (E) | 94 | SUB H      | RES 2,H    |            |
| 149 | (F) | 95 | SUB L      | RES 2,L    |            |
| 150 | (G) | 96 | SUB (HL)   | RES 2,(HL) |            |
| 151 | (H) | 97 | SUB A      | RES 2,A    |            |
| 152 | (I) | 98 | SBC A,B    | RES 3,B    |            |
| 153 | (J) | 99 | SBC A,C    | RES 3,C    |            |
| 154 | (K) | 9A | SBC A,D    | RES 3,D    |            |
| 155 | (L) | 9B | SBC A,E    | RES 3,E    |            |
| 156 | (M) | 9C | SBC A,H    | RES 3,H    |            |
| 157 | (N) | 9D | SBC A,L    | RES 3,L    |            |
| 158 | (O) | 9E | SBC A,(HL) | RES 3,(HL) |            |
| 159 | (P) | 9F | SBC A,A    | RES 3,A    |            |
| 160 | (Q) | A0 | AND B      | RES 4,B    | LDI        |
| 161 | (R) | A1 | AND C      | RES 4,C    | CPI        |
| 162 | (S) | A2 | AND D      | RES 4,D    | INI        |
| 163 | (T) | A3 | AND E      | RES 4,E    | OUTI       |
| 164 | (U) | A4 | AND H      | RES 4,H    |            |
| 165 | RND | A5 | AND L      | RES 4,L    |            |

|     |         |    |            |            |      |
|-----|---------|----|------------|------------|------|
| 166 | INKEY*  | A6 | AND (HL)   | RES 4,(HL) |      |
| 167 | PI      | A7 | AND A      | RES 4,A    |      |
| 168 | FN      | A8 | XOR B      | RES 5,B    | LDD  |
| 169 | POINT   | A9 | XOR C      | RES 5,C    | CPD  |
| 170 | SCREEN* | AA | XOR D      | RES 5,D    | IND  |
| 171 | ATTR    | AB | XOR E      | RES 5,E    | OUTD |
| 172 | AT      | AC | XOR H      | RES 5,H    |      |
| 173 | TAB     | AD | XOR L      | RES 5,L    |      |
| 174 | VAL*    | AE | XOR (HL)   | RES 5,(HL) |      |
| 175 | CODE    | AF | XOR A      | RES 5,A    |      |
| 176 | VAL     | B0 | OR B       | RES 6,B    | LDIR |
| 177 | LEN     | B1 | OR C       | RES 6,C    | CDIR |
| 178 | SIN     | B2 | OR D       | RES 6,D    | INIR |
| 179 | COS     | B3 | OR E       | RES 6,E    | OTIR |
| 180 | TAN     | B4 | OR H       | RES 6,H    |      |
| 181 | ASN     | B5 | OR L       | RES 6,L    |      |
| 182 | ACS     | B6 | OR (HL)    | RES 6,(HL) |      |
| 183 | ATN     | B7 | OR A       | RES 6,A    |      |
| 184 | LN      | B8 | CP B       | RES 7,B    | LDDR |
| 185 | EXP     | B9 | CP C       | RES 7,C    | CPDR |
| 186 | INT     | BA | CP D       | RES 7,D    | INDR |
| 187 | SQR     | BB | CP E       | RES 7,E    | OTDR |
| 188 | SGN     | BC | CP H       | RES 7,H    |      |
| 189 | ABS     | BD | CP L       | RES 7,L    |      |
| 190 | PEEK    | BE | CP (HL)    | RES 7,(HL) |      |
| 191 | IN      | BF | CP A       | RES 7,A    |      |
| 192 | USR     | C0 | RET NZ     | SET 0,B    |      |
| 193 | STR*    | C1 | POP BC     | SET 0,C    |      |
| 194 | CHR*    | C2 | JP NZ,NN   | SET 0,D    |      |
| 195 | NOT     | C3 | JP NN      | SET 0,E    |      |
| 196 | BIN     | C4 | CALL NZ,NN | SET 0,H    |      |
| 197 | OR      | C5 | PUSH BC    | SET 0,L    |      |
| 198 | AND     | C6 | ADD A,N    | SET 0,(HL) |      |
| 199 | <=      | C7 | RST 0      | SET 0,A    |      |
| 200 | >=      | C8 | RET Z      | SET 1,B    |      |
| 201 | <>      | C9 | RET        | SET 1,C    |      |
| 202 | LINE    | CA | JP Z,NN    | SET 1,D    |      |
| 203 | THEN    | CB |            | SET 1,E    |      |
| 204 | TO      | CC | CALL Z,NN  | SET 1,H    |      |
| 205 | STEP    | CD | CALL NN    | SET 1,L    |      |
| 206 | DEF FN  | CE | ADC A,NN   | SET 1,(HL) |      |
| 207 | CAT     | CF | RST 8      | SET 1,A    |      |
| 208 | FORMAT  | D0 | RET NC     | SET 2,B    |      |
| 209 | MOVE    | D1 | POP DE     | SET 2,C    |      |
| 210 | ERASE   | D2 | JP NC,NN   | SET 2,D    |      |
| 211 | OPEN#   | D3 | OUT (N),A  | SET 2,E    |      |
| 212 | CLOSE#  | D4 | CALL NC,NN | SET 2,H    |      |
| 213 | MERGE   | D5 | PUSH DE    | SET 2,L    |      |
| 214 | VERIFY  | D6 | SUB N      | SET 2,(HL) |      |
| 215 | BEEP    | D7 | RST 16     | SET 2,A    |      |
| 216 | CIRCLE  | D8 | RET C      | SET 3,B    |      |
| 217 | INK     | D9 | EXX        | SET 3,C    |      |
| 218 | PAPER   | DA | JP C,NN    | SET 3,D    |      |
| 219 | FLASH   | DB | IN A,(NN)  | SET 3,E    |      |
| 220 | BRIGHT  | DC | CALL C,NN  | SET 3,H    |      |

