



ZX SPECTRUM

ГЛАВА 1.	ВВЕДЕНИЕ.....	1
ГЛАВА 2.	ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ БЕЙСИК.....	3
ГЛАВА 3.	УСЛОВИЯ.....	4
ГЛАВА 4.	ЦИКЛЫ.....	5
ГЛАВА 5.	ПОДПРОГРАММЫ.....	5
ГЛАВА 6.	ОПЕРАТОРЫ READ, DATA И RESTORE.....	6
ГЛАВА 7.	АРИФМЕТИЧЕСКИЕ ОПЕРАЦИИ.....	6
ГЛАВА 8.	СТРОКИ СИМВОЛОВ.....	7
ГЛАВА 9.	ФУНКЦИИ.....	8
ГЛАВА 10.	МАТЕМАТИЧЕСКИЕ ФУНКЦИИ.....	9
ГЛАВА 11.	СЛУЧАЙНЫЕ ЧИСЛА.....	10
ГЛАВА 12.	МАССИВЫ.....	10
ГЛАВА 13.	ЛОГИЧЕСКИЕ ОПЕРАЦИИ.....	11
ГЛАВА 14.	НАБОР СИМВОЛОВ.....	11
ГЛАВА 15.	ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ ОБ ОПЕРАТОРАХ "PRINT" И "INPUT".....	13
ГЛАВА 16.	ЦВЕТА.....	14
ГЛАВА 17.	ГРАФИКА.....	17
ГЛАВА 18.	УКАЗАНИЯ.....	18
ГЛАВА 19.	ПРОГРАММИРОВАНИЕ ЗВУКОВ.....	19
ГЛАВА 20.	ВНЕШНЯЯ ПАМЯТЬ НА МАГНИТНОЙ ЛЕНТЕ.....	19
ГЛАВА 21.	УСТРОЙСТВО ПЕЧАТИ.....	21
ГЛАВА 22.	ДРУГОЕ ПЕРИФЕРИЙНОЕ ОБОРУДОВАНИЕ.....	21
ГЛАВА 23.	ВВОД И ВЫВОД.....	22
ГЛАВА 24.	ПАМЯТЬ.....	22
ГЛАВА 25.	СИСТЕМНЫЕ ПЕРЕМЕННЫЕ.....	24
ГЛАВА 26.	ИСПОЛЬЗОВАНИЕ МАШИННЫХ КОДОВ.....	26
ПРИЛОЖЕНИЕ А.	ПОЛНЫЙ НАБОР СИМВОЛОВ.....	28
ПРИЛОЖЕНИЕ В.	СООБЩЕНИЯ.....	30
ПРИЛОЖЕНИЕ С.	ФУНКЦИИ И ОПЕРАТОРЫ.....	32
ПРИЛОЖЕНИЕ Д.	ПРИМЕРЫ ПРОГРАММ.....	37
ПРИЛОЖЕНИЕ Е.	ШЕСТНАДЦАТИРИЧНАЯ И ДВОИЧНАЯ СИСТЕМЫ СЧИСЛЕНИЯ.....	40
УКАЗАТЕЛЬ КОМАНД БЕЙСИКА.....		40



ГЛАВА 1

ВВЕДЕНИЕ

ZX SPECTRUM-СОВМЕСТИМЫЕ КОМПЬЮТЕРЫ ИМЕЮТ ВСТРОЕННЫЙ ЯЗЫК ПРОГРАММИРОВАНИЯ БЕЙСИК (BASIC - BEGINNER ALL-PURPOSE SYMBOLIC INSTRUCTION CODE - МНОГОЦЕЛЕВОЙ ЯЗЫК СИМВОЛИЧЕСКИХ КОМАНД ДЛЯ НАЧИНАЮЩИХ). ЭТА ВЕРСИЯ ИМЕЕТ НЕКОТОРЫЕ ОСОБЕННОСТИ. НАПРИМЕР, ТАКИЕ КОМАНДЫ КАК PRINT, LET И INPUT ИСПОЛЮЮТСЯ ВО ВСЕХ КОМПЬЮТЕРАХ, ИМЕЮЩИХ БЕЙСИК, А ТАКИЕ КОМАНДЫ КАК BORDER, PAPER И BEER ИСПОЛЮЮТСЯ В ZX SPECTRUM.

ИЗУЧАЯ ЭТО РУКОВОДСТВО, 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 ВСТРЕТИТЕ ПРЕДЛОЖЕНИЕ ЧТО-НИБУДЬ ВВЕСТИ В КОМПЬЮТЕР И ВЫПОЛНИТЬ НА НЕМ, СПРАШИВАЙТЕ СЕБЯ: "ЧТО Я МОГУ СДЕЛАТЬ ВМЕСТО ЭТОГО?" И ПОПРОБУЙТЕ ЭТО ПРОДЕЛАТЬ. ЧЕМ БОЛЬШЕ СОБСТВЕННЫХ ПРОГРАММ ВЫ НАПИШИТЕ, ТЕМ ЛУЧШЕ ВЫ БУДЕТЕ ПОНИМАТЬ, КАК РАБОТАЕТ КОМПЬЮТЕР.

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

КЛАВИАТУРА

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

ДЛЯ ТОГО, ЧТОБЫ РЕАЛИЗОВАТЬ ВСЕ ЭТИ ФУНКЦИИ И КОМАНДЫ, НЕКОТОРЫЕ КЛАВШИ 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.

СОСТОЯНИЕ ИНДИЦИРУЕТСЯ КУРСОРОМ - ПЕРЧАТКОЙ БУКВЫ, КОТОРАЯ ПОКАЗЫВАЕТ, ГДЕ БУДЕТ ПОЯВЛЯТЬСЯ НА ЭКРАНЕ СЛЕДУЮЩИЙ НАБИРАЕМЫЙ СИМВОЛ.

РЕЖИМ [C] АВТОМАТИЧЕСКИ ЗАМЕНЯЕТ РЕЖИМ [L], КОГДА КОМПЬЮТЕР ОЖИДАЕТ КОМАНДУ ИЛИ ПРОГРАММНУЮ СТРОКУ (ОТЛИЧАЮЩУСЯ ОТ ВВОДИМЫХ 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 ЖЕ К ЗНАКОМЕСТУ СРАЗУ ЖЕ ПОСЛЕ ":" (ЗА ИСКЛЮЧЕНИЕМ АВТОСТОПОВ В СТРОКЕ). ЕСЛИ НЕ ИЗМЕНЕН РЕЖИМ, ТО НАЖАТИЕ СЛЕДУЮЩЕЙ КЛАВИШИ БУДЕТ ИНТЕРПРЕТИРОВАТЬСЯ КАК КЛЮЧЕВОЕ СЛОВО, НАПИСАННОЕ НА КЛАВИШЕ, 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 [C] И В [L] РЕЖИМАХ ОДНОВРЕМЕННОЕ НАЖАТИЕ КЛАВИШ SYMBOL SHIFT И КАКОЙ-ЛИБО КЛАВИШИ ВОСПРИНИМАЕТСЯ КАК ВСПОМОГАТЕЛЬНЫЙ СИМВОЛ, ИЗБРАЖЕННЫЙ НА КЛАВИШЕ, А В СЛУЧАЕ CAPS SHIFT С ЦИФРОВОЙ КЛАВИШЕЙ - КАК УПРАВЛЯЮЩАЯ ФУНКЦИЯ, НАПИСАННАЯ НА ЦИФРОВОЙ КЛАВИШЕ.

НАЖАТИЕ КЛАВИШИ CAPS SHIFT С ДРУГИМИ КЛАВИШАМИ В РЕЖИМЕ КУРСОРА [C] НЕ ВЛИЯЕТ НА КЛЮЧЕВЫЕ СЛОВА, А В РЕЖИМЕ КУРСОРА [L] ВЫЗЫВАЕТ ПОЯВЛЕНИЕ ЗАГЛАВНЫХ БУКВ.

РЕЖИМ КУРСОРА [S] (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 [S] И НАОБОРОТ.

РЕЖИМ КУРСОРА [S] (РАСШИРЕНИЕ) ИСПОЛЬЗУЕТСЯ ДЛЯ ПОЛУЧЕНИЯ ДОПОЛНИТЕЛЬНЫХ СИМВОЛОВ (ОБЫЧНО ЗНАКОВ). КУРСОР [S] ПОЯВЛЯЕТСЯ ПОСЛЕ ОДНОВРЕМЕННОГО НАЖАТИЯ ОБЕИХ КЛАВИШ СМЕНИ РЕЖИМА И СОХРАНЯЕТСЯ ДО НАЖАТИЯ КАКОЙ-ЛИБО ОДНОЙ ИЗ НИХ. В ЭТОМ РЕЖИМЕ НАЖАТИЕ ДАЕТ ОДИН СИМВОЛ ИЛИ ЗНАК, ЕСЛИ РЕЖИМ СОХРАНЯЕТСЯ, И ДРУГОЙ, ЕСЛИ ОДНОВРЕМЕННО НАЖАТА ОДНА ИЗ КЛАВИШ СМЕНИ РЕЖИМА.

ОДНОВРЕМЕННОЕ НАЖАТИЕ ЦИФРОВЫХ КЛАВИШ С КЛАВИШЕЙ СМЕНИ РЕЖИМА SYMBOL SHIFT ВЫЗЫВАЕТ ПОЯВЛЕНИЕ ЗНАКА, В ПРОТИВНОМ СЛУЧАЕ ОНИ УПРАВЛЯЮТ ЦВЕТОМ.

РЕЖИМ КУРСОРА [G] ВОЗНИКАЕТ ПОСЛЕ НАЖАТИЯ КЛАВИШИ GRAPHICS (CAPS SHIFT И 9) И СОХРАНЯЕТСЯ ДО ТЕХ ПОР, ПОКА НЕ БУДЕТ НАЖАТА КЛАВИША 9 ОДНА ИЛИ СОВМЕСТНО С CAPS SHIFT.

ЦИФРОВЫЕ КЛАВШИ ДАЮТ ТАКЖЕ ГРАФИЧЕСКИЕ СИМВОЛЫ, ЗА ИСКЛЮЧЕНИЕМ GRAPHICS ИI DELFTE; КАЖДАЯ ИЗ БУКВЕННЫХ КЛАВИШ, КРОМЕ V, W, X, Y И Z МОГУТ ВЫЗЫВАТЬ ПОЯВЛЕНИЕ ОПРЕДЕЛЕННЫХ ПОЛЬЗОВАТЕЛЕМ ГРАФИЧЕСКИХ СИМВОЛОВ.

ЕСЛИ НЕКОТОРАЯ КЛАВИША УДЕРЖИВАЕТСЯ В НАЖАТОМ СОСТОЯНИИ БОЛЕЕ ЧЕМ 2 ИЛИ 3 СЕКУНДЫ, ЭТО ВЫЗОВЕТ 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 ПО ЭКРАНУ КЛАВШАМИ

ВЛЕВО - CAPS SHIFT И 5;

ВПРАВО - CAPS SHIFT И 8 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 ВВОДИТСЯ КАК ОЧЕРЕДНАЯ СТРОКА В ПРОГРАММУ, ЛИБО ИСПОЛЬЗУЕТСЯ КАК СПИСОК ДАННЫХ ДЛЯ INPUT-ВВОДА. ЕСЛИ ЖЕ ОНА СОДЕРЖИТ СИНТАКСИЧЕСКИЕ ОШИБКИ, ТО ОШИБОЧНОЕ МЕСТО УКАЗЫВАЕТСЯ ПЕРЧАТКОЙ ЗНАКОМЕСТА ВОПРОСА [?].

КОГДА ВВОДЯТСЯ СТРОКИ ПРОГРАММЫ, ТО ЛИСТИНГ ОТОБРАЖАЕТСЯ В ВЕРХНЕЙ ПОЛОВИНЕ ЭКРАНА. ПОСЛЕДНЯЯ ВВЕДЕННАЯ СТРОКА НАЗЫВАЕТСЯ ТЕКУЩЕЙ И УКАЗЫВАЕТСЯ СИМВОЛОМ [I], ЕГО МОЖНО ПЕРЕМЕЩАТЬ ВШЕ ИЛИ НИЖЕ, ИСПОЛЬЗУЯ КЛАВШИ 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 КОМАНД И ПРОГРАММ, ВВОД ОСУЩЕСТВЛЯЕТСЯ В ВЕРХНЮЮ ЧАСТЬ ЭКРАНА И СОХРАНЯЕТСЯ ДО ВВОДА СТРОКИ ПРОГРАММЫ, ЛИБО НАЖАТИЯ КЛАВИШИ ENTER ПРИ НАЛИЧИИ ПУСТОЙ СТРОКИ, ЛИБО НАЖАТИЕ КЛАВИШ ПЕРЕМЕЩЕНИЯ КУРСОРА ВВЕРХ, ВНИЗ.

В НИЖНЮЮ ЧАСТЬ ЭКРАНА ВЫВОДЯТСЯ ТАКЖЕ СООБЩЕНИЯ ОБ ОШИБКАХ, КОТОРЫЕ СОХРАНЯЮТСЯ ТАМ ДО НАЖАТИЯ ЛЮБОЙ ИЗ КЛАВИШ (ЭТО ИНДИЦИРУЕТСЯ ПЕРЕХОДОМ В РЕЖИМ [C]).

В ОПРЕДЕЛЕННЫХ СОСТОЯНИЯХ КЛАВШИ 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 МАГНИСКОМ.

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 ЗАПОЛНИТСЯ 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 REPEATS". НАЖАТИЕ ЛЮБОЙ ДРУГОЙ ИЗ КЛАВИШ РАЗРЕШАЕТ СВЕРТКУ.

НИЖНЯЯ ЧАСТЬ ЭКРАНА ИСПОЛЬЗУЕТСЯ ДЛЯ ВВОДА КОМАНД, СТРОК ПРОГРАММЫ И ВВОДИМЫХ ДАННЫХ, А ТАКЖЕ ДЛЯ ВЫВОДА СООБЩЕНИЙ СИСТЕМЫ.

РЕЖИМ КАЛЬКУЛЯТОРА

=====

ЕСЛИ ПОСЛЕ ВВОДА СЛОВА PRINT МЫ НАЖИМ НА ЦИФРУ 2, НА ЭКРАНЕ ПОЯВИТСЯ:

PRINT 2

ТЕПЕРЬ ДЛЯ ВЫПОЛНЕНИЯ КОМАНДЫ НАЖИМИТЕ ENTER. В ВЕРХНЕЙ ЧАСТИ ЭКРАНА ПОЯВИТСЯ ПРОСТО ЦИФРА 2. МОЖНО ТАКЖЕ СОЧЕТАТЬ БУКВУ И ЦИФРУ:

PRINT 2;"ABC"

ИЛИ

PRINT 2;"ABC"

И ПОТОМ

PRINT 2;"ABC"

НАПОМИНИ, ЧТО ДЛЯ ВЫПОЛНЕНИЯ ЛЮБОЙ КОМАНДЫ НУЖНО НАЖАТЬ ENTER. ЗАПЯТАЯ, ПОСТАВЛЕННАЯ МЕЖДУ ЭЛЕМЕНТАМИ ПОСЛЕ СЛОВА PRINT, РАЗДЕЛЯЕТ ИХ НА 16 КОЛОНОК; ИСПОЛЬЗОВАНИЕ ТОЧКИ С ЗАПЯТОЙ НЕ ОСТАВЛЯЕТ ПРОСТРАНСТВА. КОГДА НЕ ИСПОЛЬЗУЕТСЯ НИ ТОТ, НИ ДРУГОЙ ЗНАК, ПОЯВЛЯЕТСЯ ОШИБКА.

ИНСТРУКЦИЯ PRINT МОЖЕТ ИСПОЛЬЗОВАТЬСЯ С МАТЕМАТИЧЕСКИМИ ФУНКЦИЯМИ, ОБОЗНАЧЕННЫМИ НА КЛАВИАТУРЕ, В РЕЗУЛЬТАТЕ ЧЕГО СПЕКТРУМ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ И КАК ЭЛЕКТРОННЫЙ КАЛЬКУЛЯТОР. НАПРИМЕР:

PRINT 2+2

ПОЯВЛЯЕТСЯ В ВЕРХНЕЙ ЧАСТИ ЭКРАНА. СРАВНИТЕ ЭТО С PRINT "2+2" ЭТИ ДВЕ ЗАДАЧИ МОЖНО ОБЪЕДИНИТЬ В ОДНУ ДЛЯ ТОГО, ЧТОБЫ ПОЛУЧИТЬ ЧТО-ЛИБО БОЛЕЕ ПОЛЕЗНОЕ. ПОПРОБУЙТЕ PRINT "2+2"="12+2" ПОПРОБУЙТЕ ДРУГИЕ АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ:

PRINT 3-2

PRINT 4/5

PRINT 12*2

ЗНАК * ИСПОЛЬЗУЕТСЯ КАК ЗНАК УМНОЖЕНИЯ ВМЕСТО X ДЛЯ ТОГО, ЧТОБЫ ИЗБЕЖАТЬ ВОЗМОЖНОЙ ПУТАНИЦЫ С ЛАТИНСКОЙ БУКВОЙ "X"; ЗНАК / ИСПОЛЬЗУЕТСЯ ДЛЯ ОБОЗНАЧЕНИЯ ДЕЛЕНИЯ.

ПОЭКСПЕРИМЕНТИРУЙТЕ С РАЗЛИЧНЫМИ МАТЕМАТИЧЕСКИМИ ПОДСЧЕТАМИ. ЕСЛИ ХОТИТЕ, МОЖЕТЕ ИСПОЛЬЗОВАТЬ ОТРИЦАТЕЛЬНЫЕ ЧИСЛА, А ТАКЖЕ ДЕСЯТИЧНЫЕ ДРОБИ.

ЕСЛИ ВЫ ИСПОЛЬЗОВАЛИ БОЛЬШЕ, ЧЕМ 22 СТРОКИ В ВЕРХНЕЙ ЧАСТИ ЭКРАНА, ТО ЗАМЕТИТЕ, ЧТО ПРОИСХОДИТ НЕЧТО ИНТЕРЕСНОЕ: ВСЕ СМЕЩАЕТСЯ ВВЕРХ НА ОДНУ СТРОКУ И ВЕРХНЯЯ СТРОКА ПРОПАДАЕТ. ЭТО НАЗЫВАЕТСЯ "SCROLL" (ЗАВОРАЧИВАТЬ).

АРИФМЕТИЧЕСКИЕ ДЕЙСТВИЯ НЕ ВСЕГДА ВЫПОЛНЯЮТСЯ В ТОЙ ОЧЕРЕДНОСТИ, КОТОРУЮ ВЫ ОЖИДАЕТЕ. НАПРИМЕР, ПОПРОБУЙТЕ НАПИСАТЬ:

PRINT 2+3*5

ОЧЕВИДНО, ВЫ ПРЕДПОЛАГАЕТЕ, ЧТО КОМПЬЮТЕР СНАЧАЛА СЛОЖИТ 2 И 3, А ЗАТЕМ РЕЗУЛЬТАТ УМНОЖИТ НА 5, ЧТО ДАСТ 25. ОДНАКО, ЭТО НЕ ТАК. ОПЕРАЦИИ УМНОЖЕНИЯ И ДЕЛЕНИЯ ВСЕГДА ВЫПОЛНЯЮТСЯ РАНЬШЕ, ЧЕМ СЛОЖЕНИЕ И ВЫЧИТАНИЕ. ПОЭТОМУ ВЫРАЖЕНИЕ 2+3*5 ОЗНАЧАЕТ, ЧТО СНАЧАЛА БУДЕТ ВЫПОЛНЕНО УМНОЖЕНИЕ 3*5=15, И ТОЛЬКО ЗАТЕМ СЛОЖЕНИЕ 2+15=17. НА ЭКРАНЕ ВЫСВЕТИТСЯ ОТВЕТ-17.

ТАК КАК ОПЕРАЦИИ УМНОЖЕНИЯ И ДЕЛЕНИЯ ВЫПОЛНЯЮТСЯ В ПЕРВУЮ ОЧЕРЕДЬ, МЫ ГОВОРИМ: ЧТО ОНИ ИМЕЮТ БОЛЕЕ ВЫСОКИЙ ПРИОРИТЕТ, ЧЕМ СЛОЖЕНИЕ И ВЫЧИТАНИЕ. МЕЖДУ СОБОЙ ОПЕРАЦИИ УМНОЖЕНИЯ И ДЕЛЕНИЯ ИМЕЮТ ОДИНАКОВЫЙ ПРИОРИТЕТ, ЧТО ОЗНАЧАЕТ, ЧТО ЭТИ ОПЕРАЦИИ ВЫПОЛНЯЮТСЯ ПОСЛЕДОВАТЕЛЬНО, ОДНА ЗА ДРУГОЙ, ПРИ ДВИЖЕНИИ ВДОЛЬ АРИФМЕТИЧЕСКОГО ВЫРАЖЕНИЯ СЛЕВА НАПРАВО.

ДАВАЙТЕ ПОСМОТРИМ, КАК КОМПЬЮТЕР БУДЕТ ВЫПОЛНЯТЬ СЛЕДУЮЩУЮ ИНСТРУКЦИЮ:

PRINT 20-2*9+4/2*3

СНАЧАЛА ВЫПОЛНЯЮТСЯ ОПЕРАЦИИ УМНОЖЕНИЯ И ДЕЛЕНИЯ ПО ПОРЯДКУ, СЛЕВА НАПРАВО, А ЗАТЕМ СЛОЖЕНИЕ И ВЫЧИТАНИЕ

ХОТЯ ВСЕ, ЧТО ВАМ НУЖНО ЗНАТЬ, ЭТО ИМЕЮТ ЛИ ОПЕРАЦИИ ПРИОРИТЕТ ОДНА НАД ДРУГОЙ; КОМПЬЮТЕР ДЕЛАЕТ ЭТО АВТОМАТИЧЕСКИ, ПОСКОЛЬКУ ВНУТРИ СИСТЕМЫ КАЖДОЙ ОПЕРАЦИИ СООТВЕТСТВУЕТ СВОЕ ЗНАЧЕНИЕ ПРИОРИТЕТА ОТ 1 ДО 16. ТАК, НАПРИМЕР, * И / ИМЕЮТ ПРИОРИТЕТ 8, А + И - ПРИОРИТЕТ 6.

ТАКОЕ ПРЯМОЕ ВЫПОЛНЕНИЕ АРИФМЕТИЧЕСКИХ ДЕЙСТВИЙ НЕИЗМЕНЕНО, ОДНАКО МЫ МОЖЕМ ПРЕОДОЛЕТЬ ЭТО, ИСПОЛЬЗУЯ СКОБКИ; ВСЕ, ЧТО ЗАКЛЮЧЕНО В СКОБКИ, ПОДСЧИТЫВАЕТСЯ В ПЕРВУЮ ОЧЕРЕДЬ, А ПОТОМ РАССМАТРИВАЕТСЯ КАК ЕДИНОЕ ЧИСЛО. СЛЕДОВАТЕЛЬНО

PRINT 3*2+2

ДАЕТ ОТВЕТ 6+2=8, НО

PRINT 3*(2+2)

ДАЕТ ОТВЕТ 3*4=12.

ИНОГДА ПОЛЕЗНО ДАВАТЬ КОМПЬЮТЕРУ ДЛЯ ВЫЧИСЛЕНИЯ ТАКИЕ ВЫРАЖЕНИЯ, ПОТОМУ ЧТО, ХОТЯ КОМПЬЮТЕР ОЖИДАЕТ ОТ ВАС ЧИСЛА, А ВЫ ВМЕСТО ЭТОГО ПРЕДЛАГАЕТЕ ЕМУ ВЫРАЖЕНИЕ И ПОЛУЧАЕТЕ ОТВЕТ. ИСКЛЮЧЕНИЯ ИЗ ЭТОГО ПРАВИЛА НАСТОЛЬКО РЕДКИ, ЧТО ОНИ БУДУТ ПОДРОБНО ОБЪЯСНЯТЬСЯ В КАЖДОМ СЛУЧАЕ. СВИ МОЖЕТЕ ПИСАТЬ ЧИСЛА С ДЕСЯТИЧНЫМИ ДРОБЯМИ (ИСПОЛЬЗУЯ ТОЧКУ, А НЕ ЗАПЯТУЮ), И МОЖЕТЕ ТАКЖЕ ИСПОЛЬЗОВАТЬ НАУЧНЫЕ ЗАПИСИ ТАК, КАК ЭТО ПРИНЯТО НА КАРМАННЫХ КАЛЬКУЛЯТОРАХ. ПРИ ЭТОМ ПОСЛЕ ОБЫЧНОГО ЧИСЛА (ЦЕЛОГО ИЛИ ДЕСЯТИЧНОЙ ДРОБИ), ВЫ ПИШИТЕ ЭКСПОНЕНТУ, СОСТОЯЩУЮ ИЗ БУКВЫ E, ЗАТЕМ, ЕСЛИ НУЖНО "-", И ЗАТЕМ ЧИСЛО. ЭКСПОНЕНТА СМЕЩАЕТ ДЕСЯТИЧНУЮ ТОЧКУ ВПРАВО (ИЛИ ВЛЕВО, ДЛЯ ОТРИЦАТЕЛЬНОЙ ЭКСПОНЕНТЫ), УМНОЖАЯ (ИЛИ ДЕЛЯ) ТАКИМ ОБРАЗОМ ЧИСЛО НА 10 НЕКОЛЬКО РАЗ. НАПРИМЕР

2.34E0 = 2.34

2.34E3 = 2340

2.34E-2 = 0.0234

(ПОПРОБУЙТЕ НАПЕЧАТАТЬ ЭТО НА КОМПЬЮТЕРЕ). ЭТО - ОДИН ИЗ ТЕХ СЛУЧАЕВ, КОГДА ВЫ НЕ МОЖЕТЕ ЗАМЕНИТЬ ЧИСЛО ВЫРАЖЕНИЕМ. НАПРИМЕР, ВЫ НЕ МОЖЕТЕ НАПИСАТЬ

(1.34+1) E (6/2).