|     |           |    |                                         |            |
|-----|-----------|----|-----------------------------------------|------------|
| 221 | INVERSE   | DD | POPRZEDA<br>INSTRUKCIEN<br>DOTYCZACE IX | SET 3,L    |
| 222 | OVEN      | DE | SBC A,N                                 | SET 3,(HL) |
| 223 | OUT       | DF | RST 24                                  | SET 3,A    |
| 224 | LPRINT    | E0 | RET PO                                  | SET 4,B    |
| 225 | LLIST     | E1 | POP HL                                  | SET 4,C    |
| 226 | STOP      | E2 | JP PO,NN                                | SET 4,D    |
| 227 | READ      | E3 | EX (SP),HL                              | SET 4,E    |
| 228 | DATA      | E4 | CALL PO,NN                              | SET 4,H    |
| 229 | RESTORE   | E5 | PUSH HL                                 | SET 4,L    |
| 230 | NEW       | E6 | AND N                                   | SET 4,(HL) |
| 231 | BORDER    | E7 | RST 32                                  | SET 4,A    |
| 232 | CONTINUE  | E8 | RET PE                                  | SET 5,B    |
| 233 | DIM       | E9 | JP (HL)                                 | SET 5,C    |
| 234 | REM       | EA | JP PE,NN                                | SET 5,D    |
| 235 | FOR       | EB | EX DE,HL                                | SET 5,E    |
| 236 | GO TO     | EC | CALL PE,NN                              | SET 5,H    |
| 237 | GO SUB    | ED |                                         | SET 5,L    |
| 238 | INPUT     | EE | XOR N                                   | SET 5,(HL) |
| 239 | LOAD      | EF | RST 40                                  | SET 5,A    |
| 240 | LIST      | F0 | RET P                                   | SET 6,B    |
| 241 | LET       | F1 | POP AF                                  | SET 6,C    |
| 242 | PAUSE     | F2 | JP P,NN                                 | SET 6,D    |
| 243 | NEXT      | F3 | DI                                      | SET 6,E    |
| 244 | POKE      | F4 | CALL P,NN                               | SET 6,H    |
| 245 | PRINT     | F5 | PUSH AF                                 | SET 6,L    |
| 246 | PLOT      | F6 | OR N                                    | SET 6,(HL) |
| 247 | RUN       | F7 | RST 48                                  | SET 6,A    |
| 248 | SAVE      | F8 | RET M                                   | SET 7,B    |
| 249 | RANDOMIZE | F9 | LD SP,HL                                | SET 7,C    |
| 250 | IF        | FA | JP M,NN                                 | SET 7,D    |
| 251 | CLS       | FB | EI                                      | SET 7,E    |
| 252 | DRAW      | FC | CALL M,NN                               | SET 7,H    |
| 253 | CLEAR     | FD | POPRZEDA<br>INSTRUKCJE<br>DOTYCZACE IY  | SET 7,L    |
| 254 | RETURN    | FE | CP N                                    | SET 7,(HL) |
| 255 | COPY      | FF | RST 56                                  | SET 7,A    |