ВЫ МОЖЕТЕ ТАКЖЕ ИСПОЛЬЗОВАТЬ ВЫРАЖЕНИЯ, ЗНАЧЕНИЯ КОТОРЫХ НЕ ЯВЛЯЮТСЯ ЧИСЛАМИ, А ПОСЛЕДОВАТЕЛЬНЫМИ БУКВЕННЫМИ СИМВОЛАМИ, ЗАКЛЮЧЕННЫМИ В КАВЫЧКИ. ПРОСТЕЙШУЮ ФОРМУЛУ ЭТОГО ВЫ УЖЕ ВИДЕЛИ ТОГДА, КОГДА ПИСАЛИ БУКВЫ В КАВЫЧКАХ ИЛИ ЖЕ ПРОСТО ЧИСЛА БЕЗ КАВЫЧЕК. ОДНАКО ВЫ ЕЩЕ НЕ ЗНАЕТЕ, ЧТО, РАБОТАЯ СО СЛОГАМИ, МОЖНО ИСПОЛЬЗОВАТЬ ЗНАК "+" (НО ТОЛЬКО НЕ -, / ИЛИ *), ТАК ЧТО ЗАЭСЬ НЕ ВСТАЕТ ПРОБЛЕМА ПРИОРИТЕТОВ). ПРИБАВЛЯЯ СЛОГИ ДРУГ К ДРУГУ, МЫ ПОСЛЕДОВАТЕЛЬНО СОЕДИНЯЕМ ИХ ВМЕСТЕ. НАПРИМЕР, ПОПРОБУЙТЕ

PRINT "JORS" + "EY COW"

В ОДНОМ ЕДИНСТВЕННОМ ВЫРАЖЕНИИ ВЫ МОЖЕТЕ СКЛАДЫВАТЬ СКОЛЬКО УГОДНО СЛОВ И МОЖЕТЕ ТАКЖЕ, ПРИ ЖЕЛАНИИ, ИСПОЛЬЗОВАТЬ СКОБКИ.


```
40 INPUT "ENTER DEG F", F
50 PRINT F, (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 КОММЕНТАРИЙ (REMARK ИЛИ REMINDER). ВСЕ, ЧТО СЛЕДУЕТ ПОСЛЕ REM КОМПЬЮТЕРОМ ИГНОРИРУЕТСЯ ДО КОНЦА С.Р.ЖКМ.

ВЫЧИСЛЕНИЯ ДОХОДЯТ ДО СТРОКИ 40 И КОМПЬЮТЕР ПЕРЕХОДИТ В ОЖИДАНИЕ ВВОДА ВАМИ ЗНАЧЕНИЯ ПЕРЕМЕННОЙ F. ВЫ МОЖЕТЕ ВВЕСТИ ЭТО ЗНАЧЕНИЕ В РЕЖИМЕ CLJ. НАБЕРИТЕ ЧИСЛО И НАЖМИТЕ ENTER. КОМПЬЮТЕР ВЫВЕДИТ РЕЗУЛЬТАТ И СНОВА ПЕРХАЕТ 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: 'H STOP IN INPUT, 40:1', КОТОРОЕ ПОЯСНЯЕТ ПРИЧИНУ ОСТАНОВА И МЕСТО ОСТАНОВА (ПЕРВЫЙ ОПЕРАТОР В СТРОКЕ 40).

ЕСЛИ ТЕПЕРЬ ВЫ ХОТЕЛИТЕ ВНОВЬ ПРОДОЛЖИТЬ ВЫПОЛНЕНИЕ ПРОГРАММЫ, ТО ВВЕДИТЕ CONTINUE И КОМПЬЮТЕР ЗАПРОСИТ ОЧЕРЕДНОЕ ЧИСЛО.

ПРИ ИСПОЛЬЗОВАНИИ CONTINUE КОМПЬЮТЕР ЗАПОМИНАЕТ (ДО ВЫДАЧИ ИМ 'O OK') НОМЕР ПОСЛЕДНЕЙ ВЫПОЛНЯВШЕЙСЯ СТРОКИ И ПРОДОЛЖАЕТ ВЫПОЛНЕНИЕ ИМЕННО С ЭТОЙ СТРОКИ.

ПРОСМОТРИТЕ ВНИМАТЕЛЬНО ОПЕРАТОР PRINT В СТРОКЕ 50, ЗАПЯТАЯ В НЕМ ОЧЕНЬ ВАЖНА. ЗАПЯТАЯ В ОПЕРАТОРЕ PRINT ИСПОЛЬЗУЕТСЯ ДЛЯ УКАЗАНИЯ ТОГО, ЧТО ВЫВОД ЗНАКА, СЛЕДУЮЩЕГО 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 ЗНАЧЕНИЯ ТЕМПЕРАТУРЫ В ГРАДУСАХ ЦЕЛЬСИЯ С СЕРЕДИНЫ ЭКРАНА.

ЕСЛИ ИСПОЛЬЗОВАТЬ В ОПЕРАТОРЕ PRINT ВМЕСТО ЗАПЯТОЙ ТОЧКУ С ЗАПЯТОЙ (','), ТО ОЧЕРЕДНЫЕ ДАННЫЕ БУДУТ ВЫВОДИТЬСЯ 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 ДВОЕТОЧЬЕМ (':'), 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 ОТДЕЛЬНО ТОЛЬКО ПОСЛЕДНЮЮ ПРОГРАММУ НАДО ВВЕСТИ КОМАНДУ:

RUN 100

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

N\$ (УДАЛИ КАВЫЧКИ).

ЭТО СДЕЛАЕТ ОПЕРАТОР INPUT В СТРОКЕ 110 ПОДОБНЫМ ОПЕРАТОРУ LET N\$=N\$.

ЕСЛИ ВЫ РЕШИЛИ ВВЕСТИ STOP ПОД СТРОКОВЫМ ВВОД - ТО ДОЛЖНЫ

УСТАНОВИТЬ КУРСОР В НАЧАЛО СТРОКИ, ИСПОЛЬЗУЯ КЛАВИШУ УПРАВЛЕНИЯ КУРСОРОМ (ВЛЕВО).

ДЕЙСТВИЕ КОМАНДЫ 'RUN 100' ПОДОБНО ДЕЙСТВИЮ ОПЕРАТОРА 'GO TO', НО ИМЕЕТСЯ И РАЗЛИЧИЕ. 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, Н.ПРИМЕР:

```
200 GO TO 200
RUN 200
```

ОСТАНОВИТЬ ЭТУ ПРОГРАММУ МОЖНО, ЕСЛИ НАЗАТЬ КЛАВИШИ CAPS SHIFT И SPACE. ЭТО ВЫЗОВЕТ ВВОД КОМАНДЫ 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 'D 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\$ (УДАЛИ КАВЫЧКИ)

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 'O OK:0:1', А ЗАТЕМ ВВЕДЕТЕ CONTINUE, ТО УВИДИТЕ, ЧТО ПРОГРАММА ЗАВЕРШИТСЯ НОРМАЛЬНО.

КАК УЖЕ ОТМЕЧАЛОСЬ, СООБЩЕНИЕ 'L BREAK INTO PROGRAM' ОСОБОЕ, ТАК КАК ВЫАЧА ПОСЛЕ НЕГО CONTINUE НЕ ВЫЗЫВАЕТ ПОВТОРЕНИЕ КОМАНДЫ, ВЫЗВАННОЙ 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 NEW ЧАЩЕ ИСПОЛЬЗУЮТСЯ КАК КОМАНДЫ, НО МОГУТ БЫТЬ ИСПОЛЬЗОВАНЫ И В ПРОГРАММЕ.

ГЛАВА 3

У С Л О В И Я

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

ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ОПЕРАТОРОВ ПРОГРАММЫ НЕ ВСЕГДА ПРЕДСКАЗУЕНА, В ОПРЕДЕЛЕННЫХ МЕСТАХ ПРОГРАММЫ КОМПЬЮТЕР МОЖЕТ ПРИНИМАТЬ РЕШЕНИЕ О ДАЛЕЧЕЙШЕМ ХОДЕ ВЫЧИСЛЕНИЙ. 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 ОПЕРАТОР IF 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 со следующего оператора.

Простейшим 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 используются отношения:

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

Например, выражение 1<2, -2<1, -3<1 истинны, а выражения 1<0, 0<-2 ложны.

Строка программы 40 сравнивает числа 'А' и 'В', и если они равны, завершает работу, выполняя команду STOP. При этом будет выдано сообщение '9 STOP, STATEMENT, 40:3', показывающее, что команда STOP была выдана в 3-ем операторе в 40-ой строке.

Знаки условия набирают на клавиатуре следующим образом:

) - SYMBOL SHIFT вместе с T - больше
< - SYMBOL SHIFT вместе с R - меньше
<= - SYMBOL SHIFT вместе с Q - 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 бы 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 существует специальные операторы FOR и NEXT, которые всегда используются вместе.

Наша программа при использовании 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' управляемая переменная цикла должна иметь имя в одну букву. 'С' последовательно принимает значения 1, 2, 3, 4 и 5 (предел - конечное значение управляемой переменной цикла) и при каждом проходе выполняются строки 30, 40 и 50. Затем после того, как 'С' примет пятое значение, выполнится 70-я строка. Приращение значения управляемой 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 ПРИРОЩ.'.

Здесь 'НАЧАЛЬНОЕ ЗНАЧЕНИЕ', 'ПРЕДЕЛ', 'ШАГ ПРИРОЩЕНИЯ' - есть выражения, принимающие числовое значение. Итак, если бы заменили строку 20 программы на

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

то 'С' последовательно примет 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:

```
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.

ГЛАВА 5

ПОДПРОГРАММЫ

=====

КРАТКОЕ СОДЕРЖАНИЕ: GO SUB, RETURN.

Иногда бывает удобно некоторые фрагменты программы представить в виде отдельных частей, по нескольку 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 похож на 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 стеком.

RETURN выбирает верхний адрес возврата из GO SUB стека и продолжает выполнение программы с оператора следующего после оператора с этим номером.

Приведем пример использования подпрограммы:

```
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 Y=1 TO X
520 LET S=S+Y
530 NEXT Y
540 RETURN
```

В ОБЩЕМ СЛУЧАЕ ПОДПРОГРАММА МОЖЕТ ВЫЗЫВАТЬ ДРУГИЕ ПОДПРОГРАММЫ И ДАЖЕ САМУ СЕБЯ (ТАКАЯ ПОДПРОГРАММА НАЗЫВАЕТСЯ РЕКУРСИВНОЙ).

ГЛАВА 6

О П Е Р А Т О Р Ы READ, DATA И RESTORE

КРАТКОЕ СОДЕРЖАНИЕ: 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 ЭФФЕКТИВНЕЕ ОПЕРАТОРА INPUT, ОСОБЕННО КОГДА ВВОДИМЫЕ ЗНАЧЕНИЯ ПРИСВАИВАЮТСЯ ПЕРЕМЕННЫМ. В ЭТОМ СЛУЧАЕ, КОМПЬЮТЕР ИЩЕТ ВЕЛИЧИНЫ ПЕРЕМЕННЫХ В СПИСКЕ DATA.

КАЖДАЯ ОПЕРАТОР DATA - ЭТО СПИСОК ЗНАЧЕНИЙ - ЧИСЛОВЫХ ИЛИ СТРОЧНЫХ - РАЗДЕЛЕННЫХ ЗАПЯТЫМИ. ВЫ МОЖЕТЕ ВВОДИТЬ ИХ ГДЕ ВАМ УГОДНО В ПРОГРАММЕ, ТАК КАК КОМПЬЮТЕР ИГНОРИРУЕТ ИХ. ЗА ИСКЛЮЧЕНИЕМ ТЕХ СЛУЧАЕВ, КОГДА ИХ ИСПОЛЬЗУЕТ ОПЕРАТОР READ.

ВЫ ДОЛЖНЫ ПРЕДСТАВЛЯТЬ СЕБЕ ВЫРАЖЕНИЯ ДЛЯ ВСЕХ ОПЕРАТОРОВ DATA В ПРОГРАММЕ, ЧТО ПОЗВОЛИТ ВАМ ВВОДИТЬ СОВМЕСТНО ФОРМУ ОДНОГО ДЛИННОГО СПИСКА ВЫРАЖЕНИЙ (СПИСКА DATA). СНАЧАЛА КОМПЬЮТЕР ВЫБРАЕТ ПЕРВОЕ ЗНАЧЕНИЕ ИЗ СПИСКА DATA ДЛЯ ВЕЛИЧИНЫ ИЗ ОПЕРАТОРА READ, В СЛЕДУЮЩИЙ РАЗ ОН БЕРЕТ ВТОРОЕ ЗНАЧЕНИЕ ИЗ СПИСКА DATA, И, ТАКИМ ОБРАЗОМ, ВЫБРАЕМЫЕ 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 ВЫПОЛНЕНЫ В ВЕСЬМА ХАРАКТЕРНОЙ ПРОГРАММЕ:

```
СТРОКА 10 ДАЕТ УКАЗАНИЕ КОМПЬЮТЕРУ ЧИТАТЬ ЗНАЧЕНИЯ ДАННЫХ
И ПРИСВОИТЬ ИХ ПЕРЕМЕННЫМ A, B И C;
СТРОКА 20 ГОВОРИТ О ТОМ, ЧТО НАДО ВЫВЕСТИ (PRINT) ТРИ
ЗНАЧЕНИЯ;
```

В СТРОКЕ 30 ОПЕРАТОР DATA ЗАДАЕТ ЗНАЧЕНИЯ ДЛЯ A, B И C;

СТРОКА 40 ЗАКАНЧИВАЕТ ПРОГРАММУ.

ИНФОРМАЦИЯ В 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 ВСЕ 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 ВЫРАЖЕНИЙ ИЗ СПИСКА DATA: 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,Z НАЧИНАЯ С ПЕРВОГО ЗНАЧЕНИЯ В DATA.

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

ГЛАВА 7

А Р И Ф М Е Т И Ч Е С К И Е О П Е Р А Ц И И

КРАТКОЕ СОДЕРЖАНИЕ: ОПЕРАЦИИ +, -, *, /; ВЫРАЖЕНИЯ, УСЛОВНЫЕ ОБОЗНАЧЕНИЯ, ИМЕНА ПЕРЕМЕННЫХ.

ВЫ УЖЕ ВИДЕЛИ НЕКОЛЬКО ПРИМЕРОВ, В КОТОРЫХ ZX SPESTRUM МОЖЕТ ОПЕРИРОВАТЬ ЧИСЛАМИ. МОЖНО ВЫПОЛНЯТЬ ЧЕТЫРЕ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ: +, -, *, / (ПОМНИТЕ, ЧТО * ИСПОЛЬЗУЕТСЯ ДЛЯ УМНОЖЕНИЯ, А / ИСПОЛЬЗУЕТСЯ ДЛЯ ДЕЛЕНИЯ), И ПРИ ЭТОМ ОПРЕДЕЛЯЕТСЯ ЗНАЧЕНИЕ ПЕРЕМЕННОЙ, ЗАДАВАЕМОЙ ИМЕНЕМ.

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.

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

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

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

ВЫ МОЖЕТЕ ИСПОЛЬЗОВАТЬ ОПЕРАЦИИ СЛОЖЕНИЯ ('+') ДЛЯ СЦЕПЛЕНИЯ СТРОК (КОНКАТЕНЦИИ) В ВЫРАЖЕНИЯХ.

ИМЯ СТРОКОВОЙ ПЕРЕМЕННОЙ СОСТОИТ ИЗ БУКВ С ПОСЛЕДУЮЩИМИ ЗНАКАМИ '\$', 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 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 IT OUT AGAIN WITHOUT MAKING A MISTAKE
NOW WE ARE SIX \ ЭТИ ДВА ИМЕНА УКАЗЫВАЮТ
NOWWEARESIX / НА ОДНУ И ТУ ЖЕ ПЕРЕМЕННУЮ

ПРИМЕРЫ НЕДОПУСТИМЫХ ИМЕН ПЕРЕМЕННЫХ:

2001

(НАЧИНАЕТСЯ С ЦИФРЫ)

3 BEARS

(НАЧИНАЕТСЯ С ЦИФРЫ)

MY*AS*H

(ЗНАК '*' - НЕ БУКВА И НЕ ЦИФРА)

FOTHERINGTON-THOMAS (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 2.34E15

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

PRINT 4294967295, 4294967295-429E7

И ВЫ УВИДИТЕ, ЧТО КОМПЬЮТЕР МОЖЕТ ВОСПРИНЯТЬ ТОЛЬКО ЦИФРЫ 42949672.

КОМПЬЮТЕР ZX SPECTRUM ИСПОЛЬЗУЕТ АРИФМЕТИКУ С ПЛАВЮЩЕЙ ТОЧКОЙ (ЗАПЯТОЙ). ПРИ ЭТОМ РАЗЛИЧНЫЕ ЧАСТИ ЧИСЛА (МАНТИССА 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 ЕДИНИЦЫ ФАКТИЧЕСКИ ОКРУГЛИТСЯ ДО 2.

ЧИСЛА 5E9 + 1 И 5E9 + 2 ДЛЯ КОМПЬЮТЕРА РАВНЫ. НАИБОЛЬШЕЕ ЦЕЛОЕ, КОТОРОЕ МОЖЕТ ВОСПРИНЯТЬ КОМПЬЮТЕР, РАВНО 2**32 - 1 ИЛИ (4 294 967 295).

ГЛАВА В

СТРОКИ СИМВОЛОВ

КРАТКОЕ СОДЕРЖАНИЕ: СЕЧЕНИЕ, ИСПОЛЬЗОВАНИЕ ТО.

ПРИМЕЧАНИЕ: ЭТИ ОПЕРАЦИИ ОТСУТСТВУЮТ В СТАНДАРТНОМ БЕЙСИКЕ. СТРОКА "" БЕЗ ЕДИНГО СИМВОЛА НАЗЫВАЕТСЯ ПУСТОЙ ИЛИ НУЛЕВОЙ СТРОКОЙ. НЕ ПУТАЙТЕ ЕЕ С ПРОБЕЛОМ. НАВЕРИТЕ:

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

КОГДА ВЫ НАЖИТЕ КЛАВИШУ ENTER, ВЫ ПОЛУЧИТЕ МЕРЦАЮЩИЙ ЗНАК ВОПРОСА, УКАЗЫВАЮЩИЙ ОШИБОЧНОЕ МЕСТО В СТРОКЕ. КОГДА, ПРИ ИНТЕРПРЕТАЦИИ ЭТОЙ СТРОКИ, КОМПЬЮТЕР НАЙДЕТ ДВОЙНУЮ КАВЫЧКУ, ОТКРЫВАЮЩУЮ "FINNEGANS WAKE", ТО СОЧТЕТ ЕЕ ЗАКРЫВАЮЩЕЙ КАВЫЧКОЙ ДЛЯ СТРОКИ "HAVE YOU FINISHED" И ЗАТЕМ СКОЖЕТ ВЫВЕСТИ "FINNEGANS WAKE". ЗДЕСЬ НАДО ПОМНИТЬ СПЕЦИАЛЬНОЕ ПРАВИЛО: ЕСЛИ ВЫ ХОТИТЕ ВЫВЕСТИ КАВЫЧКИ ВНУТРИ СТРОКИ, ОНИ ДОЛЖНЫ УДАВЛЯТЬСЯ. НАПРИМЕР: PRINT "HAVE YOU FINISHED""FINNEGANS WAKE"" YET?"

ПУСТЬ ИМЕЕТСЯ СТРОКА СИМВОЛОВ, ТОГДА ЕЕ ПОДСТРОКА БУДЕТ НЕКОТОРАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ ИЗ ЭТОЙ СТРОКИ. ТАК "STRING" ЯВЛЯЕТСЯ ПОДСТРОКОЙ ОТ "BIGGER STRING", А "B STRING" И "BIG STRING" НЕ ЯВЛЯЮТСЯ.

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:

"ABCDE" (2 TO 5) = "BCDE"

ЕСЛИ ОПУЩЕНО 'НАЧАЛО', ТО УМОЛЧАНИЕ ПОДРАЗУМЕВАЕТСЯ 1, ЕСЛИ - 'КОНЕЦ', ТО ПОДРАЗУМЕВАЕТСЯ ДЛИНА ВСЕЙ СТРОКИ. ТАК:

"ABCDE" (TO 5) = "ABCDE" (1 TO 5) = "ABCDE"

"ABCDE" (2 TO) = "ABCDE" (2 TO 6) = "BCDE"

"ABCDE" (TO) = "ABCDE" (1 TO 6) = "ABCDE"

ПОСЛЕДНЕЕ ВЫРАЖЕНИЕ МОЖНО БЫЛО ВЫ ЗАПИСАТЬ И ТАК:

"ABCDE" ()

ЧТО ТОЖЕ ВЕРНО. МОЖНО ОПУСКАТЬ И СЛОВО TO :

"ABCDE" (3) = "ABCDE" (3 TO 3) = "C"

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

"ABCDE" (5 TO 7)

ВЫЗЫВАЕТ СООБЩЕНИЕ '3 SUBSCRIPT WRONG', ТАК КАК 'КОНЕЦ' ПРЕВЫШАЕТ ДЛИНУ СТРОКИ (6).

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

"ABCDE" (8 TO 7) = ""

"ABCDE" (1 TO 0) = ""

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

СЛЕДУЮЩАЯ ПРОГРАММА ИЛЛЮСТРИРУЕТ ЭТИ ПРАВИЛА:

10 LET A\$ = "ABCDE"

20 FOR N=1 TO 6

30 PRINT A\$(N TO 6)

40 NEXT N

50 STOP

МОЖНО ТАКЖЕ ПРИСВАИВАТЬ ЗНАЧЕНИЯ ПОДСТРОКЕ. ПОПРОБУЙТЕ:

10 LET A\$ = "I AM THE ZX SPECTRUM"

20 PRINT A\$

30 LET A\$(5 TO 9) = "*****"

40 PRINT A\$

ПОДСТРОКА A\$(5 TO 9) ИМЕЕТ ДЛИНУ ТОЛЬКО В 4 СИМВОЛА, ПОЭТОМУ БУДУТ ИСПОЛЬЗОВАНЫ ТОЛЬКО ЧЕТЫРЕ ЗВЕЗДОЧКИ. ЭТО ОСОБЕННОСТЬ ПРИСВОЕНИЯ ЗНАЧЕНИЯ ПОДСТРОКЕ: ДЛИННЫЕ ДАННЫЕ УСЕКАЮТСЯ СПРАВА, А КОРОТКИЕ ДОПОЛНЯЮТСЯ ПРОБЕЛАМИ ДО ДЛИНЫ ПОДСТРОКИ. ЭТО ДЕЙСТВИЕ НАЗЫВАЕТ ПРОКР/СТИАНОМ В ЧЕСТЬ НИФИЧЕСКОГО РАЗВОЙНИКА ПРОКРУСТА.

ЕСЛИ ВЫ ТЕПЕРЬ ВЫПОЛНИТЕ
LET A\$ = "HELLO THERE"

И

PRINT A\$; ". "

ВЫ УВИДИТЕ, ЧТО БУДУТ ВЫВЕДЕНЫ ДОПОЛНИТЕЛЬНЫЕ ПРОБЕЛЫ, ТАК КАК 'A\$()' СЧИТАЕТСЯ ПОДСТРОКОЙ. ДЛЯ ПРАВИЛЬНОГО ВЫПОЛНЕНИЯ 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 ЗАДАНИЕ В БЕЙСИК-СИСТЕМУ ПОДПРОГРАММЫ, КОТОРЫЕ, ПОЛУЧАЯ НА ВХОДЕ 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 НЕЕ СТРОКОВОГО АРГУМЕНТА. ВЫ МОЖЕТЕ ЗАПИСАТЬ:

PRINT LEN "SINCLAIR"

А КОМПЬЮТЕР ВЫВЕДЕТ ОТВЕТ '9', Т.Е. КОЛИЧЕСТВО БУКВ В СЛОВЕ 'SINCLAIR' (ДЛЯ ВВОДА С КЛАВИАТУРЫ ФУНКЦИИ LEN, ВЫ ДОЛЖНЫ ВОЙТИ В НЕОБХОДИМЫЙ РЕЖИМ, НАЖАВ КЛАВИШИ CAPS SHIFT И SYMBOL SHIFT, КУРСОР ИЗМЕНИТСЯ С CLJ НА CEJ, И НАЖАТЬ КЛАВИШУ K).

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

ФУНКЦИЯ STR\$ ПРЕОБРАЗУЕТ ЧИСЛО В СИМВОЛЬНЫЙ ВИД, ПОДЛЕВШИЙ ФОРМАТУ ВЫВОДА ЧИСЕЛ ОПЕРАТОРОМ PRINT:

LET A\$ = STR\$ 1E2

АНАЛОГИЧНО ПО ДЕЙСТВИЮ КОМАНДЕ

LET A\$ = "100"

ИЛИ ВЫПОЛНИТЕ

PRINT LEN STR\$ 100.0000

И ПОЛУЧИТЕ ОТВЕТ 3, ТАК КАК STR\$ 100.0000 = "100".

ФУНКЦИЯ VAL ОБРАТНАЯ К ФУНКЦИИ STR\$ И ПРЕОБРАЗУЕТ СТРОКУ 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 ПОЯВИТЬСЯ УЧЕТВЕРЕНИЕ ИЛИ ДАЖЕ УВОСЬМИРЕНИЕ.

ИМЕЕТСЯ ЕЩЕ ОДНА ФУНКЦИЯ ПОДОБНАЯ VAL - ЭТО VAL:1 И АРГУМЕНТОМ, И РЕЗУЛЬТАТОМ ЭТОЙ ФУНКЦИИ ЯВЛЯЕТСЯ СТРОКА СИМВОЛЫ. ОНА РАБОТАЕТ КАК VAL, ПРИМЕНЕННАЯ ДВАЖЫМ, РАСКРЫВАЯ ВСЕ КАВЫЧКИ В СТРОКАХ:

VAL\$ " " " FRUIT PUNCH" " " = "FRUIT PUNCH"
СДЕЛАЙТЕ

LET A\$ = "99"

И ЗАТЕМ ВЫВЕДЕТЕ ВСЕ СЛЕДУЮЩИЕ ЗНАЧЕНИЯ:

VAL A\$

VAL "A\$"

VAL " " "A\$" " " "

VAL\$ A\$

VAL\$ "A\$"

VAL\$ " " "A\$" " " "

НЕКОТОРЫЕ ИЗ НИХ СРАБОТАЮТ, А НЕКОТОРЫЕ НЕТ, ПРОАНАЛИЗИРУЙТЕ ВСЕ ОТВЕТЫ.

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

1, ЕСЛИ АРГУМЕНТ ПОЛОЖИТЕЛЕН;

0, ЕСЛИ АРГУМЕНТ РАВЕН 0;

-1, ЕСЛИ АРГУМЕНТ ОТРИЦАТЕЛЕН.

ФУНКЦИЯ ABS ПРЕОБРАЗУЕТ АРГУМЕНТ В ПОЛОЖИТЕЛЬНОЕ ЧИСЛО:

ABS -3.2 = 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 ВОЗНИКАЮТ ПРИ ОТРИЦАТЕЛЬНОМ АРГУМЕНТЕ, ТАК КАК ОКРУГЛЕНИЕ ПРОИСХОДИТ К БЛИЖАЙШЕМУ ЦЕЛОМУ НЕ БОЛЬШЕМУ, ЧЕМ АРГУМЕНТ:

INT -3.9 = -4

ФУНКЦИЯ SQR ВЫЧИСЛЯЕТ КОРЕНЬ КВАДРАТНЫЙ ОТ ЧИСЛА, НАПРИМЕР:

SQR 4 = 2

SQR 0.25 = 0.5

SQR 2 = 1.4142136 (ПРИБЛИЖЕННО).

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 ЗАКЛЮЧЕННЫМИ В СКОБКИ.

ВЫ МОЖЕТЕ ОПРЕДЕЛИТЬ ФУНКЦИЮ ВВОДОМ ОПЕРАТОРА DEF В НЕКОТОРОМ МЕСТЕ ПРОГРАММЫ. НАПРИМЕР, ЗАДАЛИМ ФУНКЦИЮ, ВЫЧИСЛЯЮЩУЮ КВАДРАТ ЧИСЛА:

10 DEF FN S(X) = X*X: REM THE SQUARE OF X

DEF ВВОДИТСЯ В РЕЖИМ Е НАЖАТИЕМ КЛАВИШ SYMBOL SHIFT И 1. ТЕПЕРЬ ФУНКЦИЯ МОЖЕТ ИСПОЛЬЗОВАТЬСЯ В ПРОГРАММЕ:

PRINT FN S(2)

PRINT FN S(3+4)

PRINT 1 + INT FN S(LEN "CHICKEN"/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:

20 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 Q(X)=A + X*Y

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

ЕСТЬ ОДНА ТОЧНОСТЬ В ЭТОЙ ПРОГРАММЕ. ВО-ПЕРВЫХ, ФУНКЦИЯ FN Q НЕ ИСПОЛЬЗУЕТ АРГУМЕНТОВ, НО СКОБКИ ПРИ ЭТОМ ДЛ ДН ОБЯЗАТЕЛЬНО ИСПОЛЬЗОВАТЬСЯ. ВО-ВТОРЫХ, ОПЕРАТОРЫ DEF НЕ ВЫПОЛНЯЕМЫЕ, 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 И ОПРЕДЕЛЕНО В ПРОГРАММЕ, А ЗНАЧЕНИЯ 'X' И 'Y' СООТВЕТСТВЕННО 2 И 3, ТАК КАК ОНИ АРГУМЕНТЫ И РЕЗУЛЬТАТ БУДЕТ $10 + 2 \times 3 = 16$.

ПРИ ВЫЧИСЛЕНИИ FN Q() УЧАСТВУЮТ ТОЛЬКО ПЕРЕМЕННЫЕ ПРОГРАММЫ, ТАК КАК АРГУМЕНТОВ НЕТ, И ОТВЕТ В ЭТОМ СЛУЧАЕ БУДЕТ $10 + 0 \times 0 = 10$. ТЕПЕРЬ ИЗМЕНИМ СТРОКУ 20 НА

```
20 DEF FN P(X,Y) = FN Q()
```

В ЭТОМ СЛУЧАЕ FN P(2,3) БУДЕТ ВОЗВРАЩАТЬ ЗНАЧЕНИЕ 10.

НЕКОТОРЫЕ ВЕРСИИ БЕСИКА ИМЕЮТ ФУНКЦИИ LEFT\$, RIGHT\$ И 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.

П Р И М Е Ч А Н И Е. ФУНКЦИЯ МОЖЕТ ИМЕТЬ ДО 25 ЧИСЛОВЫХ АРГУМЕНТОВ И В ТО ЖЕ 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. СПОЛНЕ ВОЗМОЖНО, ЧТО ВАМ НИКОГДА НЕ ПРИДЕТСЯ ВОСПОЛЬЗОВАТЬСЯ НИМИ И ЕСЛИ ВЫ СЧЕТАЕТЕ ИХ СЛИШКОМ СЛОЖНЫМИ, МОЖЕТЕ ПРОПУСТИТЬ ЭТУ ГЛАВУ. ВСЕ СКАЗАННОЕ ОТНОСИТСЯ К ФУНКЦИЯМ: ** (ВОЗВЕДЕНИЕ В СТЕПЕНЬ), EXP, LN, ТРИГОНОМЕТРИЧЕСКИМ ФУНКЦИЯМ: SIN, COS, TAN, И ОБРАТНЫМ К НИМ: ASN, ACS, ATN.

** И EXP

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

```
2 ** 1 = 2
```

```
2 ** 2 = 2 * 2 = 4 (ДВА В КВАДРАТЕ)
```

```
2 ** 3 = 2 * 2 * 2 = 8 (ДВА В КУБЕ)
```

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 ПОНИМАТЬ, 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:

```
A ** 0 = 1.
```

```
A ** (-B) = 1 / (A ** B)
```

```
A ** (1/B) = КОРЕНЬ B-ОЙ СТЕПЕНИ ИЗ A
```

```
A ** (B*C) = (A ** B) ** C
```

ПОЛЕЗНО ПОМНИТЬ, ЧТО:

```
A ** (-1) = 1/A
```

```
A ** (1/2) = SQRT A
```

ПОЭКСПЕРИМЕНТИРУЙТЕ С ЭТИМ, ПОПРОБУЙТЕ ВЫПОЛНИТЬ ТАКУЮ ПРОГРАММУ:

```
10 INPUT A,B,C
```

```
20 PRINT A**(B+C),A**B * A**C
```

```
30 GO TO 10
```

КОМПЬЮТЕР СТАНЕТ ВЫВОДИТЬ ДВА ОДИНАКОВЫХ 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 ПРИВЫСЬ, 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 ОПЕРАТОР:

```
FOR Y=0 TO 10:PRINT Y,10*1.15**Y:NEXT Y
```

ВЫ УВИДИТЕ, ЧТО НАЧИНАЯ С 10 ФУНТОВ МОЖНО ПОЛУЧИТЬ ВСЕ БОЛЬШИЕ И БОЛЬШИЕ ДОХОД С КАПИТАЛА.