НАБОР КОМАНД ПРОЦЕССОРА Z-80

| МНЕ-<br>МО-<br>КА | ОПИСАНИЕ |    | ФЛАГИ |   |     |     |     | АРГУМЕНТЫ           | Т  | L | REMARK                                     |
|-------------------|----------|----|-------|---|-----|-----|-----|---------------------|----|---|--------------------------------------------|
|                   |          |    | C     | Z | P/V | S   | N   |                     |    |   |                                            |
| RLCA              | CY       | A  | C     | . | .   | .   | 0 0 |                     | 4  | 1 |                                            |
| RLA               | CY       | A  | C     | . | .   | .   | 0 0 |                     | 4  | 1 |                                            |
| RRCA              | A        | CY | C     | . | .   | .   | 0 0 |                     | 4  | 1 |                                            |
| RRA               | A        | CY | C     | . | .   | .   | 0 0 |                     | 4  | 1 |                                            |
| RLC               | CY       | D  |       |   | P   | 0 0 |     | D=R                 | 8  | 2 | D=R, (HL),<br>(IX+D), (IY+D)               |
| RL D              | CY       | D  |       |   | P   | 0 0 |     | D=(HL)              | 15 | 2 |                                            |
| RRC D             | D        | CY |       |   | P   | 0 0 |     | D=(IX+D),<br>(IY+D) | 23 | 4 |                                            |
| RR D              | D        | CY |       |   | P   | 0 0 |     |                     |    |   |                                            |
| SLA D             | CY       | D  |       |   | P   | 0 0 |     |                     |    |   |                                            |
| SRA D             | D        | CY |       |   | P   | 0 0 |     |                     |    |   |                                            |
| SRL D             | D        | CY |       |   | P   | 0 0 |     |                     |    |   |                                            |
| RLD               |          |    | .     |   | P   | 0 0 |     |                     | 16 | 2 | СДВИГ ЦИФРЫ<br>В ЯЧЕЙКЕ ЗУ<br>ПО АДРЕСУ HL |
| RRD               |          |    | .     |   | P   | 0 0 |     |                     | 16 | 2 |                                            |

8-БИТОВЫЕ ПЕРЕСЫЛКИ

| МНЕМОНИКА | ОПИСАНИЕ | ФЛАГИ |   |     |       | T  | L | REMARK        |
|-----------|----------|-------|---|-----|-------|----|---|---------------|
|           |          | C     | Z | P/V | S N H |    |   |               |
| LD R, S   | R=S      |       |   |     |       | 4  | 1 | R=A, B, C, D, |
|           |          |       |   |     |       | 7  | 2 | E, H, L       |
| LD S, R   | S=R      |       |   |     |       | 19 | 3 | S=R, (HL),    |
|           |          |       |   |     |       |    |   | (IX+D),       |
|           |          |       |   |     |       |    |   | (IY+D)        |
| LD R, N   | R=N      |       |   |     |       | 7  | 2 | N= 0 - 255    |
| LD A, Y   | A=Y      |       |   |     |       | 7  | 1 | Y=(BC), (DE)  |
|           |          |       |   |     |       |    |   | (NN)          |
| LD Y, A   | Y=A      |       |   |     |       | 13 | 3 |               |
| LD A, P   | A=P      | I     | 0 | 0   |       | 9  | 2 | P=I, R        |
| LD P, A   | P=A      |       |   |     |       |    |   |               |