ТАКОМ ТИП ПОВЕДЕНИЯ ФУНКЦИИ, КОГДА ПОСЛЕ ФИКСИРОВАННОГО ЧИСЛА ИНТЕРВАЛОВ ВРЕМЕНИ, ЗНАЧЕНИЯ ФУНКЦИИ ПРОПОРЦИОНАЛЬНЫ КОЛИЧЕСТВУ УМНОЖЕНИЯ ЭТОГО ЧИСЛА САМОГО НА СЕБЯ, НАЗЫВАЕТСЯ ЭКСПОНЕНЦИАЛЬНЫМ ЗАКОНОМ.

ПРЕДПОЛОЖИМ ВЫ ЗАПИСАЛИ:

```
10 DEF FN A(X)=A**X
```

ЗДЕСЬ A ОПРЕДЕЛЕНО, В ОПЕРАТОРЕ LET, ЕГО ЗНАЧЕНИЕ ПЕРЕДАЕТСЯ ДЛЯ ВЫЧИСЛЕНИЯ СТЕПЕНИ.

ИМЕЕТСЯ ОПРЕДЕЛЕННОЕ ЗНАЧЕНИЕ A, КОТОРОЕ ДЕЛАЕТ ФУНКЦИЮ FN A ИДЕНТИФИЦИРОВАННОЙ, СПЕЦИАЛЬНУЮ МАТЕМАТИЧЕСКУЮ ФУНКЦИЮ. ЭТО ЗНАЧЕНИЕ НАЗЫВАЕТСЯ 'E'.

ZX SPECTRUM ИМЕЕТ СПЕЦИАЛЬНУЮ ФУНКЦИЮ, НАЗЫВАЕМУЮ EXP И ОПРЕДЕЛЯЕМУЮ КАК:

```
EXP X = E ** X
```

К СОЖАЛЕНИЮ, 'E' НЕ МОЖЕТ БЫТЬ ПРЕДСТАВЛЕНО ТОЧНЫМ ЧИСЛОМ. ВЫ МОЖЕТЕ УВИДЕТЬ ПЯТЬ ЕГО ПЕРВЫХ ДЕСЯТИЧНЫХ ЗНАКОВ, ВЫПОЛНИВ

```
PRINT EXP 1
```

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

LN

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

ВАМ ДОЛЖНО БЫТЬ ИЗВЕСТНО КАК ИСПОЛЬЗУЕТСЯ ЛОГАРИФМ ПО ОСНОВАНИЮ 10 ДЛЯ УМНОЖЕНИЯ, ТАКОМ ЛОГАРИФМ НАЗЫВАЕТСЯ ДЕСЯТИЧНЫМ. ZX SPECTRUM ИМЕЕТ ФУНКЦИЮ LN, КОТОРАЯ ВЫЧИСЛЯЕТ ЛОГАРИФМ ПО ОСНОВАНИЮ 'E', НАЗЫВАЕМЫЙ НАТУРАЛЬНЫМ. ДЛЯ ВЫЧИСЛЕНИЯ ЛОГАРИФМА С ДРУГИМ ОСНОВАНИЕМ, НАДО РАЗДЕЛИТЬ НАТУРАЛЬНЫЙ ЛОГАРИФМ ИСХОДНОГО ЧИСЛА НА НАТУРАЛЬНЫЙ ЛОГАРИФМ ОСНОВАНИЯ:

```
LOGA X = LN X / LN A
```

PI

ДОПУСТИМ ИМЕЕТСЯ НЕКОТОРЫЙ КРУГ. ВЫ МОЖЕТЕ НАЙТИ ЕГО ПАРАМЕТР (ДЛИНУ ОКРУЖНОСТИ), УМНОЖИВ ЕГО ДИАМЕТР НА ЧИСЛО, НАЗЫВАЕМОЕ PI. ПОДОБНО 'E', PI ПРЕСТАВЛЯЕТСЯ ВЕСКОНЕЧНОЙ ДЕСЯТИЧНОЙ ДРОБЬЮ, ЕГО НАЧАЛО:

3,141592653589...

СЛОВО 'PI' В ZX SPECTRUM ОБОЗНАЧАЕТ ЭТО ЧИСЛО.

ВЫПОЛНИТЕ, НАПРИМЕР:

PRINT PI

SIN, COS, TAN И ACS, ATN.

ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ ОПИСЫВАЮТ ТЕ СЛУЧАИ, КОГДА ТОЧКА ПЕРЕМЕЩАЕТСЯ ВОКРУГ ОКРУЖНОСТИ ЕДИНИЧНОГО РАДИУСА. ТОЧКА СТАНОВИТСЯ С ПОЗИЦИИ 3-Х ЧАСОВ И ПЕРЕМЕЩАЕТСЯ ПРОТИВ ЧАСОВОЙ СТРЕЛКИ. НАЧАЛО КООРДИНАТ НАХОДИТСЯ В ЦЕНТРЕ ЭТОЙ ОКРУЖНОСТИ. ТОГДА SIN УГЛА МЕЖДУ РАДИУСОМ, СОЕДИНЯЮЩИМ ДВИЖУЩУСЯ ПО ОКРУЖНОСТИ ТОЧКУ С НАЧАЛОМ КООРДИНАТ, БУДЕТ ОРИНТАТА ЭТОЙ ТОЧКИ, А COS - АБСЦИССА. НЕОБХОДИМО ПОМНИТЬ, ЧТО ЕСЛИ ТОЧКА НАХОДИТСЯ СЛЕВА ОТ ОСИ Y, ТО КОСИНУС ОТРИЦАТЕЛЬНЫЙ, А ЕСЛИ ТОЧКА НАХОДИТСЯ ПРАВО ОТ ОСИ X, ТО ОТРИЦАТЕЛЬНЫЙ СИНУС. НЕОБХОДИМО ТАК ЖЕ ПОМНИТЬ, ЧТО:

$\sin(A + 2\pi) = \sin A$

$\cos(A + 2\pi) = \cos A$

ИМЕЮТСЯ И ДРУГИЕ ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ:

TAN - ТАНГЕНС

ASN - АРККИНУС

ACS - АРККОСИНУС

ATN - АРКТАНГЕНС

ПОМНИТЕ, В ZX SPECTRUM ТРИГОНОМЕТРИЧЕСКИЕ ФУНКЦИИ ВЫЧИСЛЯЮТСЯ 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 НИКОГДА 1). ПОПРОБУЙТЕ ВЫПОЛНИТЬ:

10 PRINT RND

20 GO TO 10

ВЫ УВИДИТЕ, КАК МЕНЯЕТСЯ РЕЗУЛЬТАТ. ФАКТИЧЕСКИ RND НЕ АБСОЛЮТНО СЛУЧАЙНОЕ ЧИСЛО, А ВЫБИРАЕТСЯ ИЗ ОПРЕДЕЛЕННОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ДЛИНОЙ В 65536 ЧИСЕЛ, ПОЭТОМУ ОБЫЧНО ГОВОРЯТ, ЧТО RND - ПСЕВДОСЛУЧАЙНОЕ ЧИСЛО.

ДЛЯ ПОЛУЧЕНИЯ СЛУЧАЙНОГО ЧИСЛА В ИНТЕРВАЛЕ ОТЛИЧНОМ ОТ 0...1 МОЖНО ИСПОЛЬЗОВАТЬ ВЫРАЖЕНИЯ, НАПРИМЕР:

$1.3 + 0.7 * \text{RND}$

ДАСТ ИНТЕРВАЛ ОТ 1.3 ДО 2.

ДЛЯ ПОЛУЧЕНИЯ СЛУЧАЙНЫХ ЦЕЛЫХ ЧИСЕЛ ИСПОЛЬЗУЙТЕ ФУНКЦИЮ INT (ОКРУГЛЯЕТ С ОТБРАСЫВАНИЕМ ДРОБНОЙ ЧАСТИ). НАПРИМЕР:

$1 + \text{INT}(\text{RND} * 6)$

БУДЕТ ДАВАТЬ ЧИСЛА 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 9.0022735596.

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 ИСПОЛЬЗУЮТ RND И RANDOMIZE ДЛЯ ГЕНЕРАЦИИ СЛУЧАЙНЫХ ЧИСЕЛ, НО ЭТО НЕ ЕДИНСТВЕННОЕ ИХ ПРИМЕНЕНИЕ.

НИЖЕ ПРИВОДИТСЯ ТЕКСТ ПРОГРАММЫ МОДЕЛИРУЮЩЕЙ ВЫБРАСЫВАНИЕ КОИНЫ И ПОДСЧЕТА ЧИСЛА ВЫПАДЕНИЯ 'ОРЛА' И 'РЕШКИ'.

(ПЕРЕВОД ИМЕН ПРОГРАММЫ: HEADS-ОРЛЫ, TAILS-РЕШКИ, COIN-КОИНА):

10 LET HEADS=0:LET TAILS=0

20 LET COIN=INT(RND*2)

30 IF COIN=0 THEN LET HEADS=HEADS+1

40 IF COIN=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 'ОРЛОВ' 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 ЗАВЕСТИ ПЕРЕМЕННУЮ НА КАЖДОГО ЧЕЛОВЕКА. ЭТО НЕ УДОБНО, ТАК КАК ПРИХОДИТСЯ ОБРАЩАТЬСЯ К ДАННЫМ, НАЗЫВАЯ КАЖДЫЙ РАЗ НОВУЮ ПЕРЕМЕННУЮ. НАПРИМЕР BLOGG51, BLOGG52 И Т.Д. ДО BLOGG510.

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 'B' (ПОДОБНО УПРАВЛЯЮЩИМ ПЕРЕМЕННЫМ В FOR - NEXT УТВЕРЖДЕНИЯХ, ИМЯ МАССИВА ДОЛЖНО БЫТЬ УНИКАЛЬНЫМ В ДАННОЙ ПРОГРАММЕ), И ДЕСЯТЬ ПЕРЕМЕННЫХ БУДУТ B(1), B(2) И Т.Д. ДО B(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 B(10), КОТОРЫЙ ОПРЕДЕЛЯЕТ МАССИВ С ИМЕНЕМ 'B' И РАЗМЕРНОСТЬЮ 10 (т.е. 10 ИНДЕКСИРУЕМЫХ ПЕРЕМЕННЫХ B(1), B(2), ..., B(10) И ПРИСВАИВАЕТ ВСЕМ ЭЛЕМЕНТАМ МАССИВА ЗНАЧЕНИЕ 0. ИТАК, ТЕПЕРЬ МЫ МОЖЕМ ЗАПИСАТЬ:

11

```

5 DIM B(10)
10 FOR N=1 TO 10
20 READ B(N)
30 NEXT N
40 DATA 10,2,5,19,3,11,1,0,6

```

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. ТАКОЕ МАССИВ КАК БЫ 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 'C' С РАЗМЕРНОСТЬЮ 3 И 6:

```
DIM C(3,6)
```

ЧТО ДАСТ 3*6=18 ИНДЕКСИРУЕМЫХ ПЕРЕМЕННЫХ:

```
C(1,1), C(1,2), ..., C(1,6)
```

```
C(2,1), C(2,2), ..., C(2,6)
```

```
C(3,1), C(3,2), ..., C(3,6)
```

МОГУТ БЫТЬ ТАКЖЕ СТРОКОВЫЕ МАССИВЫ. СТРОКИ В ТАКИХ МАССИВАХ ОТЛИЧАЮТСЯ ОТ СКАЛЯРНЫХ ТЕМ, ЧТО ИМЕЮТ ФИКСИРОВАННУЮ ДЛИНУ, А ПРИСВАИВАНИЕ ИЛИ ЗНАЧЕНИЯ ОСУЩЕСТВЛЯЕТСЯ С УСЕЧЕНИЕМ СПРАВА ИЛИ ДОБАВЛЕНИЕМ ДО ПОЛНОЙ ДЛИНЫ ПРОБЕЛАМИ. ИМЯ СТРОКОВОГО МАССИВА ОБРАЗУЕТСЯ ДОБАВЛЕНИЕМ СПРАВА К ИМЕНИ СПЕЦИАЛЬНОГО СИМВОЛА '\$'.

ДОПУСТИМ, ВАМ НЕОБХОДИМО ОБЪЯВИТЬ МАССИВ A\$ НА 5 СТРОК ПО 10 СИМВОЛОВ В КАЖДОЙ, ВЫ ДОЛЖНЫ ЗАПИСАТЬ:

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

ТЕПЕРЬ ВЫ МОЖЕТЕ ОБРАЩАТЬСЯ КАК ЦЕЛЫМ 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\$(2,10). СЛЕДУЮЩАЯ ПРОГРАММА:

```
5 DIM A$(2,10)
```

```
10 LET A$(2)="1234567890"
```

```
20 PRINT A$(2),A$(2,7)
```

ДАСТ '1234567890 7'

МОЖНО ИСПОЛЬЗОВАТЬ ТАКЖЕ СЕЧЕНИЕ 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 ВЗГЛЯДЕМ 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, 'AND', НЕКОТОРОЕ ДРУГОЕ ВЫРАЖЕНИЕ ИСТИНЫ, ЕСЛИ ИСТИНЫ 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, 'OR', НЕКОТОРОЕ ДРУГОЕ ВЫРАЖЕНИЕ ИСТИНЫ, ЕСЛИ ИСТИНО ХОТЯ БЫ ОДНО ИЗ ЭТИХ ВЫРАЖЕНИЙ. 'NOT', ВЫРАЖЕНИЕ ИСТИНО, ЕСЛИ ЛОЖНО САМО ВЫРАЖЕНИЕ И НАОБОРОТ.

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

УСЛОВИЕ '< >' ОБРАТНО В ЛОГИЧЕСКОМ СМЫСЛЕ УСЛОВИЮ '= ', ТО ЕСТЬ:

A<>B ТО ЖЕ, ЧТО И NOT A=B

NOT A<>B ТО ЖЕ, ЧТО И A=B

ТЕМ, КТО БОИТСЯ СЛОЖНОСТЕЙ, СЛЕДУЮЩИЕ РАЗДЕЛЫ МОЖНО ОПУСТИТЬ

1. УСЛОВИЯ =, <, >, <=, >=, <=, >= ДАЮТ ЧИСЛОВЫЙ РЕЗУЛЬТАТ 1 ДЛЯ ИСТИНЫ И 0 ЕСЛИ ЛОЖЬ. НАПРИМЕР, ОПЕРАТОР PRINT 1=2, 1<2 ВЫВЕДЕТ 0 ДЛЯ '1=2', КОТОРОЕ ЛОЖНО И 1 ДЛЯ '1<2', КОТОРОЕ ИСТИНО.

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

```
LET TOTAL PRICE=PRICE LESS TAX*(1.15 OR V$="ZERO RATED")
```

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

X\$ AND Y\$ ИМЕЕТ ЗНАЧЕНИЕ X\$, ЕСЛИ Y\$ ≠ "" И "", ЕСЛИ Y\$=""

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$;" ";C$ AND A$<B$+(" " AND A$=B$);
```

```
40 PRINT " "B$
```

```
50 GO TO 10
```

ГЛАВА 14

НАБОР СИМВОЛОВ

КРАТКОЕ СОДЕРЖАНИЕ: CODE, CHR\$, POKE, PEEK, 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 ПРИВЕДЕНЫ 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 ОТБРАЖАЕМЫЕ СИМВОЛЬНЫЕ НАБОР: 10 FOR A=32 TO 127:PRINT CHR\$ A:NEXT A

ВСЕ ЭТИ СИМВОЛЫ (КРОМЕ ЗНАКА ФУНТА И 'C' В КРУЖОЧКЕ) ОБРАЗУЮТ КОД ASCII (AMERICAN STANDARD CODES FOR INFORMATION

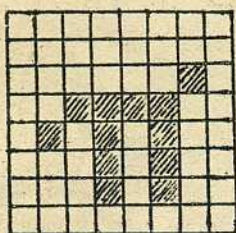
INTERCHANGE).

СЛЕДУЮЩИЕ СИМВОЛЫ НЕ ВХОДЯТ В ASCII, НО ИСПОЛЗУЮТСЯ В ZX SPECTRUM. ПЕРВЫЕ ИЗ НИХ - ЭТО 15 ЧЕРНО-БЕЛЫХ ЗНАЧКОВ, НАЗЫВАЕМЫХ ГРАФИЧЕСКИМИ СИМВОЛАМИ И ИСПОЛЗУЕМЫХ ДЛЯ ИЗОБРАЖЕНИЯ РИСУНКОВ. ИХ МОЖНО ВВЕСТИ С КЛАВИАТУРЫ, ИСПОЛЗУЯ ТАК НАЗЫВАЕМЫЙ ГРАФИЧЕСКИЙ РЕЖИМ. ЕСЛИ ВЫ НАЖИТЕ GRAPHICS (CAPS SHIFT 9), ТО КУРСОР ИЗМЕНИТСЯ НА (G). ТЕПЕРЬ ЦИФРОВЫЕ КЛАВИШИ 0-1 ПО В ВЫДАЮТ ГРАФИЧЕСКИЕ СИМВОЛЫ, ОБОЗНАЧЕННЫЕ НА КЛАВИШАХ, А ЕСЛИ ПРИ ЭТОМ УДЕРЖИВАТЬ SHIFT, ТО ОНИ БУДУТ ВЫДАВАТЬ ИНВЕРСИЮ СИМВОЛЫ, Т.Е. ЧЕРНОЕ СТАНОВИТСЯ БЕЛЫМ, А БЕЛОЕ - ЧЕРНЫМ.

НЕЗАВИСИМО ОТ SHIFT, КЛАВИША С ЦИФРОЙ '9' ОБЕСПЕЧИВАЕТ ВАМ ВОЗВРАТ К ОБЫЧНОМУ РЕЖИМУ (КУРСОР (L)), А КЛАВИША '0' ФУНКЦИЕЙ DELETE.

ПОСЛЕ ГРАФИЧЕСКИХ СИМВОЛОВ НА КЛАВИАТУРЕ РАСПОЛАГАЮТСЯ СИМВОЛЫ АЛФАВИТА ОТ A ДО U. ГРАФИЧЕСКИЕ ЗНАЧЕНИЯ ЭТИХ КЛАВИШ МОГУТ ОПРЕДЕЛЯТЬСЯ САМИ ПОЛЬЗОВАТЕЛЕМ, А ЗАТЕМ ИСПОЛЗОВАТЬСЯ В ГРАФИЧЕСКОМ РЕЖИМЕ. ОПРЕДЕЛЕНИЕ ГРАФИКИ ЭТИХ КЛАВИШ ПРОИЛЛЮСТРИРУЕМ НА ПРИМЕРЕ ОПРЕДЕЛЕНИЯ СИМВОЛА БУКВЫ ГРЕЧЕСКОГО АЛФАВИТА 'ПИ'.

1. КАЖДАЯ СИМВОЛ ПРЕДСТАВЛЯЕТСЯ ТОЧКАМИ В МАТРИЦЕ 8 X 8, 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 'P' ТАК, ЧТОБЫ ПРИ НАЖАТИИ КЛАВИШИ В ГРАФИЧЕСКОМ РЕЖИМЕ ВЫДАВАЛСЯ СИМВОЛ ПИ.



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

```

BIN 00000000 = 0
BIN 00000000 = 0
BIN 00000010 = 2
BIN 00111100 = 60
BIN 01010100 = 84
BIN 00010100 = 20
BIN 00010100 = 20
BIN 00000000 = 0

```

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 ЕДИНСТВЕННЫЙ 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
```

С ПОМОЩЬЮ ЭТОЙ ПРОГРАММЫ МОЖНО ВВЕСТИ ВОСЕМЬ ЧИСЕЛ, ОПРЕДЕЛЯЮЩИХ ГРАФИКУ СИМВОЛА, ЗАКРЕПЛЕННОГО ЗА КЛАВИШЕЙ 'P' (В РЕЖИМЕ (G)).

ОПЕРАТОР 'POKE' ЗАПИСЫВАЕТ ДАННЫЕ НЕПОСРЕДСТВЕННО В ПАМЯТЬ, ИМНУЯ 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 НАБОРА	СИМВОЛ	КОД НАБОРА
	128 (G) 8		143 (G) SH8
	129 (G) 1		142 (G) SH1
	130 (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 ТЕЛЕВИЗОРЕ ОТОБРАЖАЕТСЯ ЗНАК '?'. НАЗНАЧЕНИЕ ЭТИХ СИМВОЛОВ ОПИСАНО В ПРИЛОЖЕНИИ А.

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

CHR\$ 6 ПЕЧАТАЕТ ПРОБЕЛ, ИСПОЛЗУЕМЫЙ КАК ЗАПЯТАЯ В ОПЕРАТОРЕ PRINT.

```
PRINT 1;CHR$ 6;2
```

ДАСТ РЕЗУЛЬТАТ ЧТО И ОПЕРАТОР: PRINT 1,2

НО ЭТО НЕ СОВСЕМ КОРРЕКТНОЕ ИСПОЛЗОВАНИЕ, ВЕРНЕЕ БУДЕТ СДЕЛАТЬ:

```
10 LET A$="1"+CHR$ 6+"2"
```

```
20 PRINT A$
```

CHR\$ 8 - СИМВОЛ ЗАВОЯ, ОБЕСПЕЧИВАЕТ ВОЗВРАТ НА ОДНУ ПОЗИЦИЮ НАЗАД ОПЕРАТОР

```
PRINT "1234";CHR$ 8;"5"
```

ДАСТ СТРОКУ '1235'.

CHR\$ 13 - ПЕРЕВОД СТРОКИ, ПРОДОЛЖАЕТ ВЫВОД С НОВОЙ СТРОКИ. ГЛАВА 15
ТЕЛЕВИЗОР ТАКЖЕ ИСПОЛЬЗУЕТСЯ 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
РАСПОЛОЖЕНЫ ПОСЛЕ ЗАГЛАВНЫХ, ТАК ЧТО "A" СЛЕДУЕТ ПОСЛЕ "Z".

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 СЛУЧАЕ, СТРОКИ СЧИТАЮТСЯ РАВНЫМИ.

ОТНОШЕНИЯ =, <, >, (=) =, <(>) ПРИМЕНЯЮТСЯ К СТРОКАМ СИМВОЛОВ
ТАКЖЕ КАК И К ЧИСЛАМ: ЗНАК "<" ОЗНАЧАЕТ - НАХОДИТСЯ ВПЕРЕДИ
В КОДОВОЙ ТАБЛИЦЕ, А ">" - НАХОДИТСЯ ПОЗАДИ. ТАК ЧТО ВЫРАЖЕНИЯ

"AARDVARK" < "AARDVARK"

"AARDVARK" > "AARDVARK"

ОБА ИСТИНЫ.

ДЛЯ ИЛЛЮСТРАЦИИ ВСЕГО СКАЗАННОГО ПРИВЕДЕН ПРОГРАММУ,
КОТОРАЯ 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 " "
70 GO TO 10
```

СЛЕДУЮЩАЯ ПРОГРАММА ЗАКРЕПЛЯЕТ ОПРЕДЕЛЕННЫЕ ПОЛЬЗОВАТЕЛЕМ
СИМВОЛЫ ДЛЯ ИГРЫ В ШАХМАТЫ ЗА КЛАВИШАМИ:

P - ЗА ПЕШКОМ: (PAWN)
R - ЗА ЛАДЬЕЙ: (ROOK)
N - ЗА КОНЕМ: (KNIGHT)
B - ЗА СЛОНОМ: (BISHOP)
K - ЗА КОРОЛЕМ: (KING)
Q - ЗА КОРОЛЕВОЙ: (QUEEN)

```
5 LET B=BIN 01111100:LET C=BIN 00111000:
  LET D=BIN 00010000
10 FOR N=1 TO 6:READ P$:REM 6 PIECES
20 FOR F=0 TO 8:REM READ PIECE INTO 8 BYTES
30 READ A:POKE USR P$+F,A
40 NEXT F
50 NEXT N
100 REM BISHOP
110 DATA "B",0,D,BIN 00101000,BIN 01000100
120 DATA BIN 01101100,C,B,0
130 REM KING
140 DATA "K",0,D,C,D
150 DATA C,BIN 01000100,C,0
160 REM ROOK
170 DATA "R",0,BIN 01010100,B,C
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 PAWN
230 DATA "P",0,0,D,C
240 DATA C,D,B,0
250 REM KNIGHT
260 DATA "N",0,D,C,BIN 01111000
270 DATA BIN 00011000,C,B,0
```

ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ ОБ ОПЕРАТОРАХ PRINT И INPUT

КРАТКОЕ СОДЕРЖАНИЕ: CLS, PRINT-ПАРАМЕТРЫ: ИХ ОТСУТСТВИЕ
ВООБЩЕ, ВЫРАЖЕНИЕ (ЧИСЛОВОЕ ИЛИ
СТРОКОВОЕ): TAB ЧИСЛОВОЕ ВЫРАЖЕНИЕ,
PRINT РАЗДЕЛИТЕЛЬ: ; , ; , ;
INPUT-ПАРАМЕТРЫ,
LINE СТРОЧНАЯ ПЕРЕМЕННАЯ,
СВЕРТКА, SCREEN\$.

ВЫРАЖЕНИЯ, ЗНАЧЕНИЯ КОТОРЫХ ИСПОЛЬЗУЮТСЯ В ОПЕРАТОРЕ
PRINT, НАЗЫВАЮТСЯ PRINT-ПАРАМЕТРАМИ, ОНИ РАЗДЕЛЯЮТСЯ ЗАПЯТЫМИ
ИЛИ ТОЧКАМИ С ЗАПЯТОЙ, НАЗЫВАЕМЫМИ PRINT-РАЗДЕЛИТЕЛЯМИ.

В ОПЕРАТОРЕ PRINT ВОЗМОЖНО ОТСУТСТВИЕ НЕКОТОРЫХ ПАРАМЕТРОВ,
В ЭТОМ СЛУЧАЕ СТАВЯТСЯ ДВЕ ЗАПЯТЫЕ ПОДРЯД.

PRINT AT 11,16;"X"

ВЫВЕДЕТ "X" В СЕРЕДИНЕ ЭКРАНА.

AT "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 ЗАКЛЮЧАЮТСЯ В СКОБКИ.

ОПЕРАТОР SCREEN\$ СООБЩАЕТ ВАМ КАКОЕ СИМВОЛ НАХОДИТСЯ НА
ЭКРАНЕ В УКАЗАННОЙ ПОЗИЦИИ. ВЫПОЛНИВ:

PRINT SCREEN\$ (11,16)

МЫ ПОЛУЧИМ "X", ВЫВЕДЕННУЮ ПРЕЖДУЩИМ ОПЕРАТОРОМ.

В КАЧЕСТВЕ ВОЗВРАЩАЕМОГО ЗНАЧЕНИЯ МОГУТ ИСПОЛЬЗОВАТЬСЯ:

АЛФАВИТНО-ЦИФРОВЫЕ СИМВОЛЫ, СПЕЦИАЛЬНЫЕ СИМВОЛЫ, ПРОВЕЛИ,
ЛИНИИ, НАРИСОВАННЫЕ С ПОМОЩЬЮ ОПЕРАТОРОВ: PLOT, DRAW, CIRCLE,
ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ СИМВОЛЫ И ГРАФИЧЕСКИЕ СИМВОЛЫ,
ВОЗВРАЩАЮТСЯ КАК ПУСТАЯ СТРОКА. ТО ЖЕ, КОГДА ФУНКЦИЯ OVER
ИСПОЛЬЗУЕТСЯ ДЛЯ ПОЛУЧЕНИЯ КОМБИНИРОВАННЫХ ЗНАКОВ.

TAB СТОЛБЕЦ

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

ПОНИМАЙТЕ, ЧТО КОМПЬЮТЕР УМЕНЬШАЕТ ПОЗИЦИИ ПО МОДУЛЮ 32
(Т.Е. ДЕЛИТ НА 32 И ИСПОЛЬЗУЕТ ОСТАТОК). ТАК, "TAB 33"
РАВНОЗНАЧНО "TAB 1". К ПРИМЕРУ:

PRINT TAB 30;1:TAB 12;"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 B*N;N
30 NEXT N
```

БОЛЕЕ НАГЛЯДНЫЙ ПРИМЕР ПОЛУЧИТСЯ ПРИ ЗАМЕНЕ В 20 СТРОКЕ
"B" НА "6".

НЕСКОЛЬКО ЗАМЕЧАНИЙ:

1) В РАССМОТРЕННЫХ ПРИМЕРАХ В КАЧЕСТВЕ ОГРАНИЧИТЕЛЕЙ
ИСПОЛЬЗОВАЛИСЬ ";", МОЖНО ИСПОЛЬЗОВАТЬ " " (ИЛИ ВООБЩЕ НИЧЕГО).
ПРИ ЭТОМ НЕОБХОДИМО СЛЕДИТЬ ЗА УСТАНОВКОЙ ТЕКУЩЕЙ ПОЗИЦИИ
ВЫВОДА.

2) НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ ДЛЯ ВЫВОДА ДВЕ НИЖНИЕ СТРОКИ
ЭКРАНА (22,23) 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 СОМЕСТНО С PRINT
ЯВЛЯЕТСЯ CLS, ОН ПРОИЗВОДИТ ОЧИСТКУ ЭКРАНА ДО ПОСЛЕДНЕГО ОПЕРАТОРА

CLEAR и RUN.

ПРИ ЗАПОЛНЕНИИ ЭКРАНА ПРОИСХОДИТ ЕГО СВЕРТКА, В ЭТОМ МОЖНО УБЕДИТЬСЯ, ПРОДЕЛАВ:

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

И ДАЛЕЕ ВЫПОЛНИТЬ 'PRINT 99' НЕКОТОРЫЕ 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?". ПОСЛЕ ПРОСМОТРА НАЖМИТЕ 'Y' (ДА) И ВЫВОД ПРОДОЛЖИТСЯ. ВОЗМОЖЕН ОТРИЦАТЕЛЬНЫЙ ОТВЕТ 'N' (НЕТ), STOP (SYMBOL SHIFT и 'A') ИЛИ SPACE (BREAK). В ПОСЛЕДНЕМ СЛУЧАЕ КОМПЬЮТЕР ОСТАНОВИТ ПРОГРАММУ И ВЫДАСТ СООБЩЕНИЕ:

```
'D BREAK-CONT REPEATS'.
```

ОПЕРАТОР INPUT ИСПОЛЬЗУЕТСЯ ДЛЯ ВЫВОДА РАЗЛИЧНЫХ ЗНАЧЕНИЙ. НАПРИМЕР:

```
INPUT "HOW 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, "HOW 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 ЖЕРЕ ВЫДАЧИ ОПЕРАТОРОВ INPUT ПРОИСХОДИТ СВЕРТКА ЭКРАНА. РАССМОТРИМ ПРИМЕР ИСПОЛЬЗОВАНИЯ AT В ОПЕРАТОРЕ INPUT:

```
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 САДВИГАТЬСЯ ВВЕРХ, ОСВОБОЖДАЯ МЕСТО, НО НУМЕРАЦИЯ ОСТАНЕТСЯ ПРЕЖНЕЙ. 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, ПРОИЗОИДЕТ СВЕРТКА ЭКРАНА. ЕЩЕ ОДНИМ ПАРАМЕТРОМ ОПЕРАТОРА INPUT ЯВЛЯЕТСЯ LINE. ОН 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. ЗНАЧЕНИЕ ЗАДАВАЕМЫХ ИМ ПАРАМЕТРОВ НАХОДЯТСЯ В ПРЕДЕЛАХ ОТ 0 ДО 65535

```
PRINT CHR$23+CHR$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:

```
5 РОКЕ 23692,255
```

```
10 FOR N=0 TO 10000
```

```
20 PRINT N
```

```
30 NEXT N
```

И СЛЕДИТЕ СКОЛЬКО СВЕРТОК СДЕЛАЕТ КОМПЬЮТЕР.

ЗАПУСТИМ СЛЕДУЮЩУЮ ПРОГРАММУ, ПРОВЕРЯЮЩУЮ ЗНАНИЕ ТАБЛИЦЫ УМНОЖЕНИЯ:

```
10 LET M$=""
```

```
20 LET A=INT(RND*12)+1:LET B=INT(RND*12)+1
```

```
30 INPUT(M$), "WHAT IS ";(A);"*(B);"? "C
```

```
100 IF C=A*B THEN LET M$="RIGHT": GOTO 20
```

```
111 LET M$="WRONG TRY AGAIN.": GOTO 30
```

МОЖНО НЕКОЛЬКО ИЗМЕНИТЬ ПРОГРАММУ, ТАК ЧТОБЫ НЕ ЗНАЯ ПРАВИЛЬНОГО 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 M$="TYPE IT PROPERLY,  
AS NUMBER.": PRINT C$;"=";"VAL C$": GOTO 30
```

ГЛАВА 16

ЦВЕТА

=====

КРАТКОЕ СОДЕРЖАНИЕ: INK, PAPER, FLASH, BRIGHT, INVERSE, OVER, BORDER, ATTR.

ВЫПОЛНИМ СЛЕДУЮЩУЮ ПРОГРАММУ:

```
10 FOR M=0 TO 1:BRIGHT M
```

```
20 FOR N=1 TO 12
```

```
30 FOR C=0 TO 7
```

```
40 PAPER C:PRINT " ";REM 4 COLOURED SPACES
```

```
50 NEXT C:NEXT N: NEXT M
```

```
60 FOR M=0 TO 1:BRIGHT M:PAPER 7
```

```
70 FOR C=0 TO 3
```

```
80 INK C:PRINT C;" "
```

```
90 NEXT C:PAPER 6
```

```
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 ЦВЕТНОГО СПЕКТРУМ НА ЦВЕТНОМ ТЕЛЕВИЗОРЕ 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 - ЗЕЛЕННЫЙ

5 - ГОЛУБОЙ

6 - ЖЕЛТЫЙ

7 - БЕЛЫЙ

ДЛЯ ЧЕРНО-БЕЛОГО ТЕЛЕВИЗОРА ЭТОТ РЯД ПРЕСТАВЛЯЕТ 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 ЗНАКМЕСТА), КАЖДАЯ ИЗ КОТОРЫХ ПРЕСТАВЛЯЕТ ИЗ СЕБЯ МАТРИЦУ 8 НА 8 ПИКСЕЛЕЙ.

ВСПОМНИМ: 0 - БЕЛАЯ ТОЧКА, 1 - ЧЕРНАЯ ТОЧКА.

[K], [L], [C]	[G]	[E]
HET : SYMBOL : CAPS :	HET : ЛЮБОЯ :	HET : CAPS : SYMBOL :
1 : ! : EDIT :	 	ФОН : ТОН : DEF FN : : ГОЛУБОЯ : ГОЛУБОЯ :
2 : @ : CAPS : : LOCK :	 	ФОН : ТОН : FN : : КРАСНАЯ : КРАСНАЯ :
3 : * : TRUE : : VIDEO :	 	ФОН : ТОН : : ФИОЛЕТОВАЯ : ФИОЛЕТОВАЯ : LINE : : ВЯЯ : ВЯЯ :
4 : \$: INVERS : : VIDEO :	 	ФОН : ТОН : OPEN : : ЗЕЛЕНАЯ : ЗЕЛЕНАЯ :
5 : X : KUPCOP : : ВЛЕВО :	 	ФОН : ТОН : CLOSE : : СИНИА : СИНИА :
6 : & : KUPCOP : : ВНИЗ :	 	ФОН : ТОН : MOVE : : ЖЕЛТАЯ : ЖЕЛТАЯ :
7 : ' : KUPCOP : : ВВЕРХ :	 	ФОН : ТОН : ERASE : : БЕЛАЯ : БЕЛАЯ :

```

:      :      : КУРСОР :      :      :      :
: 8 : ( : ВПРАВО :  :  : НОРМ. : БЕЗ : POINT :
:      :      :      :      :      :      :
:      :      :      :      :      :      :
:      :      : ГРАФИЧ. : ГРАФИЧ. : ПОВЫШ. : С :
: 9 : ) : ЧЕСКИЙ : ВЫХОД : ВЫХОД : ЯРКОСТЬ : МЕРЦАН. : SAT :
:      :      : РЕЖИМ :      :      :      :      :
:      :      :      :      :      :      :
: 0 : _ : DELETE : DELETE : DELETE : ФОН : ЗАКРАШ. : FORMAT :
:      :      :      :      :      :      :
:      :      :      :      :      :      :
:      :      : ЧЕРНЫЙ : ЧЕРНЫЙ :

```

INK и другие операторы не действуют в нижней части экрана, предназначенной для ввода команд и INPUT-данных. Для изменения цветов в этой части экрана служит оператор:

BORDER 'цвет'

Кодировка цветов прежняя. Возможны мерцание и повышенная яркость, для этого используйте соответствующие параметры в INPUT (напоминание 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?"IN

Возможно изменение цветов и с помощью управляющих символов, поданных 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 (EJ) цифрами. Цифры от 0 до 7 устанавливают 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 введенное, 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 образом до тех пор пока курсор не выйдет к предыдущему управляющему символу. Действия в расширенном режиме (EJ):

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 CLJ-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 введены значения атрибутов 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) В*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)

УПРАВЛЕНИЕ:

1) PRINT "B";CHR\$8;OVER 1:"/";

Здесь '/' ПЕРЕКЕРКНЕТ 'B'. Этим способом можно выводить слова из комбинационных знаков на ZX SPECTRUM: два фона или два тона дадут фон, один из них даст тон. Это интересное свойство, если вы повторите вывод одного символа дважды, то он не будет отображен. Так если дополнительно к описанному выше утверждению дать:

PRINT CHR\$8;OVER 1:"/";

ТО МЫ В РЕЗУЛЬТАТЕ УВИДИМ 'B' НЕ ПЕРЕКЕРКНУТОЕ '/'. ТАК ЛИ ЭТО?

2) ВЫПОЛНИМ:

PAPER 0:INK 0

ДЕЙСТВУЮТ ЛИ ЭТИ ОПЕРАТОРЫ В НИЖНЕЙ ЧАСТИ ЭКРАНА?

ЧТО МЫ УВИДИМ ЕСЛИ ДОБАВИТЬ

BORDER 0?

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 СТУПЕНЬКАХ, ЭТО ЯВЛЯЕТСЯ СЛЕДСТВИЕМ ТОГО, ЧТО МЫ ПОЛЬЗУЕМСЯ КВАЗИСЛУЧАЙНЫМ РАСПРЕДЕЛЕНИЕМ, КОТОРОЕ ЛИШЬ ПРИБЛИЖЕННО ВОСПРОИЗВОДИТ СЛУЧАЙНОЕ РАСПРЕДЕЛЕНИЕ.

4) ВВЕДЕН ВРУЧНУЮ, ИЛИ С ПОМОЩЬЮ ОПЕРАТОРА LOAD ПРОГРАММУ ДЛЯ ВВОДА ШАХМАТНЫХ ФИГУР ИЗ ГЛАВЫ 14 И ВВЕДЕН 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

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":REM BIG LINE 210

300 REM DISPLAY BOARD

310 FOR N=1 TO 8:FOR M=1 TO 8

320 LET BC=CODE B\$(N,M):INK PW

325 IF BC=CODE " " THEN GO TO 350:REM SPACE

330 IF BC>CODE "Z" THEN INK PB:LET BC=BC-32:REM LOWER CASE FOR BLACK

340 LET BC=BC+79:REM CONVERT TO GRAPHICS

350 PRINT AT 5+N,9+M:PAPERB:CHR\$BC

360 NEXT M:NEXT N

400 PAPER 7:INK 0

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 $22 \times 32 = 704$ СИМВОЛЬНЫЕ ПОЗИЦИИ. КАК ВЫ УЖЕ ПОНЯЛИ ИЗ 16 ГЛАВЫ, КАЖДАЯ СИМВОЛЬНАЯ ПОЗИЦИЯ ПРЕСТАВЛЯЕТСЯ КВАДРАТОМ 8×8 ТОЧЕК, НАЗЫВАЕМЫХ ПИКСЕЛЯМИ.

ПИКСЕЛЬ ЗАДАЕТСЯ ДВУМЯ ЧИСЛАМИ, ЕГО КООРДИНАТАМИ. ПЕРВОЕ ЗАДАЕТ КООРДИНАТУ X, ТО ЕСТЬ 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) ЗАДАЮТ СООТВЕТСТВЕННО НИЖНИЙ ЛЕВЫЙ, НИЖНИЙ ПРАВЫЙ, ВЕРХНИЙ ЛЕВЫЙ И ВЕРХНИЙ ПРАВЫЙ УГОЛ ЭКРАНА.

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

НАПРИМЕР ПРОГРАММА:

```
10 PLOT INT(RND*256),INT(RND*176)
20 INPUT A$
30 GO TO 10
```

БУДЕТ ВЫСВЕЧИВАТЬСЯ 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,80+80*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 SQR(X) (ЧАСТЬ ПАРАБОЛЫ) В ИНТЕРВАЛЕ ОТ 0 ДО 4.

ПОМНИТЕ, ЧТО КООРДИНАТЫ ПИКСЕЛЕЙ ОТЛИЧАЮТСЯ ОТ АДРЕСАЦИИ СТРОК И ПОЗИЦИИ В ПОДКОМАНДЕ AT.

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 ПРЕДУДУЩИХ ОПЕРАТОРОВ PLOT, DRAW ИЛИ CIRCLE (ЭТОТ ПИКСЕЛЬ НАЗЫВАЕТСЯ ТЕКУЩЕЙ PLOT-ПОЗИЦИЕЙ. ОПЕРАТОРЫ RUN, CLEAR, CLS И NEW УСТАНОВЛИВАЮТ ЕЕ В ЛЕВЫЙ НИЖНИЙ УГОЛ ЭКРАНА). ТАКИМ ОБРАЗОМ, ОПЕРАТОР DRAW ЗАДАЕТ ДЛИНУ И НАПРАВЛЕНИЕ ВЫЧЕРЧИВАНИЯ ЛИНИИ, НО НЕ 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 ДЕМОНСТРИРУЕТ ЭТО:

```
10 BORDER 0:PAPER 0:INK 7:CLS:REM BLACK OUT SCREEN
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 LAST ONE FINISHED
```

```
70 LET C=C+1:IF C=8 THEN LET C=1:REM NEW COLOUR
```

```
80 GO TO 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 56).

ПРИ ПОМОЩИ DRAW МОЖНО ТАКЖЕ ВЫЧЕРТИТЬ ОТРЕЗОК ДУГИ, ИСПОЛЬЗУЯ ДЛЯ ЭТОГО ДОПОЛНИТЕЛЬНОЕ ЧИСЛО, ЗАДАЮЩЕЕ УГОЛ (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 НА СЕВЕРО-ЗАПАД.

ОПЕРАТОР CIRCLE ВЫЧЕРЧИВАЕТ ПОЛНЫЙ КРУГ ЗАДАВАННЫМИ 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 ВЫДАЕТ '1' ЕСЛИ ПИКСЕЛЬ ИМЕЕТ ЦВЕТ ТОНА И 0 ЕСЛИ - ФОНА. НАПРИМЕР, СТРОКА ПРОГРАММЫ:

```
CLS:PRINT POINT(0,0):PLOT(0,0):PRINT POINT(0,0)
```

ВЫВЕДЕТ: 0

1

ДОПУСКАЕТСЯ ТАКЖЕ ЗАДАВАТЬ УПРАВЛЯЮЩИЕ СИМВОЛЫ INVERSE И OVER В ОПЕРАТОРЕ PLOT. ПО УМОЛЧАНИЮ ОНИ ПРЕДПОЛАГАЮТСЯ РАВНЫМИ 0 (ОТКЛЮЧЕНО), НО ВЫ МОЖЕТЕ ЗАДАТЬ И 1, ПРИ ЭТОМ:

PLOT INVERSE 1 - УСТАНОВЛИВАЕТ ДЛЯ ЗАДАННОГО ПИКСЕЛЯ ЦВЕТ ФОНА,

PLOT OVER 1 - ИЗМЕНЯЕТ ЦВЕТ ПИКСЕЛЯ НА ПРОТИВООПЛОЖНЫЙ, ЕСЛИ БЫЛ ЦВЕТ ТОНА, ТО СТАНОВИТСЯ ЦВЕТ ФОНА И НАОБОРОТ.

PLOT INVERSE 1:OVER 1 - СОХРАНЯЕТ ЦВЕТ ПИКСЕЛЯ БЕЗ ИЗМЕНЕНИЯ, 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 1:255,175
```

И ПЕРЕЧЕРТИТЕ ЕЕ КОМАНДОЙ

```
DRAW OVER 1:-255,-175
```

ЭТО НЕ ИЗМЕНИТ КАРТИНКУ, ТАК КАК ПРИ ЧЕРЧЕНИИ КАК ВПЕРЕД, ТАК И НАЗАД ИСПОЛЬЗУЮТСЯ ОДИН 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 USR"A"+N+1,BIN 10101010
1020 NEXT N
```

ОНА ЗАДАЕТ ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ СИМВОЛ ДЛЯ ШАХМАТНОЙ ДОСКИ, КОТОРЫЙ ЗАКРЕПЛЯЕТСЯ ЗА КЛАВИШУ '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', ЗАДАВШИМ ОБЛАСТЬ ЗНАЧЕНИЙ АРГУМЕНТА (Т.Е. ГРАФИК БУДЕТ СТРОИТЬСЯ ДЛЯ ЗНАЧЕНИЙ АРГУМЕНТА В ДИАПАЗОНЕ ОТ -N ДО +N).

ВТОРОЙ ОТВЕТ - ЭТО ВЫРАЖЕНИЕ В ВИДЕ СИМВОЛЬНОЙ СТРОКИ, ЗАДАЮЩЕЙ ФУНКЦИЮ, ИСПОЛЗУЮУЮ 'X' В КАЧЕСТВЕ АРГУМЕНТА.

```
10 PLOT 0,87:DRAW 255,0
20 PLOT 127,0:DRAW 0,175
30 INPUT N,E$
35 LET T=0
40 FOR F=0 TO 254
50 LET X=(F-128)*N/128:LET Y=VAL E$
60 IF ABS Y > 87 THEN LET T=0:GO TO 100
70 IF NOT T THEN PLOT F,Y+88:LET T=1:GO TO 100
80 DRAW 1,Y-OLDY
100 LET OLDY=INT(Y+0.5)
110 NEXT F
```

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

Г Л А В А 18

У К А З А Н И Я

=====

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 ТЕЧЕНИЕ 'N' ТЕЛЕВИЗИОННЫХ КАДРОВ (50 КАДРОВ/СЕК. В ЕВРОПЕ ИЛИ 60 В АМЕРИКЕ). 'N' МОЖЕТ БЫТЬ ВПЛОТЬ ДО 65535, ЧТО СОСТАВЛЯЕТ 22 МИНУТЫ, ЕСЛИ N=0, ТО ЭТО ОЗНАЧАЕТ, ЧТО ОПЕРАТОР PAUSE НЕ ИМЕЕТ ОГРАНИЧЕНИЙ 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 T IS 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 ПРИМЕРНО 55,5 ЧАСОВ, ЧТО ЗАДАЕТСЯ В ОПЕРАТОРЕ С НОМЕРОМ 60. ОПЕРАТОР 210 ПРОИЗВОДИТ ОТСЧЕТ ВРЕМЕНИ. КАЗАЛОСЬ БЫ ЗАСЬ ДОЛЖЕН БЫТЬ ОПЕРАТОР PAUSE 50 (ЕВРОПА), ДЛЯ ТОЧНОГО ОТСЧЕТА ОДНОЙ СЕКУНДЫ, НО ТОГДА БЫ МЫ НЕ УЧЛИ ВРЕМЯ, ЗАТРАЧИВАЕМОЕ НА ВЫПОЛНЕНИЕ ОСТАЛЬНЫХ ОПЕРАТОРОВ ПРОГРАММЫ. РАССМАТРИВАЕМЫЙ ВАРИАНТ ЧАСОВ ОБЕСПЕЧИВАЕТ 2X-0 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. ЗАСЬ ЖЕ В КАЧЕСТВЕ ПРИМЕРА РАССМОТРИМ ВЫРАЖЕНИЕ:

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

ОНО ДАЕТ КОЛИЧЕСТВО СЕКУНД, ПРОШЕДШИХ С ТЕХ ПОД КАК КОМПЬЮТЕР БЫЛ ВКЛЮЧЕН ИЛИ БЫЛ ААН 'СВРОС' (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 N=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 SECOND SINCE
   START
100 REM NOW WE START THE CLOCK
110 LET T1=FN T()
120 LET A=T1/30*PI:REM A 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 GO TO 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:GO TO 120
```

ЭТИ ЧАСЫ ОБЕСПЕЧИВАЮТ ТОЧНОСТЬ 0,01X ИЛИ УХОД НА 10 СЕКУНД В ДЕНЬ. ОДНАКО, ЭТО ВОЗМОЖНО ПРИ УСЛОВИИ, ЧТО ВЫ НЕ ИСПОЛЬЗОВАЛИ ОПЕРАТОР ВЕР, ВВОД/ВЫВОД НА МАГНИТОФОН И ПРИНТЕР ВСЕ ЭТИ ОПЕРАЦИИ УВЕЛИЧИВАЮТ ПОГРЕШНОСТЬ.

ЧИСЛА PEEK 23674, PEEK 23673 И PEEK 23672 ВЫДЕЛЯЮТ АДРЕСА ЯЧЕЕК ПАМЯТИ КОМПЬЮТЕРА И ИСПОЛЗУЕМЫХ ДЛЯ ПОДСЧЕТА 1/50 ДОЛЕЙ СЕКУНД. 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 PEEK 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 PEEK 23673 УВЕЛИЧИТСЯ НА 1. ЧЕРЕЗ КАКИЕ 256/50 СЕК. СОДЕРЖИМОЕ ЭТОЙ ЯЧЕЙКИ ПЕРЕХОДИТ ИЗ СОСТОЯНИЯ 255 В 0, А СОДЕРЖИМОЕ ЯЧЕЙКИ PEEK 23674 УВЕЛИЧИВАЕТСЯ НА 1.

ПРИ ЗНАЧЕНИЯХ 0 ДЛЯ ЯЧЕЙКИ PEEK 23674 И 255 ДЛЯ ЯЧЕЕК PEEK 23673 И PEEK 23672 (ЭТОТ МОМЕНТ НАСТУПИТ ЧЕРЕЗ 21 МИНУТУ) НАШЕ ВЫРАЖЕНИЕ ПРИМЕТ ЗНАЧЕНИЕ:

$(65536*0 + 256*255 + 255)/50=1310.7$

НО ЗАСЬ ИМЕЕТСЯ СКРЫТАЯ ОПАСНОСТЬ. ЧЕРЕЗ СЛЕДУЮЩУЮ 1/50 СЕК. ЯЧЕЙКИ БУДУТ СОДЕРЖАТЬ СООТВЕТСТВЕННО СЛЕДУЮЩИЕ ЗНАЧЕНИЯ: 1, 0, 0.

ПОКА ПРОИЗВОДИТСЯ ВЫЧИСЛЕНИЕ ВЫРАЖЕНИЯ, КОМПЬЮТЕР МОЖЕТ ОЦЕНИТЬ ЗНАЧЕНИЕ ЯЧЕЙКИ PEEK 23674 КАК 0 ДО ЗАВЕРШЕНИЯ ЦИКЛИЧЕСКОГО ПЕРЕНОСА. В РЕЗУЛЬТАТЕ ПОЛУЧИМ:

$(65536*0 + 256*0 + 0)/50=0,$

ЧТО БЕЗНАЧЕЖНО НЕВЕРНО.

ПРОСТОЕ ПРАВИЛО ПОЗВОЛЯЕТ РЕШИТЬ ЭТУ ПРОБЛЕМУ: "СЛЕДУЕТ ВЫЧИСЛЯТЬ ВЫРАЖЕНИЕ АВАЖАМ В НЕКОТОРОЙ ПОСЛЕДОВАТЕЛЬНОСТИ И ИСПОЛЬЗОВАТЬ СОХРАНЕННЫЙ ОТВЕТ".

ПРИМЕР:

```
10 DEF FN M(X,Y)=(X+Y+ABS(X-Y))/2:REM THE LARGER OF X
   AND Y
20 DEF FN U()=(65536*PEEK 23674 + 256*PEEK 23673 +
   PEEK 23672)/50:REM TIME, MAY BY 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 ПОДСЧИТАЛИ, ЧТО ЭТО

$10*60*60*50 = 1800000$ 50-Х ДОЛЕЙ СЕКУНДЫ И ЗНАЧИТ

$$1800000 = 65536 * 27 + 256 * 119 + 64.$$

ДЛЯ ПРИСВОЕНИЯ ТРЕМ ЯЧЕЙКАМ ЗНАЧЕНИЯ 27, 119 И 64
НЕОБХОДИМО ВЫПОЛНИТЬ

POKE 23674,27:POKE 23673,119:POKE 23672,64

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

ВЫПОЛНИТЕ ПРОГРАММУ, КОТОРАЯ ИСПОЛЬЗУЕТ ЭТУ ФУНКЦИЮ:

```

10 IF INKEY$ <> " " THEN GO TO 30
20 IF INKEY$ = " " THEN GO TO 20
30 PRINT INKEY$;
40 GO TO 10

```

ПОМНИТЕ, ЧТО ФУНКЦИЯ INKEY\$ НЕ БУДЕТ ПОДОБНО INPUT ЖДАТЬ ВАС.

ГЛАВА 19

ПРОГРАММИРОВАНИЕ ЗВУКОВ

КРАТКОЕ СОДЕРЖАНИЕ: ВЕР.

ZX SPECTRUM МОЖЕТ ВОСПРОИЗВОДИТЬ ЗВУКИ ПРИ ПОМОЩИ
ОПЕРАТОРА BEEP:

ВЕР 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:

-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

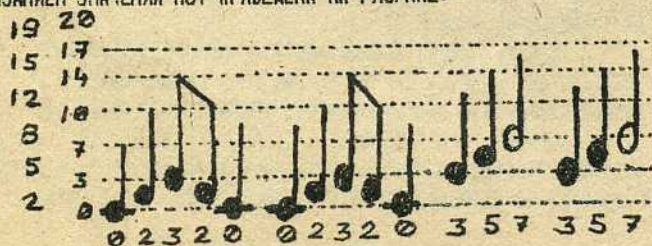
ДЛЯ ПОЛУЧЕНИЯ БОЛЕЕ ВЫСОКИХ ИЛИ БОЛЕЕ НИЗКИХ НОТ, ВЫ ДОЛЖНЫ ПРИБАВИТЬ ИЛИ ОТНЯТЬ 12 ДЛЯ КАЖДОЙ ОКТАВЫ ВВЕРХ ИЛИ ВНИЗ. НАПРИМЕР:

```

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

```

КОГДА ВЫ ЗАПУСТИТЕ ЭТУ ПРОГРАММУ, ВЫ УСЛЫШИТЕ ПОХОРОННЫЙ МАРШ ИЗ ПЕРВОЙ СИМФОНИИ МОЛЬБЕРА, ТУ ЧАСТЬ, КОГДА ГОБЛЕИНЫ ХОРОЧАТ РЫДАРИЯ. 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 BEEP 1,KEY+0:BEEP 1,KEY+2:BEEP .5,KEY+3:

```
BEEP .5,KEY+2:BEEP 1,KEY+0
```

ТЕПЕРЬ, ПРИ ВЫПОЛНЕНИИ 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 НОТ. НО ПОМНИТЕ, ЧТО КОМПЬЮТЕР МОЖЕТ ОДНОВРЕМЕННО ИСПОЛНЯТЬ ОДНУ НОТУ, ЧТО НЕ ПОЗВОЛЯЕТ ВОСПРОИЗВОДИТЬ СЛОЖНЫЕ МЕЛОДИИ.

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

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

БУДЕТ ИСПОЛНЯТЬ ПОСЛЕДОВАТЕЛЬНО ПОТМ ДО ПРЕДЕЛЬНО ВЫСОКОЙ
И ЗАВЕРШИТСЯ С СООБЩЕНИЕМ ОБ ОШИБКЕ 'B'. ВЫ МОЖЕТЕ ТАКЖЕ
ПАРАЛЛЕЛЬНО ВЫВОДИТЬ ЗНАЧЕНИЯ 'N', ЧТОБЫ ЗНАТЬ ЗНАЧЕНИЕ
ИСПОЛНЯЕМОЙ ПОТМ. ФРАГМЕНТ ПРОГРАММЫ:

10 BEEP .5,0:BEEP .5,2:BEEP .5,4:BEEP .5,5:BEEP .5,7:

BEEP .5,9:BEEP .5,11:BEEP .5,12:STOP

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

20 BEEP .5,0=BEEP .5,2.039=BEEP .5,3.86=BEEP .5,4.98=

BEEP .5,7.02:BEEP .5,8.84:BEEP .5,10.88:

BEEP .5,12:STOP

ЭТИ ЖЕ ИНТЕРВАЛЫ БУДУТ ЕСТЕСТВЕННЫМИ ДЛЯ ГАМИН 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 ВЕР. НАПРИМЕР, ДЛЯ ЗВУКА НА ЧЕТВЕРТЬ ТОНА ВЫШЕ СРЕДНЕГО ПО НАДО УКАЗАТЬ ЗНАЧЕНИЕ ВЫСОТЫ ЗВУКА РАВНОЕ .5.

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

POKE 23609,255

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

МОЖНО ТАКЖЕ ВЫВЕСТИ МУЗЫКУ НА ВНЕШНИЕ УСТРОЙСТВА.