I - СОСТОЯНИЕ IFF КОПИРУЕТСЯ В P/V ДЛЯ LDA, I

16-БИТОВЫЕ ПЕРЕСЫЛКИ

| МНЕМОНИКА    | ОПИСАНИЕ            | ФЛАГИ |   |     |       | T  | L | REMARK                 |
|--------------|---------------------|-------|---|-----|-------|----|---|------------------------|
|              |                     | C     | Z | P/V | S N H |    |   |                        |
| LD RR', NN   | RR'=NN              |       |   |     |       | 10 | 3 | RR'=BC, DE, HL, SP     |
|              |                     |       |   |     |       | 14 | 4 | RR'=IX, IY             |
| LD RR', (NN) | RR'=(NN)            |       |   |     |       | 16 | 3 |                        |
|              |                     |       |   |     |       | 20 | 4 | RR'=BC, DE, SP, IX, IY |
| LD (NN), RR' | (NN)=RR'            |       |   |     |       |    |   |                        |
| PUSH SS'     | (SP)=S'<br>(SP-1)=S |       |   |     |       | 11 | 1 | SS'=BC, DE, HL, AF     |
|              |                     |       |   |     |       | 15 | 2 | SS'=IX, IY             |
| POP SS'      | S'=(SP)<br>S=(SP-1) |       |   |     |       | 10 | 1 | SS'=BC, DE, HL, AF     |
|              |                     |       |   |     |       | 14 | 2 | SS'=IX, IY             |
| LD SP, YY'   | SP=YY'              |       |   |     |       | 6  | 1 | YY'=HL, IX, IY         |
|              |                     |       |   |     |       | 10 | 2 | YY'=IX, IY             |

ОБМЕН РЕГИСТРОВ

| МНЕМОНИКА       | ОПИСАНИЕ                   | ФЛАГИ         | АРГУМЕН-                  | Т        | Л      | REMARK                                                   |
|-----------------|----------------------------|---------------|---------------------------|----------|--------|----------------------------------------------------------|
|                 |                            |               | ТЫ                        |          |        |                                                          |
|                 |                            | C Z P/V S N H |                           |          |        |                                                          |
| EX              | DE, HL                     | DE HL         |                           | 4        | 1      |                                                          |
| EX              | AF, AF'                    | AF AF'        |                           | 4        | 1      | ПЕРЕКЛЮЧЕНИЕ<br>НА АЛЬТЕРНАТИВ.<br>ПАРУ РЕГИСТРОВ<br>AF. |
| EXX             | BC BC'<br>DE DE'<br>HL HL' |               |                           | 4        | 1      | ПЕРЕКЛЮЧЕНИЕ НА<br>АЛЬТЕРНАТИВН.<br>BC', DE', HL'.       |
| EX (SP),<br>RR' | (SP) RR'                   |               | RR'=HL .<br>RR'=IX,<br>IY | 19<br>23 | 1<br>2 | RR'=HL, IX, IY                                           |

БЛОКОВЫЕ ПЕРЕСЫЛКИ И ОПЕРАЦИИ СРАВНЕНИЯ

| МНЕ-<br>МОНИ<br>КА | ОПИСАНИЕ                                                                    | ФЛАГИ         | УСЛОВИЯ                                   | T        | L | REMARK                                                     |
|--------------------|-----------------------------------------------------------------------------|---------------|-------------------------------------------|----------|---|------------------------------------------------------------|
|                    |                                                                             | C Z P/V S N H |                                           |          |   |                                                            |
| LDI                | (DE)=(HL)<br>DE=DE+1<br>HL=HL+1<br>BC=BC-1                                  | . . * . 0 0   |                                           | 16       | 2 | ПЕРЕСЫЛКА ИЗ<br>(HL) В (DE).                               |
| LDD                | (DE)=(HL)<br>DE=DE-1<br>HL=HL-1<br>BC=BC-1                                  | . . * . 0 0   |                                           | 16       | 2 | ПЕРЕСЫЛКА ИЗ<br>(HL) В (DE).                               |
| LDIR               | 1 (DE)=(HL)<br>2 DE=DE+1<br>3 HL=HL+1<br>4 BC=BC-1<br>IF BC<>0 GO<br>TO 1   | . . 0 . 0 0   | ЕСЛИ<br>BC=0<br>BC<>0                     | 16<br>21 | 2 | ПЕРЕСЫЛКА ИЗ<br>(HL) В (DE) ДО<br>ТЕХ ПОР ПОКА<br>BC <> 0. |
| LDDR               | 1 (DE)=(HL)<br>2 DE=DE-1<br>3 HL=HL-1<br>4 BC=BC-1<br>IF BC<>0 GO<br>TO 1   | . . 0 . 0 0   | ЕСЛИ<br>BC=0<br>BC<>0                     | 16<br>21 | 2 | ПЕРЕСЫЛКА ИЗ<br>(HL) В (DE) ДО<br>ТЕХ ПОР ПОКА<br>BC <> 0. |
| CPI                | A (HL)<br>BC=BC-1<br>HL=HL+1                                                | . 2 * . 1     |                                           | 16       | 2 |                                                            |
| CPD                | A (HL)<br>BC=BC-1<br>HL=HL-1                                                | . 2 * . 1     |                                           | 16       | 2 |                                                            |
| CPIR               | 1 A (HL)<br>2 BC=BC-1<br>3 HL=HL+1<br>IF (BC<>0)<br>AND(A<>(HL))<br>GO TO 1 | . 2 * . 1     | BC=0 ИЛИ<br>A=(HL).<br>BC<>0 И<br>A<>(HL) | 16<br>21 | 2 |                                                            |
| CPDR               | 1 A (HL)<br>2 BC=BC-1<br>3 HL=HL-1<br>IF (BC<>0)<br>AND(A<>(HL))<br>GO TO 1 | . 2 * . 1     | BC=0 ИЛИ<br>A=(HL).<br>BC<>0 И<br>A<>(HL) | 16<br>21 | 2 |                                                            |

\* ФЛАГ P/V=0, ЕСЛИ BC-1=0, ИНАЧЕ P/V=1  
2 ФЛАГ Z=1, ЕСЛИ A=(HL), ИНАЧЕ Z=0

8-БИТОВАЯ АРИФМЕТИКА И ЛОГИКА

| МНЕМО<br>НИКА | ОПИСАНИЕ   | ФЛАГИ         | УСЛОВИЯ | T                   | L                   | REMARK |                                                                                                                           |                            |
|---------------|------------|---------------|---------|---------------------|---------------------|--------|---------------------------------------------------------------------------------------------------------------------------|----------------------------|
|               |            | C Z P/V S N H |         |                     |                     |        |                                                                                                                           |                            |
| ADD S         | A=A+S      | V             | Ø       | S=R                 | 4                   | 1      | S=R,N,(HL),<br>(IX+D),(IY+D)<br>CY-СОСТОЯНИЕ<br>ФЛАГА C.<br>AND,OR,XOR -<br>ПРОИЗВОДИТСЯ<br>НАД КАЖДЫМ<br>БИТОМ АРГУМЕНТА |                            |
|               |            |               |         | S=N                 | 7                   | 2      |                                                                                                                           |                            |
| ADC S         | A=A+S+CY   | V             | Ø       | S=(HL)              | 7                   | 1      |                                                                                                                           |                            |
|               |            |               |         | S=(IX+D),<br>(IY+D) | 19                  | 3      |                                                                                                                           |                            |
| SUB S         | A=A-S      | V             | 1       |                     |                     |        |                                                                                                                           |                            |
| SBC S         | A=A-S-CY   | V             | 1       |                     |                     |        |                                                                                                                           |                            |
| AND S         | A=A AND S  | Ø             | P       | Ø                   |                     |        |                                                                                                                           |                            |
| OR S          | A=A OR S   | Ø             | P       | Ø                   |                     |        |                                                                                                                           |                            |
| XOR S         | A=A XOR S  | Ø             | P       | Ø                   |                     |        |                                                                                                                           |                            |
| CP S          | A S        | V             | 1       |                     |                     |        |                                                                                                                           |                            |
| INC D         | D=D+1      | .             | V       | Ø                   | D=R                 | 4      | 1                                                                                                                         | D=R,(HL),<br>(IX+D),(IY+D) |
|               |            |               |         |                     | S=(HL)              | 11     | 1                                                                                                                         |                            |
| DEC D         | D=D-1      | .             | V       | 1                   | D=(IX+D),<br>(IY+D) | 23     | 3                                                                                                                         |                            |
| DAA           | (A)2-(A)10 | P             | .       |                     |                     | 4      | 1                                                                                                                         | ДЕСЯТИЧ, КОРРЕК            |
| CPL           | A=NOT(A)   | .             | .       | .                   | .                   | 1      | 1                                                                                                                         | ИНВЕРСИЯ РЕГ. A            |
| NEG           | A=-A       | V             | 1       |                     |                     | 8      | 2                                                                                                                         |                            |