ГЛАВА 20

ВНЕШНЯЯ ПАМЯТЬ ИЛИ КАГНИТНОЕ ЧЕЛТЕ

КРАТКОЕ СОДЕРЖАНИЕ: LOAD, SAVE, VERIFY, MERGE

ОСНОВНЫЕ КОМАНДЫ РАБОТЫ С МАГНИТОФОННОМ 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 СТАРОЙ ПРОГРАММЫ ИЛИ ПЕРЕМЕННЫЕ, КОТОРЫЕ СОВПАДАЮТ С НОМЕРАМИ СТРОК НОВОЙ ПРОГРАММЫ ИЛИ ИМЕНАМИ НОВЫХ ПЕРЕМЕННЫХ. ПРОГРАММУ "DISC" ("ИГРАЛЬНАЯ КОСТЬ") ИЗ ГЛАВЫ 11 ЗАТИШЕЛ НА ЛЕНТУ ПОД ИМЕНЕМ "DISC". А ТЕПЕРЬ ВВЕДЕМ И ВЫПОЛНИМ СЛЕДУЮЩУЮ ПРОГРАММУ:

```
1 PRINT 1
```

2 PRINT 2

16 PRINT 10

```
20 LET X=20
```

А ЗАТЕМ ОСУЩЕСТВИМ ЕЕ ПРОВЕРКУ, ЗАКЛИНИВ КОМАНДУ VERIFY "DICE"
НА КОМАНДУ MERGE "DICE". ВМ УВИДИТЕ, ЧТО СТРОКИ 1 И 2 СОХРАНЯ-

ОТСЯ, А СТРОКИ 10 И 20 ЗАМЕНЯЮТСЯ НА СТРОКИ С ЭТИМИ НОМЕРАМИ ИЗ 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 СОВПАДАТ НОМЕРА ИЛИ ИМЕНА С ТАКИМИ ЖЕ НА МАГНИТНОЙ ЛЕНТЕ.

ЗА КАЖДОЙ ИЗ ЭТИХ КОМАНД СЛЕДУЕТ КЛЮЧЕВОЕ СЛОВО - ИМЯ ПРОГРАММЫ, ОПРЕДЕЛЕННОЕ ПЕРВОНАЧАЛЬНО В КОМАНДЕ SAVE. ПОКА КОМПЬЮТЕР ИМЕЕТ УКАЗАННУЮ ПРОГРАММУ, ОН ВЫВОДИТ ИМЕНА ВСЕХ ПРОГРАММ, УЖЕ ПРОЧИТАННЫХ С ЛЕНТЫ. ИМЕЮТСЯ 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 ВСТРЕТИВШИЙСЯ ФАЙЛ.

ВАРИАНТ 2. С ИСПОЛЬЗОВАНИЕМ ОПЕРАТОРА SAVE:

SAVE "NAME" 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 УКАЗАНИЕМ НОМЕРА И САМА ИНИЦИИРУЕТ ВЫПОЛНЕНИЕ.

КРОМЕ ТЕКСТОВ ПРОГРАММ НА ЛЕНТУ МОЖНО ЗАПИСАТЬ ТАКЖЕ МАССИВЫ ИЛИ ДАННЫЕ.

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

SAVE "NAME" DATA LETTER()

ЗДЕСЬ "NAME" - ИМЯ, ПРИСВАИВАЕМОЕ ФАЙЛУ ДАННЫХ, КОТОРОЕ МОЖЕТ СОСТОЯТЬ ИЗ БУКВ ИЛИ БУКВ И СИМВОЛОВ. ДЛЯ СТРОКОВЫХ ДАННЫХ ЭТО ТРЕБОВАНИЕ ЗДЕСЬ НЕ ВАЖНО. ЗАГРУЖАЮТСЯ ТАКИЕ ДАННЫЕ ПО КОМАНДЕ LOAD

НЕЛЬЗЯ ИСПОЛЬЗОВАТЬ ОПЕРАТОР MERGE.

ЕСЛИ ЗАГРУЖАЕТСЯ СТРОКОВЫЙ МАССИВ, ТО ПОСЛЕ ОБНАРУЖЕНИЯ ЕГО НА ЛЕНТЕ. КОМПЬЮТЕР ВЫДАЕТ "CHARACTER ARRAY:" И ДАЛЕЕ ИМЯ ЭТОГО МАССИВА.

СУЩЕСТВУЕТ ВОЗМОЖНОСТЬ ЗАПИСИ НА МАГНИТНУЮ ЛЕНТУ И ОТДЕЛЬНЫХ БАЙТОВ ИНФОРМАЦИИ. ТАК, НАПРИМЕР, ЭТО МОЖЕТ БЫТЬ ТЕЛЕВИЗИОННАЯ КАРТИНКА ИЛИ ОПРЕДЕЛЯЕМЫЕ ПОЛЬЗОВАТЕЛЕМ ГРАФИЧЕСКИЕ СИМВОЛЫ И Т.Д. ДЛЯ ЭТОГО ИСПОЛЬЗУЕТСЯ КЛЮЧЕВОЕ СЛОВО CODE, НАПРИМЕР:

SAVE "PICTURE" CODE 16384,6912

ЗДЕСЬ ПЕРВОЕ СЛОВО - АДРЕС ПЕРВОГО БАЙТА В ОБЛАСТИ ПАМЯТИ, ГДЕ РАСПОЛОЖЕНЫ ДАННЫЕ, А ВТОРОЕ СЛОВО - КОЛИЧЕСТВО БАЙТОВ, КОТОРОЕ НУЖНО ЗАПИСАТЬ НА ЛЕНТУ (6912 - ОБЪЕМ В БАЙТАХ ОДНОГО ЭКРАНА, А 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 (В БАЙТАХ) НАДО ЗАГРУЗИТЬ С ЛЕНТЫ. ЕСЛИ ДЛИНА БОЛЬШЕ, ЧЕМ ЗАПИСАНО НА ЛЕНТУ, ТО ВЫДАЕТСЯ СООБЩЕНИЕ "R TAPE 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 ОПУСКАТЬ 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, ИСПОЛЬЗУЮТСЯ ТОЛЬКО ПЕРВЫЕ 10 СИМВОЛОВ. СУЩЕСТВУЕТ ЧЕТЫРЕ ТИПА ИНФОРМАЦИИ, КОТОРЫЕ МОГУТ БЫТЬ ЗАПИСАНЫ НА ЛЕНТУ:

- ПРОГРАММЫ И ПЕРЕМЕННЫЕ (СОВМЕСТНО);
- ЧИСЛОВЫЕ МАССИВЫ;
- СТРОКОВЫЕ МАССИВЫ;
- НЕПОСРЕДСТВЕННЫЕ БАЙТЫ.

КОГДА КОМАНДА VERIFY, LOAD И MERGE ОСУЩЕСТВЛЯЮТ ПОИСК ДАННЫХ НА ЛЕНТЕ, ОНИ ВЫВОДЯТ НА ЭКРАН ВСЕ СЧИТАННЫЕ ИМЯ С ЛЕНТЫ: ИМЕНА С УКАЗАНИЕМ ТИПА ДАННЫХ В ВИДЕ

"PROGRAM", "NUMBER ARRAY:", "CHARACTER ARRAY:" ИЛИ "BYTES:".

ЕСЛИ ИМЯ - ПУСТАЯ СТРОКА, ЭТИ КОМАНДЫ БЕРУТ ПЕРВЫЙ ВСТРЕТИВШИЙСЯ ФАЙЛ С УКАЗАНИЕМ 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 ХВАТ НАЖАТИЯ, ПОСЛЕ ЧЕГО 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, LENGTH

ЗАПИСЫВАЕТ НА ЛЕНТУ "LENGTH" 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\$()

ЗАПИСЫВАЕТ ЧИСЛОВОЙ ИЛИ СТРОКОВЫЙ МАССИВ (ТРЕБОВАНИЕ \$ НЕ ОТНОСИТСЯ К "NAME").

КОМАНДА VERIFY ПРОВЕРЯЕТ (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 "LENGTH", ТО ВЫДАЕТСЯ СООБЩЕНИЕ ОБ ОШИБКЕ "R".

VERIFY "NAME" CODE START

ЗДЕСЬ ОСУЩЕСТВЛЯЕТСЯ СРАВНЕНИЕ БАЙТОВ В ФАЙЛЕ "NAME" С ДАННЫМИ В ПАМЯТИ, НАЧИНАЯ С АДРЕСА "START".

VERIFY "NAME" CODE

ЭТОТ ОПЕРАТОР ОСУЩЕСТВЛЯЕТ СРАВНЕНИЕ ДАННЫХ НА ЛЕНТЕ С ДАННЫМИ В ПАМЯТИ, НАЧИНАЯ С АДРЕСА, С КОТОРОГО ЗАПИСЫВАЛСЯ НА ЛЕНТУ

ПЕРВЫЙ БАЙТ ДАННЫХ.

VERIFY "NAME" SCREEN\$

ИЛИ ЭКВИВАЛЕНТНО

VERIFY "NAME" CODE 16384,6912

ОДНАКО, ЭТО БУДЕТ ПРОВЕРКА УЖЕ ПРОВЕРЕННОГО ФАЙЛА.

3. МАССИВЫ.

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

3. МАССИВЫ.

LOAD "NAME" DATE LETTER()

ИЛИ

LOAD "NAME" DATE LETTER\$()

УНИЧТОЖАЕТ В ПАМЯТИ МАССИВ С ИМЕНЕМ "LETTER" ИЛИ "LETTER\$", ФОРМИРУЕТ НОВЫЙ МАССИВ И ПЕРЕПИСЫВАЕТ ТУДА ДАННЫЕ ИЗ ФАЙЛА "NAME". МОЖЕТ ВЫДАВАТЬ 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.

ДВА ОПЕРАТОРА LPRINT И LLIST ПОДОБНЫ ОПЕРАТОРАМ PRINT И LIST, НО С ТОЙ ЛИШЬ 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, ВВЕДВ BREAK (CAPS SHIFT И SPACE).

ЕСЛИ ВЫ ЗАДАЛИ ОПЕРАТОРЫ УПРАВЛЕНИЯ ПРИНТЕРОМ БЕЗ ПОДКЛЮЧЕННОГО РЕАЛЬНОГО УСТРОЙСТВА, ТО ВЫВОД ПРОСТО НЕ БУДЕТ И ВЫПОЛНЕНИЕ ПРОГРАММЫ ПРОДОЛЖИТСЯ СО СЛЕДУЮЩЕГО ОПЕРАТОРА. ТЕПЕРЬ ПОПРОБУЙТЕ ВЫПОЛНИТЬ ТАКУЮ 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 'TAB N', ПРОГРАММА БУДЕТ РАБОТАТЬ ТАКЖЕ, КАК И ПРЕЖДЕ. ТЕПЕРЬ ЗАМЕНИМ В СТРОКЕ 20 PRINT НА LPRINT И ЗАМЕТИМ, ЧТО РАЗВЕРТКИ ПО ДИАГОНАЛИ НЕ ПОЛУЧАЕТСЯ. А ЗАМЕНИВ ТЕПЕРЬ 'TAB N' НА 'AT 31-N,N' И СОХРАНИВ LPRINT, ПОЛУЧИМ ПО ОДНОМУ СИМВОЛУ НА СТРОКУ, ЧТО И ТРЕБОВАЛОСЬ ПОЛУЧИТЬ.

ВООБЩЕ, ПРИ ПЕЧАТИ ПЕРЕВОДА 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 ТАБ 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.

СТАНДАРТИМ 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.

5. БАЙТ, АДРЕСУЕМЫЙ ТО РАБОТОМ СОДЕРЖИТ ВЕРХНИЙ АДРЕС,
ИСПОЛЬЗУЕМЫЙ БЕЯСИКОМ.
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 (ПОДАНО 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 ЗНАЧЕНИ РАМТОР.

ИСПОЛЬЗУЯ ФУНКЦИЮ CLEAR, ВЫ МОЖЕТЕ СЖИМАТЬ РАМТОР, УВЕЛИЧИВАЯ ОБЛАСТЬ ДЛЯ БЕЙСИКА, УМЕНЬШАЯ ТЕМ САМЫМ ОБЛАСТЬ ОПРЕДЕЛЯЕМЫХ 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 MEMORY FULL" (ПЕРЕПОЛНЕНИЕ ПАМЯТИ);

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

МОЖНО ИЗМЕНИТЬ ДЛИТЕЛЬНОСТЬ ПОДАЧИ ЗВУКОВОГО СИГНАЛА, ИЗМЕНИВ ЧИСЛО ПО АДРЕСУ 23609. ПО УМОЛЧАНИЮ ПРЕДПОЛАГАЕТСЯ 64. ЧИСЛА (ЗА ИСКЛЮЧЕНИЕМ 0) МОГУТ ЗАПИСЫВАТЬСЯ В ПОКАЗАТЕЛЬНУЮ ФОРМУ КАК:

$M * 2^{**E}$,

ГДЕ М - МАНТИССА В ИНТЕРВАЛЕ 0.5 ... 1 (НЕ МОЖЕТ БЫТЬ 1);

Е - ЭКСПОНЕНТА, ПОЛОЖИТЕЛЬНОЕ ИЛИ ОТРИЦАТЕЛЬНОЕ ЧИСЛО.

ДОПУСТИМ, ВЫ ЗАПИСАЛИ "М" В ДВОИЧНОЙ СИСТЕМЕ СЧИСЛЕНИЯ,

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

1/2 ---> .1
1/4 ---> .01
3/4 ---> .11

НАШЕ ЧИСЛО "М" 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 3 БИТА МАНТИССЫ ВО ВТОРОЙ БАЙТ (МЫ ПОНИМАЕМ, ЧТО ПЕРВЫЙ БИТ ЭТО 1), ВТОРЫЕ 8 БИТОВ В ТРЕТИЙ БАЙТ И Т.Д.;

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

В) ЗАПИСЫВАЕМ В ПЕРВЫЙ БАЙТ ЭКСПОНЕНТУ +128.

НАПРИМЕР, МЫ ХОТИМ ЗАПИСАТЬ ЧИСЛО 4/5.

МАНТИССА БУДЕТ: .11001100110011001100110011001100

ПОСКОЛЬКУ 33-И БИТ РАВЕН 1, МЫ ДОЛЖНЫ ОКРУГЛИТЬ 32-И БИТ, ЗАПИСАВ 1 ВМЕСТО 0, А ЭКСПОНЕНТУ РАВНУ -3.

ТЕПЕРЬ ПРИМЕНЯЯ НАШИХ ТРИ ПРАВИЛА, ЗАПИШЕМ 5 БАЙТ:

!---ЗНАК ЧИСЛА

```

+-----+
| 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |
| 0 1 1 1 1 0 1 1 0 1 1 0 0 1 1 0 0 |
| 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 |
+-----+
-3+128   МАНТИССА 4/5, ИСКЛЮЧАЯ ЛЕВЫЙ ЗАКОНОВЫЙ 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 И 4 БАЙТА СОДЕРЖАТ МЛАДШИЕ И СТАРШИЕ ЗНАЧЕНИЕ БАЙТ ЧИСЛА (ИЛИ ЧИСЛО +131072), ЕСЛИ ОНО ОТРИЦАТЕЛЬНОЕ;
- Г) 5 БАЙТ РАВЕН 0.

ГЛАВА 25

СИСТЕМНЫЕ ПЕРЕМЕННЫЕ

БАЙТЫ ПАМЯТИ С 23552 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 БЕЙСИК-ПРОГРАММЫ ПО ИХ ИМЕНАМ. ИМЕНА ИСПОЛЬЗУЮТСЯ ТОЛЬКО ДЛЯ МНЕМОНИЧЕСКОГО ОБОЗНАЧЕНИЯ ЭТИХ ПЕРЕМЕННЫХ В ЭТОМ ОПИСАНИИ.

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

X - ПЕРЕМЕННАЯ НЕ ДОЛЖНА ИЗМЕНЯТЬСЯ, ТАК КАК ЭТО МОЖЕТ НАРУШИТЬ РАБОТУ СИСТЕМЫ.

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: PRINT PEEK N-256*PEEK(N+1)

ЗНАЧЕНИЕ	АДРЕС	ИМЯ	СОДЕРЖАНИЕ
1	2	3	4
N0	23552	KSTATE	ИСПОЛЬЗУЕТСЯ ПРИ ЧТЕНИИ С КЛАВИАТУРЫ
N1	23560	LAST K	ЗАПОМИНАЕТСЯ ВНОВЬ НАЖАТАЯ КЛАВИША
1	23561	REPER	ВРЕМЯ, В 50-ИХ ДОЛЯХ СЕКУНД, В ТЕЧЕНИЕ КОТОРОГО КЛАВИША ДОЛЖНА БЫТЬ ЗАФИКСИРОВАНА В НАЖАТОМ СОСТОЯНИИ. НАЧАЛЬНОЕ ЗНАЧЕНИЕ 35, НО МОЖЕТ БЫТЬ ИЗМЕНЕНО.
1	23562	REPPER	ЗАДЕРЖКА, В 50-ИХ ДОЛЯХ СЕКУНД, 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 ТЕЛЕВИЗ.
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 В CHARS АДРЕСА РАЗМЕЩЕНИЯ.

1	2	3	4	1	2	3	4
1	23608	RASP	ПРОДОЛЖИТЕЛЬНОСТЬ ЗВУКОВОГО СИГНАЛА.	N2	23656	MEM	АДРЕС ОБЛАСТИ, ИСПОЛЬЗУЕМОЙ КАК ПАМЯТЬ КАЛЬКУЛЯТОРА (ОБЫЧНО НЕИДЕАЛ, НО НЕ ВСЕГДА).
1	23609	PIP	ДЛИТЕЛЬНОСТЬ ЗАДЕРЖКИ, УСТРАНЯЮЩЕЙ АРЕБЕЗГ КЛАВИАТУРЫ.				
1	23610	ERR N	КОД СООБЩЕНИЯ -1. НАЧАЛЬНОЕ ЗНАЧЕНИЕ 255 (ДЛЯ '-1'), Т.Е. РЕК 23610 = 255.	1	23658	FLAG52	СТАРШИЕ ФЛАЖКИ
X1	23611	FLAGS	УПРАВЛЯЮЩИЕ ФЛАЖКИ БЕСИКА.	X1	23659	DF SZ	ЧИСЛО СТРОК (ВКЛЮЧАЯ И ОДИН ЧИСТУЮ) В НИЖНЕЙ ЧАСТИ ЭКРАНА.
X1	23612	TV FLAG	ФЛАЖОК ТЕЛЕВИЗОРА.	2	23660	S TOP	КОЛИЧЕСТВО ВЕРХНИХ СТРОК ПРОГРАММЫ В АВТОМАТИЧЕСКОМ ЛИСТИНГЕ.
X2	23613	ERR SP	АДРЕС В АППАРАТНОМ СТЕКЕ, ИСПОЛЬЗУЕМЫЙ КАК АДРЕС ВОЗВРАТА ПРИ ОШИБКЕ.	2	23662	OLDPPC	НОМЕР СТРОКИ, НА КОТОРУЮ УКАЗЫВАЕТ CONTINUE.
N1	23617	MODE	РЕЖИМ. СПЕЦИФИКАЦИЯ [K], [L], [C], [E] ИЛИ [G] КУРСОРА.	1	23664	OSPPC	НОМЕР ОПЕРАТОРА В СТРОКЕ, НА КОТОРУЮ УКАЗЫВАЕТ CONTINUE.
2	23618	NEW PPC	НОМЕР СТРОКИ, НА КОТОРУЮ ДОЛЖЕН БЫТЬ СДЕЛАН ПЕРЕХОД.	N1	23665	FLAGX	ПЕРЕМЕННЫЕ ФЛАЖКИ.
1	23623	SUB PPC	ПОРЯДКОВЫЙ НОМЕР ВЫПОЛНЯЕМОГО ОПЕРАТОРА В СТРОКЕ.	N2	23666	STR LEN	РАЗМЕР РАССТОЯНИИ МЕЖДУ СТРОКАМИ.
1	23624	DORDCR	ЦВЕТ РАМКИ ЭКРАНА, СОДЕРЖИТ АТРИБУТЫ.	N2	23668	T ADDR	АДРЕС СЛЕДУЮЩЕГО СИМВОЛА В СИНТАКСИЧЕСКОЙ ТАБЛИЦЕ.
2	23625	E PPC	КОЛИЧЕСТВО ТЕКУЩИХ СТРОК (С КУРСОРОМ).	2	23670	SEED	НАЧАЛЬНОЕ ЗНАЧЕНИЕ ДЛЯ RND, ИЗМЕНЯЕТСЯ ФУНКЦИЕЙ RANDOMIZE.
X2	23627	VAR5	АДРЕСА ПЕРЕМЕННЫХ.	3	23672	FRAMES	СЧЕТЧИК КАДРОВ - ПРИРАЩЕНИЕ ЧЕРЕЗ КАЖДЫЕ 20 MS (СМ. ГЛАВУ 18).
N2	23629	DEST	АДРЕС ПЕРЕМЕННОЙ В ЗАДАНИИ.	2	23675	UD6	АДРЕС ПЕРВОГО ОПРЕДЕЛЯЕМОГО ПОЛЬЗОВАТЕЛЕЙ СИМВОЛА.
X2	23631	CHANS	АДРЕС КАНАЛА ДАННЫХ.	1	23677	COORDS	X - КООРДИНАТА ТОЧКИ ГРАФОПОСТРОИТЕЛЯ
X2	23633	CURCHL	АДРЕС ДАННЫХ ДЛЯ ВВОДА-ВЫВОДА.	1	23678		Y - КООРДИНАТА ТОЧКИ ГРАФОПОСТРОИТЕЛЯ
X2	23635	PRO6	АДРЕС БЕСИК-ПРОГРАММЫ.	1	23679	P POSN	33 - ПОЗИЦИОННОЕ ЧИСЛО ДЛЯ ПОЗИЦИОНИРОВАНИЯ ПРИНТЕРА.
X2	23637	NXTLIN	АДРЕС СЛЕДУЮЩЕЙ СТРОКИ В ПРОГРАММЕ.	1	23680	PR CC	МЛАДШИЙ БАЙТ АДРЕСА ПОЗИЦИИ ДЛЯ LPRINT ДЛЯ ПЕЧАТИ.
X2	23639	DATADD	АДРЕС ТЕРМИНАТОРА ПОСЛЕДНЕГО СИМВОЛА В ДАТА.	1	23681		НЕ ИСПОЛЬЗУЕТСЯ
X2	23641	E LINE	АДРЕС ВЫВЕДЕННОЙ КОМАНДЫ.	2	23682	ECHO E	33-ПОЗИЦИОННОЕ И 24-СТРОКОВОЕ ЧИСЛА (В НИЖНЕЙ ПОЛОВИНЕ) КОНЦА ВХОДНОГО БУФЕРА
2	23643	K CUR	АДРЕС КУРСОРА.	2	23684	DF CC	АДРЕС PRINT-ПОЗИЦИИ В ОБЛАСТИ ЭКРАНА
X2	23645	CH ADD	АДРЕС СЛЕДУЮЩЕГО ИНТЕРПРЕТИРУЕМОГО СИМВОЛА: СИМВОЛ АРГУМЕНТА В РЕК, NEW, LINE ИЛИ РОКЕ ОПЕРАТОРАХ.	2	23686	DF CCL	ПОДОБНО DF CC ДЛЯ НИЖНЕЙ ЧАСТИ ЭКРАНА
2	23647	X PRT	АДРЕС СИМВОЛА СЛЕДУЮЩЕГО ЗА МАРКЕРОМ [?].	X1	23688	S POSN	33-ПОЗИЦИОННОЕ ЧИСЛО ДЛЯ PRINT ПОЗИЦИИ
X2	23649	WORK SP	АДРЕС ВРЕМЕННОЙ РАБОЧЕЙ ОБЛАСТИ.	X1	23689		23-СТРОКОВОЕ ЧИСЛО ДЛЯ PRINT ПОЗИЦИИ
X2	23651	STK BOT	АДРЕС "ДНА" ПРОГРАММИРУЕМОГО СТЕКА.	X2	23690	S POSNL	ПОДОБНО S POSN ДЛЯ НИЖНЕЙ ЧАСТИ
X2	23653	STK END	АДРЕС НАЧАЛА РЕЗЕРВНОЙ ОБЛАСТИ ПАМЯТИ.				
N1	23655	BREB	В-РЕГИСТР КАЛЬКУЛЯТОРА.				

```

: 1 : 2 : 3 : 4 :
: 1 : 23692 : SCR ST : СЧЕТЧИК СВЕРТОВ: ВСЕГДА НА 1 БОЛЬШЕ :
: : : : ЧИСЛА СВЕРТОВ, КОТОРЫЕ ДОЛЖНЫ БЫТЬ :
: : : : ПРОВЕДЕНЫ ПЕРЕД ОСТАНОВКОМ СО СВЕРТКОЙ :
: : : : ЕСЛИ ВЫ УСТАНОВИТЕ ЭТО ЧИСЛО БОЛЬШЕ, :
: : : : ЧЕМ НА 1 (СКАЖЕМ 255), ТО СКРАН :
: : : : БУДЕТ СВОРАЧИВАТЬСЯ БЕЗ ЗАПРОСА К ВАМ. :
: 1 : 23693 : ATTR P : СПЛОШНЫЕ ЦВЕТА :
: 1 : 23694 : MASK P : ИСПОЛЬЗУЕТСЯ ДЛЯ ВЫСВЕЧИВАНИЯ ЦВЕТОВ :
: : : : БИТ, УСТАНОВЛЕННЫЕ В 1, ПОКАЗЫВАЕТ, :
: : : : БИТЫ АТТРИБУТОВ ВЕРТСЯ НЕ ИЗ ATTR P, :
: : : : 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 : СТАРШЕЕ ФЛАЖКИ :
: N30 : 23698 : MEMBOT : ОБЛАСТЬ ПАМЯТИ ДЛЯ КАЛЬКУЛЯТОРА. :
: : : : ИСПОЛЬЗУЕТСЯ ДЛЯ ЗАПИСИ ЧИСЕЛ, :
: : : : КОТОРЫЕ НЕ МОГУТ БЫТЬ РАЗМЕЩЕНЫ В :
: : : : ПРОГРАММИРУЕМОМ СТЕКЕ КАЛЬКУЛЯТОРА. :
: 2 : 23728 : : НЕ ИСПОЛЬЗУЕТСЯ :
: 2 : 23730 : RANTOP : АДРЕС ПОСЛЕДНЕГО БАЙТА ОБЛАСТИ :
: : : : БЕЙСИК-СИСТЕМЫ. :
: 2 : 23732 : P-RANT : АДРЕС ПОСЛЕДНЕГО БАЙТА ФИЗИЧЕСКОГО :
: : : : ОЗУ. :

```

СЛЕДУЮЩАЯ ПРОГРАММА ДАСТ 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(23755+N)
И ВЫ ДОПОЛНИТЕЛЬНО ПОЛУЧИТЕ ДУМП САМОЙ ПРОГРАММЫ.

```

ГЛАВА 26

ИСПОЛЬЗОВАНИЕ МАШИННЫХ КОДОВ

КРАТКОЕ СОДЕРЖАНИЕ: USR С ЧИСЛОВЫМ АРГУМЕНТОМ.
ЭТА ГЛАВА ОПИСЫВАЕТ ПРИМЕНЕНИЕ МАШИННЫХ КОМАНД МИКРОПРОЦЕССОРА Z80.

ПРОГРАММЫ В МАШИННЫХ КОДАХ ПИШУТСЯ ОБЫЧНО НА АССЕМБЛЕРЕ С ПОСЛЕДУЮЩЕЙ ТРАНСЛЯЦИЕЙ. ПЕРЕЧЕНЬ МНЕМОКОДОВ КОМАНД МИКРОПРОЦЕССОРА Z80 ПРИВЕДЕН С ПРИЛОЖЕНИЕМ А). ТРАНСЛЯТОР С АССЕМБЛЕРА ВСТРОЕН В КОМПЬЮТЕР ZX SPECTRUM.

ПРИВЕДЕМ ПРИМЕР ПРОГРАММЫ:

```
LD BC,99
```

```
RET
```

КОТОРАЯ ЗАГРУЖАЕТСЯ В "BC" РЕГИСТР ЧИСЛО 99. ЭТА ПРОГРАММА БУДЕТ ТРАНСЛИРОВАТЬСЯ В 4-Х БАЙТНУЮ МАШИННУЮ КОДУ:

БАЙТЫ 1, 99, 0 ДЛЯ LD BC,99 И 201 ДЛЯ RET

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

```

: ОПРЕДЕЛЯЕМЫЕ :
: ПОЛЬЗОВАТЕЛЕМ СИМВОЛЫ :

```

```
RANTOP=32599 UDG=32600 PRANT=32767
```

ЕСЛИ ВЫ ТЕПЕРЬ ВЫПОЛНИТЕ:
CLEAR 32499

ТО ПОЛУЧИТЕ ДОПОЛНИТЕЛЬНО 100 БАЙТОВ ПАМЯТИ, НАЧИНАЯ С АДРЕСА 32500.

```

! 100 ! ! ОПРЕДЕЛЯЕМЫЕ ! !
! БАЙТОВ ! ! ПОЛЬЗОВАТЕЛЕМ СИМВОЛЫ ! !