СПЕЦИАЛЬНЫЕ И УПРАВЛЯЮЩИЕ КОМАНДЫ

| МНЕ<br>МОНИ<br>КА | ОПИСАНИЕ                  | ФЛАГИ |   |     |   |   |   | УСЛОВИЯ | Т | L | REMARK                                                                                                       |
|-------------------|---------------------------|-------|---|-----|---|---|---|---------|---|---|--------------------------------------------------------------------------------------------------------------|
|                   |                           | C     | Z | P/V | S | N | H |         |   |   |                                                                                                              |
| NOP               |                           | .     | . | .   | . | . | . |         | 4 | 1 |                                                                                                              |
| HALT              | ОСТАНОВ CPU               | .     | . | .   | . | . | . |         | 4 | 1 |                                                                                                              |
| EI                | IFF=0                     | .     | . | .   | . | . | . |         | 4 | 1 | РАЗРЕШЕНИЕ ПРЕР                                                                                              |
| DI                | IFF=1                     | .     | . | .   | . | . | . |         | 4 | 1 | БЛОКИР. ПРЕРЫВ.                                                                                              |
| IM 0              | 0-ОИ РЕЖ.<br>ПРИЕМА ПРЕР. | .     | . | .   | . | . | . |         | 8 | 2 | ОБСЛ ПРЕРЫВАНИИ<br>ЧЕРЕЗ RST-ПОДПР                                                                           |
| IM 1              | 1-ОИ РЕЖ.<br>ПРИЕМА ПРЕР. | .     | . | .   | . | . | . |         | 8 | 2 | ОБСЛ ПРЕРЫВАНИИ<br>ЧЕРЕЗ RST 56                                                                              |
| IM 2              | 2-ОИ РЕЖ.<br>ПРИЕМА ПРЕР. | .     | . | .   | . | . | . |         | 8 | 2 | ОБСЛ ПРЕРЫВАНИИ<br>ЧЕРЕЗ ПОДПРОГР.<br>РАСПОЛОЖЕННЫЕ В<br>ПАМЯТИ ПО АДР.<br>УКАЗ-МУ В РЕГ-Е<br>I(СТАРШ. БАЙТ) |

16-БИТОВАЯ АРИФМЕТИКА

| МНЕМОНИКА  | ОПИСАНИЕ     | ФЛАГИ |   |     |   |   |   | УСЛОВИЯ | Т | L | REMARK          |
|------------|--------------|-------|---|-----|---|---|---|---------|---|---|-----------------|
|            |              | C     | Z | P/V | S | N | H |         |   |   |                 |
| ADD HL,SS' | HL=HL+SS'    | .     | . | .   | . | 0 | X | 11      | 1 | 1 | SS'=BC,DE,HL,SP |
| ADC HL,SS' | HL=HL+SS'+CY |       |   | V   |   | 0 | X | 15      | 2 |   |                 |
| SBC HL,SS' | HL=HL-SS'-CY |       |   | V   |   | 1 | X | 15      | 2 |   |                 |
| ADD IX,PP' | IX=IX+PP'    | .     | . | .   | . | 0 | X | 15      | 2 | 2 | PP'=DC,DE,IX,SP |
| ADD IY,RR' | IX=IX+RR'    | .     | . | .   | . | 0 | X | 15      | 2 | 2 | RR'=DC,DE,IX,SP |
| INC SS'    | SS'=SS'+1    | .     | . | .   | . | . | . | 6       | 1 |   |                 |
| INC IX     | IX=IX+1      | .     | . | .   | . | . | . | 10      | 2 |   |                 |
| INC IY     | IY=IY+1      | .     | . | .   | . | . | . | 10      | 2 |   |                 |
| DEC SS'    | SS'=SS'-1    | .     | . | .   | . | . | . | 6       | 1 |   |                 |
| DEC IX     | IX=IX-1      | .     | . | .   | . | . | . | 10      | 2 |   |                 |
| DEC IY     | IY=IY-1      | .     | . | .   | . | . | . | 10      | 2 |   |                 |



ОПЕРАЦИИ НАД ОТДЕЛЬНЫМИ БИТАМИ

| МНЕМОНИ<br>КА | ОПИСАНИЕ   | ФЛАГИ         | УСЛОВИЯ                              | T L                 | REMARK                                                   |
|---------------|------------|---------------|--------------------------------------|---------------------|----------------------------------------------------------|
|               |            | C Z P/V S N H |                                      |                     |                                                          |
| BIT B,S       | Z=SB       | . X X 0 1     | S=R<br>S=(HL)<br>S=(IX+D),<br>(IY+D) | 8 2<br>12 2<br>20 4 | РЕГИСТР Z ПРИН.<br>ЗНАЧЕНИЕ БИТА<br>НОМЕР B ИЗ ЯЧ.<br>S. |
| SET B,S       | SB=1       | . . . . .     | S=R<br>S=(HL)                        | 8 2<br>15 2         |                                                          |
| RES B,S       | SB=0       | . . . . .     | S=(IX+D),<br>(IY+D)                  | 23 4                |                                                          |
| CCF           | CY=NOT(CY) | . . . 0 X     |                                      | 4 1                 | ИНВЕРСИЯ CY                                              |
| SCF           | CY=1       | 1 . . . 0 0   |                                      | 4 1                 |                                                          |

КОМАНДЫ ПЕРЕХОДА

| МНЕМО<br>НИКА | ОПИСАНИЕ                    | ФЛАГИ         | УСЛОВИЯ                                      | T L        | REMARK                     |
|---------------|-----------------------------|---------------|----------------------------------------------|------------|----------------------------|
|               |                             | C Z P/V S N H |                                              |            |                            |
| JP NN         | GO TO NN                    | . . . . .     |                                              | 10 3       |                            |
| JP CC,<br>NN  | IF CC GOTO NN               | . . . . .     |                                              | 10 3       | CC=NZ,Z,NC,C,<br>PO,PE,P,M |
| JR E          | GOTO E:<br>REM PC=PC+E      | . . . . .     |                                              | 12 2       | E=-128 - +128              |
| JR W,E        | IF W GOTO E:<br>REM PC=PC+E | . . . . .     | УСЛОВИЕ W<br>ВЫПОЛН.<br>УСЛОВИЕ W<br>НЕ ВЫП. | 12 2<br>7  | W=C,NC,Z,NZ                |
| JR E          | GOTO (RR)                   | . . . . .     | RR=HL<br>RR=IX,IY                            | 4 1<br>8 2 | RR=HL,IX,IY                |
| DJNZ E        | B=B-1<br>IF B<>0 GO<br>TO E | . . . . .     | ЕСЛИ B<>0<br>ЕСЛИ B=0                        | 13 2<br>8  |                            |

-164-  
КОМАНДЫ ВВОДА-ВЫВОДА

| МНЕМОНИ  | ОПИСАНИЕ                                             | ФЛАГИ         | УСЛОВИЯ                | T        | L | REMARK                          |
|----------|------------------------------------------------------|---------------|------------------------|----------|---|---------------------------------|
|          |                                                      | C Z P/V S N H |                        |          |   |                                 |
| IN A,(N) | A=IN (N)                                             | . . . . .     |                        | 11       | 2 | A0 - A7 = N<br>A8 - A15= AKUM.  |
| OUT(N),A | OUT N,A                                              | . . . . .     |                        | 12       | 2 | A0 - A7 = N<br>A8 - A15= AKUM.  |
| IN R,(C) | R= IN(C)                                             | . P 0         |                        | 12       | 2 | A0 - A7 = C<br>A8 - A15= B      |
| OUT(C),N | OUT C,N                                              | . . . . .     |                        | 12       | 2 | A0 - A7 = C<br>A8 - A15= B      |
| INI      | (HL)=IN(C)<br>B=B-1<br>HL=HL+1                       | X * X X 1 X   |                        | 16       | 2 | A0 - A7 = C<br>A8 - A15= B      |
| IND      | (HL)=IN(C)<br>B=B-1<br>HL=HL-1                       | X * X X 1 X   |                        | 16       | 2 | A0 - A7 = C<br>A8 - A15= B      |
| OUTI     | OUT C,(HL)<br>B=B-1<br>HL=HL+1                       | X * X X 1 X   |                        | 16       | 2 | A0 - A7 = C<br>A8 - A15= B      |
| OUTD     | OUT C,(HL)<br>B=B-1<br>HL=HL-1                       | X * X X 1 X   |                        | 16       | 2 | A0 - A7 = C<br>A8 - A15= B      |
| INIR     | (HL)=IN(C)<br>B=B-1<br>HL=HL+1<br>IF B<>0 GO<br>TO 1 | X 1 X X 1 X   | ЕСЛИ В =0<br>ЕСЛИ В<>0 | 21<br>16 | 2 | ПОВТОРЕНИЕ INI<br>ПОКА В <> 0.  |
| INDR     | (HL)=IN(C)<br>B=B-1<br>HL=HL-1<br>IF B<>0 GO<br>TO 1 | X 1 X X 1 X   | ЕСЛИ В =0<br>ЕСЛИ В<>0 | 21<br>16 | 2 | ПОВТОРЕНИЕ IND<br>ПОКА В <> 0.  |
| OTIR     | OUT C,(HL)<br>B=B-1<br>HL=HL+1<br>IF B<>0 GO<br>TO 1 | X 1 X X 1 X   | ЕСЛИ В =0<br>ЕСЛИ В<>0 | 21<br>15 | 2 | ПОВТОРЕНИЕ OUTI<br>ПОКА В <> 0. |
| OTDR     | OUT C,(HL)<br>B=B-1<br>HL=HL-1<br>IF B<>0 GO<br>TO 1 | X 1 X X 1 X   | ЕСЛИ В =0<br>ЕСЛИ В<>0 | 21<br>15 | 2 | ПОВТОРЕНИЕ OUTD<br>ПОКА В <> 0. |