```

```
RANTOP 32500 UDG=32600 PRANT=32767
=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 ЗАВЕРШАТЬСЯ С 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 СЫЧНЫМ ОБРАЗОМ ВО КОМАНДЕ МИКРОПРОЦЕССОРА RET. В МАШИННОЙ ПРОГРАММЕ ВЫ НЕ ДОЛЖНЫ ИСПОЛЬЗОВАТЬ РЕГИСТРЫ IX И I.

ВЫ МОЖЕТЕ ЗАПИСАТЬ ВАШУ ПРОГРАММУ НА ЛЕНТУ:

```
SAVE "NAME" CODE 32500,4
```

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

```
10 LOAD " " CODE 32500,4
```

```
20 PRINT USR 32500
```

ДЛЯ ЧЕГО НАДО СДЕЛАТЬ:

```
SAVE "NAME" LINE 10
```

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.

ДАЛЕЕ ПРИВОДЯТСЯ 78 КОМАНД МИКРОПРОЦЕССОРА 8085 СОВМЕСТИМЫХ С МИКРОПРОЦЕССОРОМ Z80 (158 КОМАНД).

```

: ИМЕМОНИКА : ДЕЙСТВИЕ : КОД : ЦИКЛЫ :

```

```

: 1 : 2 : 3 : 4 :

```

ПЕРЕСИЛКА, 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 PR,DD : ЗАГРУЗИТЬ НЕПОСР. В ДВА РЕГ. : 00PR 0001 : 10
```

1	2	3	4	1	2	3	4
: STAX B	: ЗАПOMНИТЬ A КОСВЕННО ПО ВС	: 0000 0010 : 7	:	:	ПЕРЕДАЧИ УПРАВЛЕНИЯ.	:	:
: STAX D	: ЗАПOMНИТЬ A КОСВЕННО ПО DE	: 0001 0010 : 7	:	:		:	:
: LDAX B	: ЗАГРУЗИТЬ A КОСВЕННО ПО ВС	: 0000 1010 : 7	:	: JMP DD	: БЕЗУСЛОВНЫЙ ПЕРЕХОД	: 1100 0011 : 10	:
: LDAX D	: ЗАГРУЗИТЬ A КОСВЕННО ПО DE	: 0001 1010 : 7	:	: JCC DD	: ПЕРЕЙТИ ПО УСЛОВИЮ CC	: 1100 0010 : 10	:
: STA DD	: ЗАПOMНИТЬ A ПО АДРЕСУ DD	: 0011 0010 : 13	:	: CALL DD	: БЕЗУСЛОВНЫЙ ВЫЗОВ	: 1100 1101 : 17	:
: LDA DD	: ЗАГРУЗИТЬ A ПО АДРЕСУ DD	: 0011 1010 : 13	:	: CCC DD	: ВЫЗОВ ПО УСЛОВИЮ CC	: 1100 1100 : 17(10)	:
: SHLD DD	: ЗАПOMНИТЬ H, L ПО АДРЕСУ DD	: 0010 0010 : 16	:	: RET	: ВОЗВРАТ ПОСЛЕ ВЫЗОВА	: 1100 1001 : 10	:
: LHLD DD	: ЗАГРУЗИТЬ H, L ПО АДРЕСУ DD	: 0010 1010 : 16	:	: RCC	: ВОЗВРАТ ПО УСЛОВИЮ CC	: 1100 0000 : 11(5)	:
: XCHG	: ОБМЕНЯТЬ DE И HL	: 1110 1011 : 4	:	: RST	: ВОЗВРАТ ПО УСЛОВИЮ CC	: 1100 1111 : 11	:
			:	: PCHL N	: ПЕРЕДАТЬ HL В PC	: 1110 1001 : 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	: 00PR 0011 : 6	:
: DCX PR	: УМЕНЬШЕНИЕ ПАРЫ РЕГИСТРОВ	: 00PR 1011 : 6	:

АРИФМЕТИЧЕСКИЕ И ЛОГИЧЕСКИЕ

: ADD R	: ПРИБАВИТЬ РЕГИСТР K A	: 1000 0RRR : 4(7)	:
: ADC R	: ПРИБАВИТЬ РЕГ. K A С ПЕРЕНОСОМ	: 1000 1RRR : 4(7)	:
: SUB R	: ВЫЧЕСТЬ РЕГИСТР ИЗ A	: 1001 0RRR : 4(7)	:
: SBB R	: ВЫЧЕСТЬ С ЗАЕМОМ	: 1001 1RRR : 4(7)	:
: ANA R	: РЕГИСТР AND A	: 1010 0RRR : 4(7)	:
: XRA R	: ИСКЛ. ИЛИ РЕГИСТРА И A	: 1010 1RRR : 4(7)	:
: ORA R	: РЕГИСТР OR A	: 1011 0RRR : 4(7)	:
: CMA R	: СРАВНИТЬ РЕГИСТР И A	: 1011 1RRR : 4(7)	:
: ADI D	: ПРИБАВИТЬ НЕПОСР. ДАННЫЕ K A	: 1100 0110 : 7	:
: ACI D	: ПРИБАВИТЬ НЕПОСР. С ПЕРЕНОСОМ	: 1100 1110 : 7	:
: SUI D	: ВЫЧЕСТЬ НЕПОСР. ИЗ A	: 1101 0110 : 7	:
: SBI D	: ВЫЧЕСТЬ НЕПОСР. С ЗАЕМОМ	: 1101 1110 : 7	:
: ANI D	: НЕПОСР. AND A	: 1110 1110 : 7	:
: XRI D	: ИСКЛ. ИЛИ НЕПОСР. И A	: 1111 0110 : 7	:
: ORI D	: НЕПОСР. OR A	: 1111 0110 : 7	:
: CPI D	: СРАВНИТЬ НЕПОСР. С A	: 1111 1110 : 7	:
: DAD RP	: ПРИБАВИТЬ ПАРУ РЕГИСТРОВ K HL	: 00PR 1001 : 11	:

ОПЕРАЦИИ С НАКОПИТЕЛЯМИ И ФЛАГАМИ

: RLC	: СДВИНУТЬ A ВЛЕВО	: 0000 0111 : 4	:
: RRC	: СДВИНУТЬ A ВПРАВО	: 0000 1111 : 4	:
: RAL	: СДВИНУТЬ A ВЛЕВО ЧЕР. РАЗР. ПЕР.	: 0001 0111 : 4	:
: RAR	: СДВИНУТЬ A ВПРАВО ЧЕР. РАЗР. ПЕР.	: 0001 1111 : 4	:
: DAA	: ДЕСЯТИЧ. КОРРЕКЦИЯ НАКОПИТЕЛЯ	: 0010 0111 : 4	:
: CMA	: ДОПОЛНЕНИЕ K НАКОПИТЕЛЮ	: 0010 1111 : 4	:
: SIC	: УСТАНОВИТЬ БИТ ПЕРЕНОСА	: 0011 0111 : 4	:
: SMS	: ОБРАТИТЬ БИТ ПЕРЕНОСА	: 0011 1111 : 4	:

I/O УПРАВЛЕНИЕ И ОПЕРАЦИИ СО СТЕКОМ

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

ОБОЗНАЧЕНИЯ:

1. ПОЛЯ ДАННЫХ.

D - ОДИН БАЙТ НЕПОСР. ДАННЫХ (ДЛИНА КОМАНДЫ 2 БАЙТА)
 DD - ДВУХБАЙТОВЫЙ АДРЕС (ДЛИНА КОМАНДЫ 3 БАЙТА)
 ВСЕ ОСТАЛЬНЫЕ КОМАНДЫ ИМЕЮТ ДЛИНУ 1 БАЙТ.

2. ДИКЛМ.

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

ПОЛЯ РЕГИСТРОВ.

: "R" : RRR :	: "RP" : PP :
: B : 000 :	: DE : 00 :
: C : 001 :	: HL : 10 :
: D : 010 :	: PS : 11 :
: E : 011 :	: PSM : 11 :
: H : 100 :	
: L : 101 :	
: M : 110 :	
: A : 111 :	

КОДЫ УСЛОВИЙ.

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

ПРИЛОЖЕНИЕ А

ПОЛНЫЙ НАБОР СИМВОЛОВ

АЕС. : СИМВОЛ : ШЕСТН. : АССЕМБЛЕР. : СВН... : EDN...
 КОД : : КОД % МНЕМОНИКА :

1	2	3	4	5	6
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 C	
10	КУРС. ВНИЗ	0A	LD A, (BC)	RRC D	
11	КУРС. 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, N	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, N	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 D	
35	#	23	INC HL	SLA E	
36	\$	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 HL	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 NS, DIS		
49	1	31	LD SP, NN		
50	2	32	LD (NN), A		
51	3	33	INC SP		
52	4	34	INC (HL)		

1	2	3	4	5	6
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)	IN 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	49	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	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)	IN 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), 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)	IN 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 (СТРОЧН)	61	LD H, C	BIT 4, C	OUT (C), H
98	B (СТРОЧН)	62	LD H, D	BIT 4, D	SBC HL, HL
99	C (СТРОЧН)	63	LD H, E	BIT 4, E	LD (NN), HL
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, HL
107	K (СТРОЧН)	6B	LD L, E	BIT 5, E	LD HL, (NN)
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	
112	P (СТРОЧН)	70	LD (HL), B	BIT 6, B	
113	Q (СТРОЧН)	71	LD (HL), C	BIT 6, C	

1	2	3	4	5	6	1	2	3	4	5	6
:114 : R (CTPO4H)	: 72 :	LD (HL),D	: BIT 6,D	:		:175 : CODE	: AF :	XOR A	: RES 5,A	:	
:115 : S (CTPO4H)	: 73 :	LD (HL),E	: BIT 6,E	:		:176 : VAL	: B0 :	OR B	: RES 6,B	: LDIR	:
:116 : T (CTPO4H)	: 74 :	LD (HL),H	: BIT 6,H	:		:177 : LEN	: B1 :	OR C	: RES 6,C	: CPIR	:
:117 : U (CTPO4H)	: 75 :	LD (HL),L	: BIT 6,L	:		:178 : SIN	: B2 :	OR D	: RES 6,D	: INIR	:
:118 : V (CTPO4H)	: 76 :	HALT	: BIT 6,(HL)	:		:179 : COS	: B3 :	OR E	: RES 6,E	: OTIR	:
:119 : W (CTPO4H)	: 77 :	LD (HL),A	: BIT 6,A	:		:180 : TAN	: B4 :	OR H	: RES 6,H	:	:
:120 : X (CTPO4H)	: 78 :	LD A,B	: BIT 7,B	: IN A,(C)		:181 : ASH	: B5 :	OR L	: RES 6,L	:	:
:121 : Y (CTPO4H)	: 79 :	LD A,C	: BIT 7,C	: OUT (C),A		:182 : ACS	: B6 :	OR (HL)	: RES 6,(HL)	:	:
:122 : Z (CTPO4H)	: 7A :	LD A,D	: BIT 7,D	: ADC HL,SP		:183 : ATN	: B7 :	OR A	: RES 6,A	:	:
:123 : {	: 7B :	LD A,E	: BIT 7,E	: LD SP,(NN)		:184 : LN	: B8 :	CP B	: RES 7,B	: LDDR	:
:124 : !	: 7C :	LD A,H	: BIT 7,H	:		:185 : EXP	: B9 :	CP C	: RES 7,C	: CPDR	:
:125 : 3	: 7D :	LD A,L	: BIT 7,L	:		:186 : INT	: BA :	CP D	: RES 7,D	: INDR	:
:126 :	: 7E :	LD A,(HL)	: BIT 7,H	:		:187 : SQR	: BB :	CP E	: RES 7,E	: OTDR	:
:127 : ©	: 7F :	LD A,A	: BIT 7,A	:		:188 : SGN	: BC :	CP H	: RES 7,H	:	:
:128 : □	: 80 :	ADD A,B	: RES 0,B	:		:189 : ABS	: BD :	CP L	: RES 7,L	:	:
:129 : □	: 81 :	ADD A,C	: RES 0,C	:		:190 : PEEK	: BE :	CP (HL)	: RES 7,(HL)	:	:
:130 : □	: 82 :	ADD A,D	: RES 0,D	:		:191 : IN	: BF :	CP A	: RES 7,A	:	:
:131 : □	: 83 :	ADD A,E	: RES 0,E	:		:192 : USR	: C0 :	RET NZ	: SET 0,B	:	:
:132 : □	: 84 :	ADD A,H	: RES 0,H	:		:193 : STR\$	: C1 :	POP BC	: SET 0,C	:	:
:133 : □	: 85 :	ADD A,L	: RES 0,L	:		:194 : CHR\$	: C2 :	JP NZ,NN	: SET 0,D	:	:
:134 : □	: 86 :	ADD A,(HL)	: RES 0,(HL)	:		:195 : NOT	: C3 :	JP NN	: SET 0,E	:	:
:135 : □	: 87 :	ADD A,A	: RES 0,A	:		:196 : BIN	: C4 :	CALL NZ,NN	: SET 0,H	:	:
:136 : □	: 88 :	ADC A,B	: RES 1,B	:		:197 : OR	: C5 :	PUSH BC	: SET 0,L	:	:
:137 : □	: 89 :	ADC A,C	: RES 1,C	:		:198 : AND	: C6 :	ADD A,N	: SET 0,(HL)	:	:
:138 : □	: 8A :	ADC A,D	: RES 1,D	:		:199 : <=	: C7 :	RST 0	: SET 0,A	:	:
:139 : □	: 8B :	ADC A,E	: RES 1,E	:		:200 : >=	: C8 :	RET Z	: SET 1,B	:	:
:140 : □	: 8C :	ADC A,H	: RES 1,H	:		:201 : <>	: C9 :	RET	: SET 1,C	:	:
:141 : □	: 8D :	ADC A,L	: RES 1,L	:		:202 : LINE	: CA :	JP Z,NN	: SET 1,D	:	:
:142 : □	: 8E :	ADC A,(HL)	: RES 1,(HL)	:		:203 : THEN	: CB :		: SET 1,E	:	:
:143 : □	: 8F :	ADC A,A	: RES 1,A	:		:204 : TO	: CC :	CALL Z,NN	: SET 1,H	:	:
:144 : (A)	: 90 :	SUB B	: RES 2,B	:		:205 : STEP	: CD :	CALL NN	: SET 1,L	:	:
:145 : (B)	: 91 :	SUB C	: RES 2,C	:		:206 : DEF FN	: CE :	ADC A,N	: SET 1,(HL)	:	:
:146 : (C)	: 92 :	SUB D	: RES 2,D	:		:207 : LIT	: CF :	RST 8	: SET 1,A	:	:
:147 : (D)	: 93 :	SUB E	: RES 2,E	:		:208 : FORMAT	: D0 :	RET NC	: SET 2,B	:	:
:148 : (E)	: 94 :	SUB H	: RES 2,H	:		:209 : MOVE	: D1 :	POP DE	: SET 2,C	:	:
:149 : (F)	: 95 :	SUB L	: RES 2,L	:		:210 : ERASE	: D2 :	JP NC,NN	: SET 2,D	:	:
:150 : (G)	: 96 :	SUB (HL)	: RES 2,(HL)	:		:211 : OPEN	: D3 :	OUT (N),A	: SET 2,E	:	:
:151 : (H)	: 97 :	SUB A	: RES 2,A	:		:212 : CLOSE	: D4 :	CALL NC,NN	: SET 2,H	:	:
:152 : (I)	: 98 :	SBC A,B	: RES 3,B	:		:213 : MERGE	: D5 :	PUSH DE	: SET 2,L	:	:
:153 : (J)	: 99 :	SBC A,C	: RES 3,C	:		:214 : VERIFY	: D6 :	SUB N	: SET 2,(HL)	:	:
:154 : (K)	: 9A :	SBC A,D	: RES 3,D	:		:215 : DEER	: D7 :	RST 16	: SET 2,A	:	:
:155 : (L)	: 9B :	SBC A,E	: RES 3,E	:		:216 : CIRCLE	: D8 :	RET C	: SET 3,B	:	:
:156 : (M)	: 9C :	SBC A,H	: RES 3,H	:		:217 : INK	: D9 :	EXX	: SET 3,C	:	:
:157 : (N)	: 9D :	SBC A,L	: RES 3,L	:		:218 : PAPER	: DA :	JP C,NN	: SET 3,D	:	:
:158 : (O)	: 9E :	SBC A,(HL)	: RES 3,(HL)	:		:219 : FLASH	: DB :	IN A,(N)	: SET 3,E	:	:
:159 : (P)	: 9F :	SBC A,A	: RES 3,A	:		:220 : BRIGHT	: DC :	CALL C,NN	: SET 3,H	:	:
:160 : (Q)	: A0 :	AND B	: RES 4,B	: LDI		:221 : INVERSE	: DD :	PREFIXES	: SET 3,L	:	:
:161 : (R)	: A1 :	AND C	: RES 4,C	: CPI		:	:	INSTRUCTIONS:	:	:	:
:162 : (S)	: A2 :	AND D	: RES 4,D	: INI		:	:	USING IX	:	:	:
:163 : (T)	: A3 :	AND E	: RES 4,E	: OUTI		:222 : OVER	: DE :	SBC A,H	: SET 3,(HL)	:	:
:164 : (U)	: A4 :	AND H	: RES 4,H	:		:223 : OUT	: DF :	RST 24	: SET 3,A	:	:
:165 : RND	: A5 :	AND L	: RES 4,L	:		:224 : L-RINT	: E0 :	RET PO	: SET 4,B	:	:
:166 : INKEY\$	: A6 :	AND (HL)	: RES 4,(HL)	:		:225 : LLIST	: E1 :	POP HL	: SET 4,C	:	:
:167 : PI	: A7 :	AND A	: RES 4,A	:		:226 : STOP	: E2 :	JP PO,NN	: SET 4,D	:	:
:168 : FN	: A8 :	XOR B	: RES 5,B	: LDD		:227 : READ	: E3 :	EX (SP),HL	: SET 4,E	:	:
:169 : POINT	: A9 :	XOR C	: RES 5,C	: CPD		:228 : DATA	: E4 :	CALL PO,NN	: SET 4,H	:	:
:170 : SCREEN\$	: AA :	XOR D	: RES 5,D	: IND		:229 : RESTORE	: E5 :	PUSH HL	: SET 4,L	:	:
:171 : ATTR	: AB :	XOR E	: RES 5,E	: OUTD		:230 : NEW	: E6 :	AND N	: SET 4,(HL)	:	:
:172 : AT	: AC :	XOR H	: RES 5,H	:		:231 : BORDER	: E7 :	RST 32	: SET 4,A	:	:
:173 : TAB	: AD :	XOR L	: RES 5,L	:		:232 : CONTINUE	: E8 :	RET PE	: SET 5,B	:	:
:174 : VAL\$	: AE :	XOR (HL)	: RES 5,(HL)	:		:233 : DIA	: E9 :	JP (HL)	: SET 5,C	:	:

1	2	3	4	5	6	1	2	3
234	REM	EA	JP PE,NN	SET 5,D		2	VARIABLE NOT FOUND (ПЕРЕМЕННАЯ НЕ НАЙДЕНА)	РАЗНОЕ
235	FOR	EB	EX DE,HL	SET 5,E				
236	GO TO	EC	CALL PE,NN	SET 5,H			ДЛЯ ПРОСТОЙ ПЕРЕМЕННОЙ ВЫДАЕТСЯ, ЕСЛИ ОНА	
237	GO SUB	ED		SET 5,L			ИСПОЛЬЗУЕТСЯ БЕЗ ПРЕДВАРИТЕЛЬНОГО ОПРЕДЕЛЕНИЯ	
238	INPUT	EE	XOR N	SET 5,(HL)			В ОПЕРАТОРАХ LET, READ ИЛИ INPUT, ИЛИ	
239	LOAD	EF	RST 40	SET 5,A			ЗАГРУЖАЕТСЯ С ЛЕНТЫ ИЛИ УСТАНАВЛИВАЕТСЯ	
240	LIST	F0	RET P	SET 6,B			В ОПЕРАТОРЕ FOR.	
241	LET	F1	POP AF	SET 6,C			ДЛЯ ИНДЕКСИРУЕМОЙ ПЕРЕМЕННОЙ СООБЩЕНИЕ ВЫ-	
242	PAUSE	F2	JP P,NN	SET 6,D			ДАЕТСЯ, ЕСЛИ ОНА НЕ БЫЛА ПРЕДВАРИТЕЛЬНО	
243	NEXT	F3	DI	SET 6,E			ОПРЕДЕЛЕНА В ОПЕРАТОРЕ DIM ПЕРЕД ИСПОЛЬЗОВА-	
244	POKE	F4	CALL P,NN	SET 6,H			НИЕМ ИЛИ ЗАГРУЗКОЙ С ЛЕНТЫ.	
245	PRINT	F5	PUSH AF	SET 6,L				
246	PLOT	F6	DR N	SET 6,(HL)		3	SUBSCRIPT WRONG (ОШИБОЧНЫЙ ИНДЕКС)	В ИНДЕКСНОЙ
247	RUN	F7	RST 48	SET 6,A				ПЕРЕМЕННОЙ
248	SAVE	F8	RET M	SET 7,B			ИНДЕКС ПРЕВЫШАЕТ РАЗМЕРНОСТЬ МАССИВА, ЛИБО	ИЛИ
249	RANDOMIZE	F9	LD SP,HL	SET 7,C			ОШИБОЧНОЕ ЧИСЛО ЗАДАЕТ ИНДЕКС. ЕСЛИ ИНДЕКС	ПОДСТРОКЕ
250	IF	FA	JP M,NN	SET 7,D			СТРИКАТЕЛЬНЫЙ ИЛИ БОЛЬШЕ 65535, ТО ВЫДАЕТСЯ	
251	CLS	FB	EI	SET 7,E			СООБЩЕНИЕ В.	
252	DRAW	FC	CALL M,NN	SET 7,H				
253	CLEAR	FD	PREFIXES	SET 7,L		4	OUT OF MEMORY (ВНЕ ПАМЯТИ)	LET,
			INSTRUCTIONS:					INPUT,
			USING TY					FOR,
254	RETURN	FE	CP N	SET 7,(HL)			В ПАМЯТИ НЕДОСТАТОЧНО МЕСТА ДЛЯ ВАШИХ ДЕЙ-	DIM,
255	COPY	FF	RST 56	SET 7,A			СТВИЙ. ВЫ МОЖЕТЕ ОСВОБОДИТЬ СЕБЕ ПАМЯТЬ	GO SUB,
							УДАЛИВ КОМАНДНЫЕ СТРОКИ, ИСПОЛЬЗУЯ DELETE,	LOAD,
							ЗАТЕМ УДАЛИТЬ ОДНУ ИЛИ ДВЕ СТРОКИ ПРОГРАММЫ	MERGE
							(С ЦЕЛЬЮ ВОЗВРАТА ИХ В ПОСЛЕДСТВИЕ), ПОЛУ-	
							ЧИТЬ ДОПОЛНИТЕЛЬНУЮ ПАМЯТЬ МАНЕВРИРУЯ	
							ОПЕРАТОРОМ CLEAR.	
						5	OUT OF SCREEN (ВНЕ ЭКРАНА)	INPUT,
								PRINT AT
							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 ВЫЧИСЛЕНИЙ ПОЛУЧИЛОСЬ ЧИСЛО	
							БОЛЬШЕ 10**38.	
						7	RETURN WITHOUT GO SUB (RETURN БЕЗ GO SUB)	RETURN
							ВСТРЕТИЛОСЬ БОЛЬШЕ ОПЕРАТОРОВ RETURN, ЧЕМ	
							БЫЛО ОПЕРАТОРОВ GO SUB.	
						8	END OF FILE (КОНЕЦ ФАЙЛА)	ОПЕРАЦИИ
								С ВНЕШНЕЙ
								ПАМЯТЬЮ
						9	STOP STATEMENT (ОПЕРАТОР STOP)	STOP
							ПОСЛЕ ЭТОГО СООБЩЕНИЯ CONTINUE НЕ МОЖЕТ	
							ПОВТОРИТЬ STOP, НО МОЖЕТ ПЕРЕДАТЬ УПРАВЛЕНИЕ	
							НА СЛЕДУЮЩЕЙ ОПЕРАТОР.	
1	NEXT WITHOUT FOR (NEXT БЕЗ FOR)				NEXT			
	УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ НЕТ (НЕ БЫЛА ОПРЕДЕ-							
	ЛЕНА В ОПЕРАТОРЕ FOR), НО ЕСТЬ ОБЫЧНАЯ							
	ПЕРЕМЕННАЯ С ТЕМ ЖЕ ИМЕНЕМ.							

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 сообщение помогает найти ошибочную строку и ошибочный оператор в этой строке. (Команда указывается как строка 0, оператор 1 располагается в строке первый, оператор 2 следует после первого THEN и т.д.).

От состояния CONTINUE зависит очень многое в сообщениях. Обычно продолжение начинается с оператора, специфицированного в предыдущем сообщении, но имеются исключения - сообщения 0, 9 и 0 (смотри также приложение С).

КОД	ЗНАЧЕНИЕ	СИТУАЦИЯ
1	2	3
0	OK (О'КЕИ!)	РАЗНОЕ
	УСПЕШНОЕ ЗАВЕРШЕНИЕ ИЛИ ПЕРЕХОД НА СТРОКУ С	
	НОМЕРОМ БОЛЬШИМ ЧЕМ ИМЕЕТСЯ ВСЕГО.	
	ЭТО СООБЩЕНИЕ НЕ МЕНЯЕТ СТРОКИ ИЛИ ОПЕРАТОРА:	
	ОПРЕДЕЛЕННОГО ДЛЯ CONTINUE.	
1	NEXT WITHOUT FOR (NEXT БЕЗ FOR)	NEXT
	УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ НЕТ (НЕ БЫЛА ОПРЕДЕ-	
	ЛЕНА В ОПЕРАТОРЕ FOR), НО ЕСТЬ ОБЫЧНАЯ	
	ПЕРЕМЕННАЯ С ТЕМ ЖЕ ИМЕНЕМ.	