Z=1 КОГДА B-1=0, Z=0 ЕСЛИ B-1<>0

ПОДПРОГРАММЫ

| МНЕМО<br>НИКА  | ОПИСАНИЕ                                   | ФЛАГИ         | УСЛОВИЯ                                            | Т        | Л | REMARK                                                               |
|----------------|--------------------------------------------|---------------|----------------------------------------------------|----------|---|----------------------------------------------------------------------|
|                |                                            | C Z P/V S N H |                                                    |          |   |                                                                      |
| CALL<br>NN     | GOSUB NN                                   | . . . . .     |                                                    | 17       | 3 |                                                                      |
| CALL<br>CC, NN | IF CC GOSUNN                               | . . . . .     | УСЛОВИЕ ВЫП<br>УСЛ, НЕ ВЫП                         | 17<br>10 | 3 | CC=NZ, Z, NC, C,<br>PO, PE, P, M                                     |
| RET            | RETURN                                     | . . . . .     |                                                    | 10       | 1 |                                                                      |
| RET<br>CC      | IF CC THEN<br>RETURN                       | . . . . .     | УСЛОВИЕ CC<br>ВЫПОЛНИЛОСЬ<br>УСЛОВИЕ W<br>НЕ ВЫП.. | 11<br>5  | 1 |                                                                      |
| RETI           | RET ИЗ ОБСЛ.<br>ПРЕРЫВАНИЯ                 | . . . . .     |                                                    | 14       | 2 |                                                                      |
| RETN           | RET ИЗ ОБСЛ.<br>НЕМАСКИРУЕМ.<br>ПРЕРЫВАНИЯ | . . . . .     |                                                    | 14       | 2 |                                                                      |
| RST P          | GOSUB P<br>НЕМАСКИРУЕМ.<br>ПРЕРЫВАНИЯ      | . . . . .     |                                                    | 11       | 1 | P=0, 8, 16, 24, .<br>32, 40, 48, 56<br>ВЫЗОВ РЕСТАР-<br>ТА ПО АДР. P |



СПИСОК КОМПЛЕКТА SPECTRUM ZX

| * [ НАИМЕНОВАНИЕ ] [ ДОП. ЗАМЕНА ] [ КОЛ. ШТ. ] |                        |                |   |
|-------------------------------------------------|------------------------|----------------|---|
| 1                                               | Z'80A                  | UB880 D        | 1 |
| 2                                               | 2764                   | K573P44        | 2 |
|                                                 |                        | K573P46        | 2 |
|                                                 |                        | 27128          | 1 |
|                                                 |                        | 27256          | 1 |
| 3                                               | K565PY5Г               | K565PY5B, Д    | 8 |
| 4                                               | K555KP23               | K555KP27       | 4 |
| 5                                               | K555KP12               | K555KP2        | 6 |
| 6                                               | K555KP15               | K555KP7        | 1 |
| 7                                               | K555HE10               |                | 3 |
| 8                                               | K555LP5                |                | 2 |
| 9                                               | K555LP8                |                | 1 |
| 10                                              | K555LN1                |                | 2 |
| 11                                              | K555LA3                |                | 2 |
| 12                                              | K555TM8                |                | 1 |
| 13                                              | K555TM2                |                | 1 |
| 14                                              | K555KD4                |                | 1 |
| 15                                              | K555LL1                |                | 1 |
| 16                                              | K555LE4                |                | 1 |
| 17                                              | K561HE10               |                | 2 |
| 18                                              | K561LN1                |                | 1 |
| 19                                              | K531LN1                |                | 1 |
| 20                                              | K155PE3                |                | 2 |
| 21                                              | K556PT4                |                | 2 |
| 22                                              | KP580BB55              | 8255           | 1 |
| 23                                              | K554CA3                |                | 1 |
| 24                                              | КВАРЦЕВЫЙ<br>РЕЗОНАТОР | 10 МГц +/- 0,2 | 1 |