1	2	3	1	2	3
A	INVALID ARGUMENT (ОШИБОЧНЫЙ АРГУМЕНТ)	SQR,	H	STOP IN INPUT	INPUT
	АРГУМЕНТ ФУНКЦИИ НЕ ДОПУСТИМ В ДАННОЙ ВЕРСИИ	LN,		НЕКОТОРЫЕ ВВЕДЕННЫЕ ДАННЫЕ НАЧИНАЮТСЯ С	
		ASH,		ОПЕРАТОРА STOP ИЛИ БЫЛА НАЖАТА INPUT LINE.	
		ACS,		ДЕЙСТВИЕ CONTINUE - ОБЫЧНОЕ.	
		USR (CO			
		СТРОКОВЫМ			
		АРГУМЕНТОМ)	I	FOR WITHOUT NEXT (FOR БЕЗ NEXT)	FOR
				ЦИКЛ FOR НИ РАЗУ НЕ ВЫПОЛНЯЛСЯ, НЕ НАЙДЕН	
B	INTEGER OUT OF RANGE (ПЕРЕПОЛНЕНИЕ ЦЕЛОГО)	RUN,		NEXT ОПЕРАТОР.	
	ВЫАЕТСЯ, КОГДА АРГУМЕНТ С ПЛАВАЮЩЕЙ ТОЧКОЙ	RANDOMIZE			
	ОКРУГЛЯЕТСЯ К ЦЕЛОМУ. ДЛЯ СЛУЧАЯ МАССИВОВ	POKE,			
	СМОТРИ ТАКЖЕ СООБЩЕНИЕ 3.	DIM,	J	INVALID I/O DEVICE (НЕВЕРНОЕ УСТРОЙСТВО	В ОПЕРАЦИЯХ
		GO TO,		ВВОДА-ВЫВОДА)	С ВНЕШ. УСТР.
		GO SUB,			
		LIST,	K	INVALID COLOUR (НЕВЕРНЫЙ ЦВЕТ)	INK,
		LLIST,			PAPER,
		PAUSE,		СПЕЦИФИЦИРОВАННОЕ ЧИСЛО ИМЕЕТ НЕВЕРНОЕ	BORDER,
		PLOT,		ЗНАЧЕНИЕ.	FLASH,
		CHR\$,			BRIGHT,
		PEEK			INVERSE,
		USR (L			OVER,
		ЧИСЛОВЫМ			А ТАКЖЕ
		АРГУМЕНТОМ)			ПОСЛЕ ОДНОЙ
					ИЗ ПЕРЕСЛАЧ
					УПР. СИМВОЛОВ
C	NONSENSE IN BASIC (ВЫРАЖЕНИЕ НЕ БЕЯСИКА)	VAL,			
	ТЕКСТ (СТРОКА) НЕ РАСПОЗНАЕТСЯ БЕЯСИКОМ КАК	VAL\$			
	ДОПУСТИМОЕ ВЫРАЖЕНИЕ.		L	BREAK INTO PROGRAM (БРЕАК ВО ВРЕМЯ ВЫПОЛНЕ-	РАЗНОЕ
				НИЯ ПРОГРАММ)	
D	BREAK-CONT REPEATS	LOAD,		НАЖАТА КЛАВИША BREAK; ЭТО ОБНАРУЖИВАЕТСЯ	
	КЛАВИША BREAK НАЖАТА ВО ВРЕМЯ ДЕЙСТВИЯ ПЕРЕ-	SAVE,		МЕЖА ДВУМЯ ОПЕРАТОРАМИ. СТРОКА И НОМЕР	
	ФЕРИОННОЙ ОПЕРАЦИИ. ДЕЙСТВИЯ CONTINUE ПОСЛЕ	VERIFY,		ОПЕРАТОРА В СТРОКЕ УКАЗЫВАЮТ НА ОПЕРАТОР,	
	ЭТОГО ОПЕРАТОРА ОБЫЧНЫЕ, ТО ЧТО КАЗАНЫ В	MERGE,		ВЪЕЗЖАЮЩИ ПЕРЕД НАЖАТИЕМ BREAK, НО	
	ОПЕРАТОРЕ. СРАВНИТЕ С СООБЩЕНИЕМ L.	LPRINT,		CONTINUE ПЕРЕХОДИТ К СЛЕДУЮЩЕМУ ОПЕРАТОРУ.	
		LLIST,			
		COPY (ТОЛЬКО	M	RANTOP NO GOOD (АДРЕС RANTOP НЕ ГОДЕН)	CLEAR,
		КОГДА КОМ-			ВОЗМОЖНО RUN:
		ПЬУТЕР ЗА-		ЧИСЛО, УКАЗАННОЕ ДЛЯ RANTOP СЛИШКОМ ВЕЛИКО	
		ПРОСИМ СВЕР-		ИЛИ СЛИШКОМ МАЛО.	
		КУ, А ВЫ ОТ-			
		ВЕТИЛИ N,	N	STATEMENT LOST (ОПЕРАТОР ОТСУТСТВУЕТ)	RETURN
		SPACE ИЛИ			NEXT
		STOP.		ПЕРЕХОД К ОПЕРАТОРУ, КОТОРОГО УЖЕ НЕТ.	CONTINUE
E	OUT OF DATA (ВНЕ ДАННЫХ)	READ	O	INVALID STREAM (ОШИБОЧНЫЙ ПОТОК ДАННЫХ)	В ОПЕРАЦИЯХ
	ПОПЫТКА ВЫЛАЗЬ READ КОГДА СПИСОК ДАННЫХ				ВВОДА-ВЫВОДА:
	В DATA КОНЧИЛСЯ.				
			P	FN WITHOUT DEF (FN БЕЗ DEF)	FN
				ОПРЕДЕЛЯЕМАЯ ПОЛЬЗОВАТЕЛЕМ ФУНКЦИЯ НЕ	
				ОПРЕДЕЛЕНА В ОПЕРАТОРЕ DEF FN.	
F	INVALID FILE NAME (НЕВЕРНОЕ ИМЯ ФАЙЛА)	SAVE			
	ОПЕРАТОР SAVE С ПУСТОЙ СТРОКОЙ ВМЕСТО ИМЕНИ				
	ИЛИ С ИМЕНЕМ ДЛИННЕЕ 10 СИМВОЛОВ.				
			Q	PARAMETER ERROR (ОШИБКА В ПАРАМЕТРЕ)	FN
G	NO ROOM FOR LINE (НЕТ МЕСТА ДЛЯ СТРОКИ)	ВВОД		ОШИБОЧНОЕ ЧИСЛО АРГУМЕНТОВ ИЛИ ОДНА ИЗ НИХ	
	НЕДОСТАТОЧНО МЕСТА В ПАМЯТИ ДЛЯ ЗАПИСИ	СТРОКИ В		НЕ ТОГО ТИПА, КАКАЯ БЫЛ ОПИСАН.	
	ОЧЕРЕДНОЙ СТРОКИ ПРОГРАММЫ.	ПРОГРАММУ			
			R	TAPE LOADING ERROR (ОШИБКА ЗАГРУЗКИ С ЛЕНТЫ)	VERIFY,
					LOAD,
				ФАЙЛ НА ЛЕНТЕ НАЙДЕН, НО НЕ СЧИТЫВАЕТСЯ.	MERGE

ПРИЛОЖЕНИЕ С

ФУНКЦИИ И ОПЕРАТОРЫ

ВСЕ ЧИСЛА В СИСТЕМЕ МОГУТ ИМЕТЬ ТОЧНОСТЬ 9 ИЛИ 10 ЗНАКОВ. НАИБОЛЬШЕЕ ЧИСЛО $10^{**}38$, А НАИМЕНЬШЕЕ ПОЛОЖИТЕЛЬНОЕ ЧИСЛО $4*10^{**}(-39)$. ЧИСЛА ИМЕЮТ ВНУТРЕННЕЕ ПРЕСТАВЛЕНИЕ КАК ЧИСЛА ПЛАВАЮЩЕЙ (ДВОИЧНОЙ) ТОЧКОЙ, С ВЫДЕЛЕНИЕМ ОДНОГО БАЙТА НА ПОКАЗАТЕЛЬ СТЕПЕНИ 'Е' (ЭКСПОНЕНТУ) В ИНТЕРВАЛЕ ОТ 1 ДО 255 И ЧЕТЫРЕХ БАЙТОВ НА МАНТИССУ 'М' В ИНТЕРВАЛЕ ОТ 0.5 ДО 1 (М НЕ РАВНО 1). ЭТО ПРЕСТАВЛЯЕТСЯ ЧИСЛОМ $M*2^{**}E(-128)$.

ПОСКОЛЬКУ $1/2 \leq M < 1$, СТАРШИЙ ЗНАЧАЩИЙ БИТ МАНТИССЫ ВСЕГДА 1. СЛЕДОВАТЕЛЬНО, МЫ МОЖЕМ ЗАМЕНЯТЬ ЕГО НА БИТ, ОБОЗНАЧАЮЩИЙ ЗНАК: 0 - ДЛЯ ПОЛОЖИТЕЛЬНОГО ЧИСЛА И 1 - ДЛЯ ОТРИЦАТЕЛЬНОГО.

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

Ф У Н К Ц И И

ИМЯ : ТИП : ДЕЙСТВИЕ (ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ)

ФУНКЦИИ: АРГУМЕНТА:

ABS : ЧИСЛО : АБСОЛЮТНОЕ ЗНАЧЕНИЕ

ACS : ЧИСЛО : АРККОСИНУС В РАДИАНАХ. ВЫДАЕТ СООБЩЕНИЕ С3
: : ОШИБКЕ А, ЕСЛИ X НЕ ЛЕЖИТ В ИНТЕРВАЛЕ
: : ОТ -1 ДО 1.

AND : ЛОГИЧЕС-

: КАЯ ОПЕ-

: РАЦИЯ.

: ПРАВИЛ

: ОПЕРАНА

: ВСЕГДА

: ЧИСЛО.

: СЛЕВА МО-

: ЖЕТ БИТЬ

: -ЧИСЛО,

: ТОГДА

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ B, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

: : $A \text{ AND } B = \begin{cases} A, & \text{ЕСЛИ } B \neq 0 \\ 0, & \text{ЕСЛИ } B = 0 \end{cases}$

1	2	3
CHG :	ЧИСЛО	СИМВОЛ, ЧЕЙ КОД ПРЕСТАВЛЯЕТ ЧИСЛО X, ОКРУГЛЕННЫМ К БЛИЖАЙШЕМУ ЦЕЛОМУ.
CODE :	СТРОКА	КОД ПЕРВОГО СИМВОЛА В СТРОКЕ X (ИЛИ 0, СИМВОЛОВ: ЕСЛИ X - ПУСТАЯ СТРОКА).
COS :	ЧИСЛО B	КОСИНУС X. РАДИАНАХ.
EXP :	ЧИСЛО	E В СТЕПЕНИ X.
FN :	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<X<FFFFH). ЗАГРУЖАЕТСЯ ПАРА РЕГИСТРОВ ВС И ВЫПОЛНЯЕТСЯ КОМАНДА АССЕМБЛЕРА IN A(S).
INKEY :	НЕТ	ЧТЕНИЕ С КЛАВИАТУРЫ. ВОЗВРАЩАЕТ СИМВОЛ, ВВЕДЕННЫЙ С КЛАВИАТУРЫ (В РЕЖИМЕ [L] ИЛИ [C]), ЕСЛИ БЫЛО ДЕЙСТВИТЕЛЬНОЕ НАЖАТИЕ КЛАВИШИ, ИЛИ ПУСТУЮ СТРОКУ В ПРОТИВНОМ СЛУЧАЕ.
INT :	ЧИСЛО	ОКРУГЛЕНИЕ К БЛИЖАЙШЕМУ МЕНЬШЕМУ ЦЕЛОМУ.
LEN :	СТРОКА	ДЛИНА СТРОКИ СИМВОЛОВ.
LN :	ЧИСЛО	НАТУРАЛЬНЫЙ ЛОГАРИФМ. ВЫДАЕТ СООБЩЕНИЕ A, ЕСЛИ X<=0.
NOT :	ЧИСЛО	0, ЕСЛИ X<>0, 1, ЕСЛИ X=0. ОПЕРАЦИЯ ИМЕЕТ ЧЕТВЕРТЫЙ ПРИОРИТЕТ.
OR :	ЛОГИЧЕСКАЯ ОПЕРАЦИЯ.	$A \text{ OR } B = \begin{cases} 1, & \text{ЕСЛИ } B \neq 0 \\ A, & \text{ЕСЛИ } B = 0 \end{cases}$
PEEK :	ЧИСЛО	ЗНАЧЕНИЕ БАЙТА В ПАМЯТИ ПО АДРЕСУ X, ОКРУГЛЕННОМУ К БЛИЖАЙШЕМУ ЦЕЛОМУ.
PI :	НЕТ	ЧИСЛО ПИ (3.14159265...).
POINT :	ДВА ЧИСЛА	1, ЕСЛИ ТОЧКА ЭКРАНА С КООРДИНАТАМИ (X,Y) ЗАКРАШЕНА. 0, ЕСЛИ ЭТА ТОЧКА ИМЕЕТ ЦВЕТ ФОНА. ВЫДАЕТ СООБЩЕНИЕ B, ЕСЛИ НЕ ВЫПОЛНЯЮТСЯ УСЛОВИЯ 0<X<255 И 0<Y<175.
	КЛЮЧЕННЫХ	В СКОБКИ.

BIN : ЭТО НЕ ОБЫЧНАЯ ФУНКЦИЯ. ЗА BIN ЗАПИСЫВАЕТСЯ ПОСЛЕДОВАТЕЛЬНОСТЬ НУЛЕЙ И ЕДИНИЦ, ПРЕСТАВЛЯЮЩАЯ СОБОЙ ДВОИЧНОЕ ПРЕСТАВЛЕНИЕ ЧИСЛА, КОТОРОЕ ЗАПИСЫВАЕТСЯ В ПАМЯТЬ.

1	2	3
RND	НЕТ	ОЧЕРЕДНОЕ ПСЕВДОСЛУЧАЙНОЕ ЧИСЛО ИЗ ПОСЛЕ-
		ДОВАТЕЛЬНОСТИ, ПОЛУЧАЕМОЕ ВОЗВЕДЕНИЕМ В 75
		СТЕПЕНЬ МОДУЛЯ ЧИСЛА 65537, ВЫЧИТАНИЕМ 1
		И ДЕЛЕНИЕМ НА 65536. ЧИСЛО ЛЕЖИТ В ИНТЕРВА-
		ЛЕ $3(Y-1)$.
SCREEN	ДВА ЧИС-	СИМВОЛ (ОБЫЧНЫЙ ИЛИ ИНВЕРСНЫЙ), КОТОРЫЙ
	ДОВЫХ АР-	ПОЯВЛЯЕТСЯ НА ЭКРАНЕ В СТРОКЕ X, ПОЗИЦИИ Y.
	ГУМЕНТА X:	ДАЕТ ПУСТУЮ СТРОКУ, ЕСЛИ СИМВОЛ НЕ ОПОЗНА-
	И Y, ЗАК-	
	ЛЮЧЕННЫХ	
	В СКОБКИ.	
SGN	ЧИСЛО	-1, ЕСЛИ $X < 0$
		0, ЕСЛИ $X = 0$
		1, ЕСЛИ $X > 0$
SIN	ЧИСЛО В	СИНУС X.
	РАДИАНАХ:	
SQR	ЧИСЛО	КОРЕНЬ КВАДРАТНЫЙ. ВЫДАЕТ СООБЩЕНИЕ A,
		ЕСЛИ $X < 0$.
STR	ЧИСЛО	СТРОКА СИМВОЛОВ, КОТОРАЯ ДОЛЖНА БЫТЬ
		ОТОВРАЖЕНА, ЕСЛИ X ВЫВОДИТСЯ.
USR	ЧИСЛО	ВЫЗЫВАЕТ ПОДПРОГРАММУ В МАШИНЫХ КОДАХ,
		НАЧАЛЬНЫЙ АДРЕС КОТОРОЙ X. ПРИ ВОЗВРАТЕ
		РЕЗУЛЬТАТОМ БУДЕТ СОДЕРЖИМОЕ РЕГИСТРОВОЙ
		ПАРИ ВС.
USR	СТРОКА	АДРЕС ГРУППЫ БАЙТОВ, ЗАДАВШИХ ОПРЕДЕЛЕННУЮ
	СИМВОЛОВ:	ПОЛЬЗОВАТЕЛЕМ СИМВОЛ ДЛЯ ЗАКРЕПЛЕНИЯ ЕГО
		ЗА X.
VAL	СТРОКА	ВЫЧИСЛЕНИЕ X КАК ЧИСЛОВОГО ВЫРАЖЕНИЯ.
	СИМВОЛОВ:	ВЫДАЕТ СООБЩЕНИЕ C, ЕСЛИ X СОДЕРЖИТ СИНТАК-
		СИЧЕСКИЕ ОШИБКИ ИЛИ ДАЕТ СТРОКОВОЕ
		ЗНАЧЕНИЕ. ВОЗМОЖНЫ И ДРУГИЕ ОШИБКИ.
VAL	СТРОКА	ВЫЧИСЛЯЕТ X КАК СТРОКОВОЕ ВЫРАЖЕНИЕ. ВЫДАЕТ:
	СИМВОЛОВ:	СООБЩЕНИЕ C, ЕСЛИ X СОДЕРЖИТ СИНТАКСИЧЕСКУЮ
		ОШИБКУ ИЛИ ДАЕТ НЕ СТРОКОВОЕ (ЧИСЛОВОЕ)
		ЗНАЧЕНИЕ.

О П Е Р А Ц И И

ПРЕФИКСНЫЕ

- ЧИСЛО ОТРИЦАТЕЛЬНОЕ ЗНАЧЕНИЕ

ИНФИКСНЫЕ

+ СЛОЖЕНИЕ ДЛЯ ЧИСЕЛ, КОНКАТЕНЦИЯ ДЛЯ СТРОК

- ВЫЧИТАНИЕ

* УМНОЖЕНИЕ

/ ДЕЛЕНИЕ

** ВОЗВЕДЕНИЕ В СТЕПЕНЬ (СТРЕЛКА ВВЕРХ). СООБЩЕНИЕ B, ЕСЛИ ЛЕВЫЙ ОПЕРАНД ОТРИЦАТЕЛЬНЫЙ.

= РАВЕНСТВО

> БОЛЬШЕ

< МЕНЬШЕ

>= БОЛЬШЕ ИЛИ РАВНО

<= МЕНЬШЕ ИЛИ РАВНО

<> НЕ РАВНО

ОБА ОПЕРАНДА ДОЛЖНЫ БЫТЬ ОДНОГО
ТИПА. РЕЗУЛЬТАТ РАВЕН 1, ЕСЛИ
СРАВНЕНИЕ КОТИНО И РАВНО 0,
ЕСЛИ НЕТ.

ФУНКЦИИ И ОПЕРАЦИИ ИМЕЮТ СЛЕДУЮЩИЕ ПРИОРИТЕТЫ:

ИНДЕКСАЦИЯ И СЕЧЕНИЕ	- 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

О П Е Р А Т О Р Ы

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
АПОСТОФИЧЕСКИМ ':'C - ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ УПРАВЛЕНИЯ ЦВЕТОМ.
КАЖДАЯ ЗАКАНЧИВАЕТСЯ ЗАПЯТОЙ ИЛИ ТОЧКОЙ С
ЗАПЯТОЙ. ЦВЕТОВЫЙ СИМВОЛ ИМЕЕТ ФОРМУ ОПЕРАНДОВ:
PAPER, INK, FLASH, BRIGHT, INVERSE, OVER.ТЕКСТ ПРОИЗВОЛЬНОГО ВЫРАЖЕНИЯ МОЖЕТ РАСПОЛАГАТЬСЯ В ЛЮБОМ
МЕСТЕ СТРОКИ (ЗА ИСКЛЮЧЕНИЕМ НОМЕРА СТРОКИ, КОТОРЫЙ ДОЛЖЕН
РАЗМЕЩАТЬСЯ В НАЧАЛЕ СТРОКИ).ВСЕ ОПЕРАТОРЫ, КРОМЕ INPUT, DEF И DATA МОГУТ
ИСПОЛЬЗОВАТЬСЯ И КАК КОМАНДЫ И В ПРОГРАММАХ.КОМАНДА ИЛИ СТРОКА ПРОГРАММЫ МОЖЕТ СОДЕРЖАТЬ НЕСКОЛЬКО
ОПЕРАТОРОВ, РАЗДЕЛЕННЫХ АПОСТОФИЧЕСКИМ ':'НЕТ ОГРАНИЧЕНИЯ НА ПОЛОЖЕНИЕ ОПЕРАТОРА В СТРОКЕ, ХОТЯ ЕСТЬ
НЕКОТОРЫЕ ОГРАНИЧЕНИЯ В IF И REM.

ВСЕ ОПЕРАТОРЫ ЯЗЫКА СВЕДЕНЫ В СЛЕДУЮЩУЮ ТАБЛИЦУ:

О П Е Р А Т О Р	Д Е Й С Т В И Е О П Е Р А Т О Р А
1	2
BEER X, Y	ВОСПРОИЗВОДИТ ЗВУК ДЛИТЕЛЬНОСТЬЮ
	X СЕК. И ВЫСОТОЙ Y ПОЛУТОНОВ ВВЕРХ
	ОТ ОСНОВНОГО ТОНА ДО (ИЛИ ВНИЗ, ЕСЛИ
	Y ОТРИЦАТЕЛЬНОЕ).
BORDER M	УСТАНОВЛИВАЕТ ЦВЕТ РАМКИ (БОРДЮРА)
	ЭКРАНА. ВЫДАЕТ СООБЩЕНИЕ ОБ ОШИБКЕ
	K, ЕСЛИ 0) И Y.

1	2	1	2
BRIGHT N	УСТАНАВЛИВАЕТ ЯРКОСТЬ ВЫВОДИМОГО СИМВОЛА: 0 - для обычной яркости; 1 - для повышенной яркости; 8 - сохраняет существующую яркость.	DIM A(N1,N2,...,NK)	УНИЧТОЖАЕТ МАССИВ С ИМЕНЕМ 'A' И УСТАНАВЛИВАЕТ ЧИСЛОВОЙ МАССИВ 'A' С 'K' ИЗМЕРЕНИЯМИ И ПРИСВАИВАЕТ ВСЕМ ЕГО ЭЛЕМЕНТАМ ЗНАЧЕНИЕ 0.
CAT	БЕЗ MICRODRIVE НЕ РАБОТАЕТ.	DIM A\$(N1,N2,...,NK)	УНИЧТОЖАЕТ МАССИВ ИЛИ СТРОКУ С ИМЕНЕМ 'A\$' И УСТАНАВЛИВАЕТ СИМВОЛЬНЫЙ МАССИВ С 'K' ИЗМЕРЕНИЯМИ И ПРИСВАИВАЕТ ВСЕМ ЕГО ЭЛЕМЕНТАМ ЗНАЧЕНИЕ " ". МАССИВ МОЖЕТ 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 ЕГО ОПИСАНИЯ В ОПЕРАТОРЕ DIM.
CIRCLE X,Y,Z	ИЗОБРАЖАЕТ ДУГУ ИЛИ ОКРУЖНОСТЬ С ЦЕНТРОМ В ТОЧКЕ С КООРДИНАТАМИ (X,Y) И РАДИУСОМ Z.	DRAW X,Y	ТО ЖЕ САМОЕ, ЧТО И DRAW X,Y,0. ЧЕРТИТ ПРЯМУЮ ЛИНИЮ.
CLEAR	УНИЧТОЖАЕТ ВСЕ ПЕРЕМЕННЫЕ И ОЧИЩАЕТ ЗАНИМАЕМУЮ ИЛИ ПАМЯТЬ. ВЫПОЛНЯЕТ RESTORE И CLS, УСТАНАВЛИВАЕТ PLOT ПОЗИЦИЮ В НИЖНЮЮ ЛЕВУЮ ТОЧКУ ЭКРАНА И ОЧИЩАЕТ GO SUB СТЕК.	DRAW X,Y,Z	ИЗОБРАЖАЕТ ЛИНИЮ ОТ ТЕКУЩЕЙ ГРАФИЧЕСКОЙ ПОЗИЦИИ В ТОЧКУ С ПРИРАЩЕНИЯМИ X,Y ПО ДУГЕ В Z РАДИАН. ВЫДАЕТ СООБЩЕНИЕ В ПРИ ВЫХОДЕ ЗА ПРЕДЕЛЫ ЭКРАНА.
CLEAR N	ПОДОБНО CLEAR, НО ДОПОЛНИТЕЛЬНО ИЗМЕНЯЕТ СИСТЕМНУЮ ПЕРЕМЕННУЮ RAMTOP НА 'N' И ЗАДАЕТ НОВЫЙ GO SUB СТЕК.	ERASE	БЕЗ MICRODRIVE НЕ РАБОТАЕТ.
CLOSE	БЕЗ MICRODRIVE НЕ РАБОТАЕТ.	FLASH N	ОПРЕДЕЛЯЕТ, БУДЕТ ЛИ СИМВОЛ МЕРЦАЮЩИМ ИЛИ С ПОСТОЯННЫМ СВЕЩЕНИЕМ. N=0 для ПОСТОЯННОГО СВЕЩЕНИЯ, N=1 - для МЕРЦАНИЯ, N=8 - для СОХРАНЕНИЯ ПРЕДЫДУЩЕГО СОСТОЯНИЯ.
CLS	(CLEAR SCREEN) ОЧИЩАЕТ ФАЙЛ ЭКРАНА.	FOR A=X TO Y	FOR A=X TO Y STEP 1
CONTINUE	ПРОДОЛЖАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ, НАЧАТОЙ РАНЕЕ И ОСТАНОВЛЕННОЙ С СООБЩЕНИЕМ, ОТЛИЧНЫМ ОТ 0. ЕСЛИ ВЫЛО 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 ПЕРЕДАЕТ 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.	FOR A=X TO Y STEP Z	УНИЧТОЖАЕТ СКАЛЯРНУЮ ПЕРЕМЕННУЮ 'A' И УСТАНАВЛИВАЕТ УПРАВЛЯЮЩУЮ ПЕРЕМЕННУЮ 'X', ПРЕДЕЛ 'Y', шаг ПРИРОЩЕНИЯ '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 НЕХТА ИЛИ ВЫДАЧА СООБЩЕНИЯ 1, ЕСЛИ НЕТ (CM.NEXT). СООБЩЕНИЕ 4 ВЫДАЕТСЯ, ЕСЛИ НЕДОСТАТОЧНО МЕСТА ДЛЯ РАЗМЕЩЕНИЯ УПРАВЛЯЮЩЕЙ ПЕРЕМЕННОЙ.
COPY	ПЕРЕСЫЛАЕТ КОПИЮ 22 СТРОК ЭКРАНА НА ПРИНТЕР, ЕСЛИ ОН ПОДКЛЮЧЕН. ПОМНИТЕ, ЧТО ПО COPY НЕЛЬЗЯ РАСПЕЧАТАТЬ НАХОДЯЩИЕСЯ НА ЭКРАНЕ АВТОМАТИЧЕСКИЙ ЛИСТИНГ. ВЫДАЕТ СООБЩЕНИЕ D, ЕСЛИ НАЖАТЬ КЛАВИШУ BREAK.	FORMAT F	БЕЗ MICRODRIVE НЕ РАБОТАЕТ.
DATA E1,E2,E3,...	ЧАСТЬ СПИСКА ДАННЫХ. ДОЛЖНА РАСПОЛАГАТЬСЯ В ПРОГРАММЕ.	GO SUB N	ПРОДОЛЖИВАЕТ СТРОКУ С ОПЕРАТОРОМ GO SUB В СТЕК ДЛЯ ИСПОЛЬЗОВАНИЯ ЗАТЕМ КАК GO TO N. ВЫДАЕТСЯ СООБЩЕНИЕ 4, ЕСЛИ НЕ ВСЕ ПОДПРОГРАММЫ ЗАВЕРШИЛИСЬ С RETURN.
DEF FN A(A1,...,AK)=E	ОПРЕДЕЛЯЕМАЯ ПОЛЬЗОВАТЕЛЕМ ФУНКЦИЯ. ДОЛЖНА РАСПОЛАГАТЬСЯ В ПРОГРАММЕ. A,A1,...,2 И Т.Д. - БУКВЫ ИЛИ БУКВЫ С \$ для СТРОКОВЫХ АРГУМЕНТОВ. ИСПОЛЬЗУЕТСЯ ФОРМА DEF FN A(), ЕСЛИ НЕТ АРГУМЕНТОВ.	GO TO N	ПРОДОЛЖАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ СО СТРОКИ 'N'. ЕСЛИ 'N' ОПУЩЕНО, ТО С ПЕРВОЙ СТРОКИ ПОСЛЕ ЭТОЙ.
DELETE F	БЕЗ MICRODRIVE НЕ РАБОТАЕТ.		

1	2	1	2
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.	LET V=E	: ПРИСВАИВАЕТ ЗНАЧЕНИЕ 'E' ПЕРЕМЕННОЙ : 'V'. КЛЮЧЕВОЕ СЛОВО LET НЕ МОЖЕТ БЫТЬ : ОПУЩЕНО. СКАЛЯРНАЯ ПЕРЕМЕННАЯ НЕ ОБ- : РЕДЕЛЕНА, ТОКА НЕ ВСТРЕТИТЬСЯ В ОПЕ- : РАТОРАХ LET, READ ИЛИ INPUT. ЕСЛИ 'V' : ИНДЕКСИРУЕМАЯ СТРОКОВАЯ ПЕРЕМЕННАЯ : ИЛИ СЕЧЕНИЕ СТРОКОВОГО МАССИВА (ПОД- : СТРОКА), ТО ПРИСВАИВАНИЕ ОСУЩЕСТВЛЯ- : ЕТСЯ С УСЕЧЕНИЕМ СПРАВА ИЛИ ДОПОЛ- : НЕНИЕМ ПРОБЕЛАМИ ДО ФИКСИРОВАННОЙ : ДЛИНЫ.
INK N	: УСТАНАВЛИВАЕТ ЦВЕТ ЗАКРАШИВАНИЯ (Т.Е. : ЦВЕТ, КОТОРЫМ БУДУТ ИЗОБРАЖАТЬСЯ : СИМВОЛЫ НА ЦВЕТЕ ФОНА). 'N' В ИНТЕР- : ВАЛЕ ОТ 0 ДО 7 УКАЗЫВАЕТ ЦВЕТ. N=8 : - ОСТАВИТЬ ЦВЕТ БЕЗ ИЗМЕНЕНИЯ, N=9 : - УВЕЛИЧЕНИЕ КОНТРАСТА. ВЫДАЕТ СООБ- : ЩЕНИЕ K, ЕСЛИ 'N' НЕ ЛЕЖИТ В ИНТЕР- : ВАЛЕ ОТ 0 ДО 9.	LIST	: ТО ТЕ, ЧТО И LIST 0.
INPUT ...	: ГДЕ '...' ЕСТЬ ПОСЛЕДОВАТЕЛЬНОСТЬ : ВВОДИМЫХ СИМВОЛОВ, РАЗДЕЛЯЕМЫХ КАК : В ОПЕРАТОРЕ PRINT ЗАПЯТЫМИ, ТОЧКАМИ : С ЗАПЯТОЙ ИЛИ АПОСТРОФАМИ. ВВОДИМЫЕ : СИМВОЛАМИ МОГУТ БЫТЬ: : А) НЕКОТОРЫЕ PRINT-СИМВОЛЫ, НАЧИНА- : ЮЩИЕСЯ НЕ С БУКВЫ; : Б) ИМЯ ПЕРЕМЕННОЙ; : В) СТРОКА ИЛИ ПЕРЕМЕННЫХ СТРОКОВО- : ГО ТИПА. : PRINT-СИМВОЛЫ В СЛУЧАЕ А) ПРЕСТАВЛЯ- : ЮТСЯ ТАК ЖЕ, КАК И В ОПЕРАТОРЕ PRINT, : ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО ОНИ ВСЕ ВЫ- : ВОДЯТСЯ В НИЖНЮЮ ЧАСТЬ ЭКРАНА. В : СЛУЧАЕ Б) КОМПЬЮТЕР ОСТАНАВЛИВАЕТСЯ : И ЖДЕТ ВВОДА НЕКОТОРОГО ВЫРАЖЕНИЯ С : КЛАВИАТУРЫ, ЗНАЧЕНИЕ КОТОРОГО БУДЕТ : ПРИСВОЕНО ПЕРЕМЕННОЙ. ВВОД ОСУЩЕСТ- : ВЛЯЕТСЯ ОБЫЧНЫМ ОБРАЗОМ, А СИНТАКСИ- : ЧЕСКИЕ ОШИБКИ ВЫДАЮТСЯ ПЕРЦАЩИМИ (?). : ДЛЯ СТРОКОВОГО ВЫРАЖЕНИЯ ВВОДНОЙ БУ- : ФЕР УСТАНАВЛИВАЕТСЯ ДЛЯ РАЗЪЕДИНЕНИЯ : ДВУХ ТАКИХ СТРОК (КОТОРЫЕ ПРИ НЕОБ- : ХОДИМОСТИ МОЖЕТ БЫТЬ УВЕЛИЧЕН). ЕСЛИ : ПЕРВЫЙ ВВОДИМЫЙ СИМВОЛ STOP, ТО ПРОГ- : РАМКА ОСТАНАВЛИВАЕТСЯ С СООБЩЕНИЕМ N. : СЛУЧАЕ В) ПОДОБЕН СЛУЧАЮ Б) С ТОЖ : ДЛИННОЙ РАЗНИЦЕЙ, ЧТО ВВОДИМАЯ ИНФОРМА- : ЦИЯ ПРЕСТАВЛЯЕТ СОБОЙ СТРОКОВЫЙ ЛИ- : ТЕРАЛ НЕОГРАНИЧЕННОЙ ДЛИНЫ, И STOP В : ЭТОМ СЛУЧАЕ НЕ СРАБОТАЕТ. ДЛЯ ОСТАНО- : ВА ВЫ ДОЛЖНЫ НАЖАТЬ ЛАВНУЮ 'КУРСОР : ВНИЗ'.	LIST N	: ЗАПИСЫВАЕТ ТЕКСТ ПРОГРАММЫ В ВЕРХНЮЮ : ЧАСТЬ ЭКРАНА, НАЧИНАЯ С ПЕРВОЙ СТРО- : КИ, МЕНЬШЕЙ, ЧЕМ 'N', И ДЕЛАЕТ 'N' : ТЕКУЩЕЙ СТРОКОЙ.
		LLIST	: ТО ТЕ, ЧТО И LIST 0.
		LLIST N	: ПОДОБНО LIST, НО ВЫВОД ОСУЩЕСТВЛЯЕТСЯ : НА ПРИНТЕР.
		LOAD F	: ЗАГРУЖАЕТ ПРОГРАММУ И ПЕРЕМЕННЫЕ.
		LOAD F DATA ()	: ЗАГРУЖАЕТ ЧИСЛОВЫЙ МАССИВ.
		LOAD F DATA\$()	: ЗАГРУЖАЕТ СТРОКОВЫЙ МАССИВ.
		LOAD F CODE M,N	: ЗАГРУЖАЕТ СТАРШИЕ 'N' БАЙТОВ, НАЧИ- : НАЯ 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 ЕГО : 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 КО- : ТОРЫХ СОВПАДАЮТ НОМЕРА ИЛИ ИМЕНА С : ТАКИМИ ЖЕ НА ЛЕНТЕ.
		MOVE F1,F2	: БЕЗ MICRODRIVE НЕ РАБОТАЕТ.
INVERSE N	: СИМВОЛ УПРАВЛЕНИЯ ИНВЕРСИЕЙ ВЫВОДИМО- : ГО СИМВОЛА. ЕСЛИ N=0, СИМВОЛ ВЫВОДИТ- : СЯ В ОБЫЧНОМ ВИДЕ С ПРОРИСОВКОЙ ЦВЕ- : ТОМ ЗАКРАШИВАНИЯ (INK) НА ЦВЕТЕ ФОНА : (PAFER). ЕСЛИ N=1, ТО ЦВЕТОВОЕ РЕШЕ- : НИЕ ИЗОБРАЖЕНИЯ СИМВОЛА МЕНЯЕТСЯ НА : ОБРАТНОЕ. СМОТРИ ПРИЛОЖЕНИЕ В. ВЫДАЕТ : СООБЩЕНИЕ K, ЕСЛИ 'N' НЕ 0 ИЛИ 1.	NEW	: ЗАПУСКАЕТ ПО НОВОЙ СИСТЕМУ ПРОГРАМ- : МИРОВАНИЯ ВЕРСИИ, УНИЧТОЖАЯ СТАРУЮ : ПРОГРАММУ И ПЕРЕМЕННЫЕ И ИСПОЛЬЗУЕМУЮ : ПАМЯТЬ, ВКЛЮЧАЯ И БАЙТ АДРЕСА В СИС- : ТЕМНОЙ ПЕРЕМЕННОЙ RANDO, НО СОХРА- : НЯЕТ СИСТЕМНЫЕ ПЕРЕМЕННЫЕ UDG, P RAND : RASP И RIR.

1	2	1	3
NEXT A	: A) НАХОДИТ УПРАВЛЯЕМУЮ ПЕРЕМЕННУЮ 'A'; : Б) ПРИДАВАЕТ 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 ПРОИХО- : ДИТ ПЕРЕХОД К ОПЕРАТОРУ ЦИКЛА. : СООБЩЕНИЕ 2 ВЫДАЕТСЯ, ЕСЛИ НЕ НАЙДЕНА : ПЕРЕМЕННАЯ 'A'. СООБЩЕНИЕ 1 ВЫДАЕТСЯ, : ЕСЛИ 'A' НЕ ЯВЛЯЕТСЯ УПРАВЛЯЮЩЕЙ : ПЕРЕМЕННОЙ ЦИКЛА.	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.
OPEN	: БЕЗ MICRODRIVE НЕ РАБОТАЕТ.	PRINT ...	: ГДЕ '...' ПОСЛЕДОВАТЕЛЬНОСТЬ PRINT- : СИМВОЛОВ, РАЗДЕЛЕННЫХ ЗАПЯТЫМИ, ТОЧ- : КОЙ С ЗАПЯТОЙ ИЛИ АПОСТРОФАМИ, КОТО- : РЫЕ ВЫВОДЯТСЯ В ЭКРАНИИ ФАЙЛ ДЛЯ : ОТОБРАЖЕНИЯ НА ЭКРАНЕ ТЕЛЕВИЗОРА. : ТОЧКА С ЗАПЯТОЙ САМА ДЕЙСТВИЯ НЕ ВЫ- : ЗЫВАЕТ, А ИСПОЛЬЗУЕТСЯ ДЛЯ РАЗГРАНИ- : ЧЕНИЯ СИМВОЛОВ. ЗАПЯТАЯ ПОРОЖДАЕТ : УПРАВЛЯЮЩИЙ СИМВОЛ 'ЗАПЯТАЯ' И АПОСТ- : РОФ ПОРОЖДАЕТ СИМВОЛ ENTER. В КОНЦЕ : ОПЕРАТОРА PRINT, ЕСЛИ ОН НЕ ЗАКАНЧИ- : ВАЕТСЯ ТОЧКОЙ С ЗАПЯТОЙ, ЗАПЯТОЙ ИЛИ : АПОСТРОФОМ, АВТОМАТИЧЕСКИ ВЫВОДИТСЯ : СИМВОЛ ENTER. PRINT-СИМВОЛ МОЖЕТ БЫТЬ: : А) ПУСТАЯ СТРОКА (Т.Е. НИЧЕГО); : Б) ЧИСЛОВОЕ ВЫРАЖЕНИЕ. : ЕСЛИ ЗНАЧЕНИЕ ВЫРАЖЕНИЯ ОТРИЦАТЕЛЬНО, : ТО ВЫВОДИТСЯ СИМВОЛ МИНУС. ЕСЛИ : $X \leq 10^{**}(-5)$ ИЛИ $X = 10^{**}13$, ВЫВОД : ОСУЩЕСТВЛЯЕТСЯ В ПОКАЗАТЕЛЬНОЙ ФОРМЕ. : МАНТИССА ПРЕДСТАВЛЯЕТСЯ 8-МЬЮ ЦИФРАМИ : (С НОРМАЛИЗАЦИЕЙ) И ДЕСЯТИЧНОЙ ТОЧКОЙ : (ОТСУТСТВУЕТ ТОЛЬКО ТОГДА, КОГДА В : МАНТИССЕ ОДНА ЦИФРА) ПОСЛЕ ПЕРВОЙ : ЦИФРЫ. ПОКАЗАТЕЛЬ СТЕПЕНИ ЗАПИСЫВАЕТ- : СЯ ПОСЛЕ БУКВЫ 'E' С ПОСЛЕДУЮЩИМ ЗНА- : КОМ 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 ВЫЗЫ- : ВАЮТ ОПРЕДЕЛЕННОЕ ИЛИ ДЕЙСТВИЕ. НЕ : ОТРАЖАЕМЫЕ НА ЭКРАНЕ СИМВОЛЫ ВЫВО- : ДЯТСЯ КАК '?'. : Г) AT M,N : ВЫВОД В СТРОКУ 'M', ПОЗИЦИЮ 'N'. : А) TAB N : ВЫВОД УПРАВЛЯЮЩЕГО СИМВОЛА TAB С : ПОСЛЕДУЮЩИМИ ДВУМЯ БАЙТАМИ 'N' (ПЕР- : ВЫЙ БАЙТ СТАРШИЙ). ВЫЗЫВАЕТ TAB-ОСТА- : НОВ : Е) ЦВЕТОВОЙ СИМВОЛ В ФОРМЕ PAPER, : INK, FLASH, BRIGHT, INVERSE ИЛИ OVER : ОПЕРАТОРА.
OVER N	: УПРАВЛЯЮЩИЙ СИМВОЛ НАДПЕЧАТЫВАНИЯ ПО : ВЫВЕДЕННОЙ СТРОКЕ. ЕСЛИ N=0, ТО ВЫВО- : ДИМЫЙ 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 СТАРОГО, ИЛИ ЦВЕТ ФОНА, : ЕСЛИ ОБА УКАЗЫВАЮТ НА ОДИН И ТОТ ЖЕ : ЦВЕТ (ЛИБО ФОНА, ЛИБО ЗАКРАШИВАНИЯ. : СЛОЖЕНИЕ ПО МОДУЛЮ 2). СМОТРИ ПРИЛО- : ЖЕНИЕ В.		: ТО ЖЕ, ЧТО И RANDOMIZE 0.
PAPER N	: ПОДОВЕИ INK, НО УПРАВЛЯЕТ ЦВЕТОМ ФОНА.		
PAUSE N	: ОСТАНАВЛИВАЕТ ВЫПОЛНЕНИЕ ПРОГРАММЫ И : ЗАДЕРЖИВАЕТ ИЗОБРАЖЕНИЕ НА ЭКРАНЕ НА : 'N' КАДРОВ (50 КАДРОВ В СЕК. - ЧИС- : ЛО КАДРОВЫХ РАЗВЕРТКИ) ИЛИ ДО НА- : ЖАТИЯ ЛЮБОЙ КЛАВИШИ. $0 \leq N \leq 65535$, : ИНАЧЕ ВЫДАЕТСЯ СООБЩЕНИЕ В. ПРИ N=0 : ВРЕМЯ ЗАДЕРЖКИ НЕ УЧИТЫВАЕТСЯ И ПРО- : ДОЛЖАЕТСЯ ДО ПЕРВОГО НАЖАТИЯ КЛАВИШИ.	RANDOMIZE	
PLOT C:M,N	: ВЫВОДИТ ТОЧКУ ЗАКРАШИВАЮЩЕГО ЦВЕТА : (ОБРАБОТАННАЯ OVER И INVERSE) С КО- : ООРДИНАТАМИ (ABS(M), ABS(N)) СМЕШАЕТ : ГРАФИЧЕСКУЮ (PLOT POSITION) ПОЗИЦИЮ. : ЕСЛИ ЦВЕТОВОЙ СИМВОЛ 'C' НЕ СПЕЦИФИ- : ЦИРОВАН ИНАЧЕ, ТО ЦВЕТ ЗАКРАШИВАНИЯ : В ПОЗИЦИИ, ГДЕ РАСПОЛОЖЕНА ЭТА ТОЧКА : ИЗМЕНЯЕТСЯ НА ТЕКУЩИЙ СПЛОШНОЙ ЗА- : КРАШИВАЮЩИЙ ЦВЕТ, И ДРУГИЕ УКАЗАНИЯ : (ЦВЕТ ФОНА, МЕРЦАНИЕ, ЯРКОСТЬ) ОСТА- : ЮТСЯ БЕЗ ИЗМЕНЕНИЯ. : $A \leq ABS(M) \leq 255$: $0 \leq ABS(N) \leq 175$, : ИНАЧЕ СООБЩЕНИЕ В.	RANDOMIZE N	: УСТАНОВЛИВАЕТ СИСТЕМНУЮ ПЕРЕМЕННУЮ : SEED, ИСПОЛЬЗУЕМУЮ ДЛЯ ВЫЧИСЛЕНИЯ : ОЧЕРЕДНОГО ЗНАЧЕНИЯ ФУНКЦИИ RND. ЕСЛИ : N < 0, ТО SEED ПРИНИМАЕТ ЗНАЧЕНИЕ : 'N'. ЕСЛИ N=0, ТО SEED ПРИНИМАЕТ : ЗНАЧЕНИЕ ДРУГОЙ СИСТЕМНОЙ ПЕРЕМЕННОЙ : FRAMES, ПОДСЧИТЫВАЮЩЕЙ КАДРЫ, ОТОБ-

1	2	1	2
	РАЖЕНИЕ НА ЭКРАНЕ, ЧТО ОБЕСПЕЧИВАЕТ ВПОЛНЕ СЛУЧАЙНОЕ ЧИСЛО. ОПЕРАТОР ДО- ПУСКАЕТ СОКРАЩЕНИЕ RAND (СН.КЛАВИШУ). ВЫДАЕТ СООБЩЕНИЕ В, ЕСЛИ 'N' НЕ ЛЕЖИТ В ИНТЕРВАЛЕ ОТ 0 ДО 65535.	VERIFY TO IF. "TO LOAD" ЗА ИСКЛЮЧЕНИЕМ ТОГО, ЧТО ДАННЫЕ ЗАГРУЖАЮТСЯ В ОЗУ, КО СРАВНИВАЮТСЯ С НАХОДЯЩИМИСЯ ТАМ. ВЫДАЕТ СООБЩЕНИЕ В, ЕСЛИ ОБНАРУЖЕН ХОТЯ БЫ ОДИН НЕ СОВПАДАЮЩИЙ БАЙТ.	
READ V1,V2,...,VK	ПРИСВАИВАЕТ ПЕРЕМЕННЫМ ОДНОЯ ЗА ДРУ- ГОЯ ЗНАЧЕНИЯ ПОСЛЕДОВАТЕЛЬНО ПРЕД- СТАВЛЕННЫЕ В СПИСКЕ DATA.	П Р И Л О Ж Е Н И Е D	ПРИМЕРЫ ПРОГРАММ
REM ...	НЕ ВЫПОЛНЯЕТСЯ. '...' МОЖЕТ БЫТЬ ПОСЛЕДОВАТЕЛЬНОСТЬ СИМВОЛОВ (ИСКЛЮ- ЧАЯ ENTER). МОЖЕТ ВКЛЮЧАТЬ ДВОЕТОЧИЕ ДЛЯ УКАЗАНИЯ ОТСУТСТВИЯ ОПЕРАТОРОВ В СТРОКЕ С REM.	ЭТО ПРИЛОЖЕНИЕ СОДЕРЖИТ НЕКОТОРЫЕ ПРИМЕРЫ ПРОГРАММ, ДЕМОНСТРИРУЮЩИЕ ВОЗМОЖНОСТИ ZX SPECTRUM. ПЕРВАЯ ИЗ ЭТИХ ПРОГРАММ ТРЕБУЕТ ВВЕСТИ ДАТУ И ДАЕТ ДЕНЬ НЕДЕЛИ, КОТОРЫЙ СООТВЕТСТВУЕТ ЭТОЙ ДАТЕ.	
RESTORE	ТО ЖЕ САМОЕ, ЧТО И RESTORE 0.	10 REM CONVERT 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 "MONTH?":MONTH 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 100 160 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)=29 190 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 CENTURY 310 LET Y=YEAR-1901 320 LET B=365*Y+INT(Y/4): REM NUMBER OF DAYS 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 CONVERT 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,TO M) 470 PRINT "IS A " : E\$: "DAY" 500 LET M(2)=20: 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" 1030 DATA 31,28,31,30,31,30,31,31,30,31,30,31: REM DAYS IN MONTH	
RESTORE N	ПЕРЕЗАПИСЫВАЕТ УКАЗАТЕЛЬ ДАННЫХ В ПЕРВЫЙ ОПЕРАТОР DATA В СТРОКЕ МЕНЬ- ШЕЙ, ЧЕМ 'N'. СЛЕДУЮЩИЙ ОПЕРАТОР READ НАЧНЕТ СЧИТЫВАНИЕ ОТСДА.		
RETURN	ССЫЛАЕТСЯ НА ОПЕРАТОР GO SUB В СТЕКЕ И ПЕРЕДАЕТ УПРАВЛЕНИЕ НА СТРОКУ ПОСЛЕ НЕГО. ВЫДАЕТ СООБЩЕНИЕ 7, ЕСЛИ НЕТ УКАЗЫВАЕМОГО ОПЕРАТОРА В СТЕКЕ. ХА- РАКТЕРНАЯ ОШИБКА, КОГДА ОПЕРАТОРЫ GO SUB НЕ СБАЛАНСИРОВАНЫ ОПЕРАТОРАМИ RETURN.		
RUN	ТО ЖЕ САМОЕ, ЧТО И RUN 0.		
RUN N	CLEAR И ЗАТЛ. GO TO N.		
SAVE F	ЗАПИСЫВАЕТ НА ЛЕНТУ ПРОГРАММУ И ПЕРЕМЕННЫЕ.		
SAVE F LINE N	ЗАПИСЫВАЕТ НА ЛЕНТУ ПРОГРАММУ И ПЕРЕ- МЕННЫЕ ТАКИМ ОБРАЗОМ, ЧТО ПРИ ЗАГРУЗ- КЕ ПРОГРАММА АВТОМАТИЧЕСКИ ВЫПОЛНЯЕТ- СЯ СО СТРОКИ 'N'.		
SAVE F DATA ()	ЗАПИСЬ НА ЛЕНТУ ЧИСЛОВОГО МАССИВА.		
SAVE F DATA\$ ()	ЗАПИСЬ НА ЛЕНТУ СТРОКОВОГО МАССИВА.		
SAVE F CODE M,N	ЗАПИСЫВАЕТ НА ЛЕНТУ 'N' БАЙТОВ, НАЧИНАЯ 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 ПРОГРАММЫ С ВЫДАЧЕЙ СООБЩЕНИЯ 9. CONTINUE (ПРО- ДОЛЖЕНИЕ) БУДЕТ ОСУЩЕСТВЛЯТЬСЯ СО СЛЕДУЮЩЕГО ОПЕРАТОРА.		

ЭТА ПРОГРАММА УСТАНАВЛИВАЕТ СООТВЕТСТВИЕ МЕЖДУ ЯРДАМ,
ФУТМ И ДЮЙМАМИ:

```

10 INPUT "YARDS?",YD,"FEET?",FT,"INCHES?",IN
40 GO SUB 1000: 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=36*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 OK
1080 LET YD=INT(FT/3)*S: LET FT=FT-S*3*YD: RETURN
2000 REM SUBROUTINE TO PRINT YD,FT AND IN
2010 PRINT YD;"YD";FT;"FT";IN;"IN": RETURN

```

СЛЕДУЮЩАЯ ПРОГРАММА - ИГРА "ЯЩЕРЫ". 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 ПОПРОСИТ ВАС ЗАДАТЬ ЕМУ НАДОБАЮЩИЕ ВОПРОСЫ,
КОТОРЫЕ ПОМОГУТ ЕМУ НАЙТИ ПРАВИЛЬНЫЙ ОТВЕТ ИЛИ ОН ПОПРОСИТ ВАС
ПРЕДЛОЖИТЬ ЕМУ ЗАДАТЬ НАЗВАНИЕ НОВОГО ЖИВОТНОГО:

```

5 REM PANGOLINS
10 LET NQ=100: REM NUMBER OF QUESTIONS AND ANIMALS
15 DIM Q$(NQ,50): DIM A(NQ,2): DIM R$(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 START PLAYING
110 PRINT "THINK OF AN ANIMAL.", "PRESS ANY KEY TO
    CONTINUE."
120 PAUSE 0
130 LET C=1: REM START WITH 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 210
210 LET C=A(C,IN): GO TO 140
300 REM ANIMAL
310 PRINT "ARE YOU THINKING OF"
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$="N" THEN GO TO 500
370 IF R$="n" THEN GO TO 500
380 PRINT "ANSWER ME PROPERLY WHEN I'M", "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 QF > 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$(QF)=Q$(C): REM MOVE OLD ANIMAL
530 PRINT "WHAT IS IT, THEN?": INPUT Q$(QF+1)
540 PRINT "TELL ME A QUESTION WHICH DIST-", "INGUISHES
    BETWEEN"
550 LET P$=Q$(QF): GO SUB 900: PRINT "AND"
560 LET P$=Q$(QF+1): GO SUB 900: PRINT " "
570 INPUT S$: 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$(QF+1): GO SUB 900: PRINT "?"
620 GO SUB 1000
630 LET IN=1: LET ID=2: REM ANSWERS FOR NEW AND OLD
    ANIMALS
640 IF R$="Y" THEN GO TO 700
650 IF R$="y" THEN GO TO 700
660 LET IN=2: LET ID=1
670 IF R$="N" THEN GO TO 700
680 IF R$="n" THEN GO TO 700
690 PRINT "THAT'S NO GOOD.": GO TO 600
700 REM UPDATE ANSWERS
710 LET A(C,IN)=QF+1: LET A(C,ID)=QF
720 LET QF=QF+2: REM NEXT FREE ANIMAL SPACE
730 PRINT "THAT FOOLED ME."
800 REM AGAIN?
810 PRINT "DO YOU WANT ANOTHER GO?": GO SUB 1000
820 IF R$="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 LIVE 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 BLANCHARGE", "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 B: INK W: CLS
30 REM BLACK IN BOTTOM OF SCREEN
40 INVERSE 1
50 FOR N=40 TO 6 STEP -8
60 PLOT PAPER 0:7,N: DRAW PAPER 0:241,0
70 NEXT N: INVERSE 0
100 REM DRAW 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 223-2*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
425 REM FILL IN EXTRA BITS
430 FOR N=0 TO 15
440 PLOT 255,160+N: DRAW 2*N-30,0
450 PLOT 0,63-N: DRAW 31-2*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 R;241,0
550 NEXT N
560 FOR N=112 TO 136 STEP 8
570 PLOT PAPER R;N,168: 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;185,83: 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;7,140: DRAW PAPER R;-70,35
690 INVERSE 0: PAPER 0: INK 7

```

ЕСЛИ ВЫ НЕ АНГЛИЧАНИН, ТО МОЖЕТЕ ИЗОБРАЗИТЬ 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 238,128: DRAW 4,0: REM MOUTH
100 REM SET UP WORD
110 INPUT W$: REM WORD TO GUESS
120 LET B=LEN W$: 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$(1)U$ THEN GO TO 200: REM GUESSED WAS RIGHT
400 REM DRAW NEXT PART OF GALLOW
410 IF D=B 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 OVER 1: REM RUB OUT MAN
520 LET X=240: GO SUB 1000
530 PLOT 238,128: DRAW 4,0: REM MOUTH
540 OVER 0: REM READRAW MAN
550 LET X=146: GO SUB 1000
560 PLOT 143,129: DRAW 6,0:PI/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 236,81: DRAW 4,21: DRAW 4,-21
690 OVER 1: REM LEGS
700 PLOT 255,66: 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 : GO TO 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 120,65,135,0,184,65,0,91

```

2010 DATA 168,65,16,16,184,81,16,-16
 2020 DATA 184,156,68,0,184,140,16,16
 2030 DATA 204,156,-20,-20,240,156,0,-16
 2020 DATA 184,156,68,0,184,140,16,16
 2030 DATA 204,156,-20,-20,240,156,0,-16

ПРИЛОЖЕНИЕ Е

ШЕСТНАДЦАТИРИЧНАЯ ДВОИЧНАЯ
СИСТЕМА 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 МАШИНОЕ 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).

А	
ABS.....	[CE]-G.....8
ACS.....	[CE]-SH-W.....10
AND.....	[CK],[CL],[CC]-SYMBOL SHIFT-Y.....11
ASN.....	[CE]-SH-O.....10
AT.....	[CK],[CL],[CC]-SYMBOL SHIFT-I.....13,21
ATN.....	[CE]-SH-E.....10
ATTR.....	[CE]-SH-L.....16
В	
BEEP.....	[CE]-SH-7.....19
BIN.....	[CE]-B.....12,40
BORDER.....	[CK]-B.....16
BREAK.....	CAPS SHIFT-SPACE.....4
BRIGHT.....	[CE]-SH-P.....15
С	
CAPS LOCK.....	[CK] [CL]-CAPS SHIFT-2.....1
CAPS SHIFT.....	1
CAT.....	[CE]-SYMBOL SHIFT-9.....21
CHR\$.....	[CE]-U.....11,12
CIRCLE.....	[CE]-SH-H.....17
CLEAR.....	[CK]-X.....22
CLOSE.....	[CE]-SYMBOL SHIFT-5.....21
CLS.....	[CK]-V.....13
CODE.....	[CE]-I.....11,26
CONTINUE.....	[CK]-C.....4
COPY.....	[CK]-Z.....21
COS.....	[CE]-W.....10
D	
DATA.....	[CE]-D.....6
DEF FN.....	[CE]-SYMBOL SHIFT-1.....8
DELETE.....	[CC],[CB]-O/[CK],[CL]-CAPS SHIFT-0.....1
DIK.....	[CK]-D.....10,11
DRAW.....	[CK]-W.....17
Е	
EDIT.....	[CK],[CL],[CC]-CAPS SHIFT-1.....1,3
ENTER.....	1
ERASE.....	[CE]-SYMBOL SHIFT-7.....21
EXP.....	[CE]-X.....9
F	
FLASH.....	[CE]-SH-V.....15
FN.....	[CE]-SYMBOL SHIFT-2.....8
FOR.....	[CK]-F.....5,23
FOR-NEXT ЦИКЛ.....	5
FORMAT.....	[CE]-SYMBOL SHIFT-0.....21
G	
GO SUB.....	[CK]-H.....5
GO TO.....	[CK]-G.....3
GRAPHICS.....	[CK],[CL],[CB]-CAPS SHIFT-0.....1
I	
IF.....	[CK]-U.....4,11
IN.....	[CE]-SH-I.....22
INK.....	[CE]-SH-X.....15
INKEY\$.....	[CE]-N.....18
INPUT.....	[CK]-I.....4,14
INVERSE.....	[CE]-SH-M.....15
INT.....	[CE]-R.....8

L		
LEN.....	[E]-K.....	8
LET.....	[K]-L.....	3,7,8
LINE.....	[E]-SYMBOL SHIFT-3.....	14
LIST.....	[K]-K.....	3
LLIST.....	[E]-V.....	21
LN.....	[E]-Z.....	9
LOAD.....	[K]-J.....	20
LPRINT.....	[E]-C.....	21

M		
MERGE.....	[E]-SH-T.....	20
MOVE.....	[E]-SYMBOL SHIFT-6.....	21

N		
NEW.....	[K]-A.....	3
NEXT.....	[K]-N.....	5,23
NOT.....	[K],[L],[E]-SYMBOL SHIFT-S.....	11

O		
OPEN.....	[E]-SYMBOL SHIFT-4.....	21
OR.....	[K],[L],[E]-SYMBOL SHIFT-U.....	11
OUT.....	[E]-O.....	22
OVER.....	[E]-H.....	15

P		
PAPER.....	[E]-C.....	15,16
PAUSE.....	[K]-M.....	18
PEEK.....	[E]-O.....	12,18
PL.....	[E]-M.....	10
PLOT.....	[K]-O.....	17
POINT.....	[E]-SYMBOL SHIFT-B.....	17
POKE.....	[K]-O.....	12
PRINT.....	[K]-P.....	3,7,13

R		
RANDOMIZE.....	[K]-T.....	10
READ.....	[E]-A.....	6
REN.....	[K]-E.....	4
RESTORE.....	[E]-S.....	6
RETURN.....	[K]-Y.....	5
RND.....	[E]-T.....	10
RUN.....	[K]-R.....	4

S		
SAVE.....	[K]-S.....	26
SCREEN\$.....	[E]-SH-K.....	13
SGN.....	[E]-F.....	8
SIN.....	[E]-O.....	10
SQR.....	[E]-H.....	8
STEP.....	[K],[L],[E]-SYMBOL SHIFT-D.....	5
STOP.....	[K],[L],[E]-SYMBOL SHIFT-A.....	5
STR\$.....	[E]-Y.....	8
SYMBOL SHIFT.....		1,5

T		
TAB.....	[E]-P.....	13,14,21
TAN.....	[E]-E.....	10
THEN.....	[K],[L],[E]-SYMBOL-S.....	11
TO.....	[K],[L],[E]-SYMBOL-F.....	5,7

U		
USR.....	[E]-L.....	12,26

V		
VAL.....	[E]-J.....	8
VAL\$.....	[E]-SH-J.....	8
VERIFY.....	[E]-SH-R.....	20

552 MS