

АКАДЕМИЯ НАУК СССР

НАУЧНЫЙ ЦЕНТР БИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

Методические рекомендации

Н.Л.Лунина

КУРС АЛГОРИТМИЧЕСКОГО  
ЯЗЫКА BASIC ДЛЯ ЭВМ  
"УАМАНА"

ПУЩИНО - 1988

АКАДЕМИЯ НАУК СССР  
НАУЧНЫЙ ЦЕНТР БИОЛОГИЧЕСКИХ ИССЛЕДОВАНИЙ  
НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЙ ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР

Методические рекомендации

Н. Л. Лунина

Курс алгоритмического  
языка BASIC для ЭВМ  
"YAMAHA"

ПУЩИНО – 1987

УДК 512.519.68

В работе изложен рассчитанный на школьников курс алгоритмического языка BASIC для ЭВМ "УМАНА". Использован опыт преподавания программирования учащимся Пушкинской экспериментальной средней школы.

Цель курса - подготовить учащихся к самостоятельной работе над учебными программами.

Идеи восходят к курсу программирования Кушниренко А.Г. (механико-математический факультет МГУ).

Научный редактор - И.Н. Семёнов

© Научный центр биологических исследований АН СССР  
в Пушкине, 1987 г.

## Предисловие

Известно, что алгоритмический язык BASIC обладает ведными возможностями. Этот факт вызывает в настоящее время бурную дискуссию о целесообразности обучения школьников BASICу. Автор считает, что, до тех пор пока альтернативы BASICу не существует, он будет использоваться в обучении. А в этих условиях нужно стремиться вести обучение BASICу так, чтобы использовать все его возможности для выработки у учащихся хорошего стиля программирования.

В основу данной работы положены идеи курса программирования А.Г.Кушниренко (механико-математический факультет МГУ). Содержащиеся в работе материалы были использованы при обучении программированию школьников Пущинской экспериментальной средней школы в рамках производственного обучения в 9-11 классах, в кружковой работе в 6-8 классах и частично – на уроках информатики. Учащиеся, овладевшие курсом, успешно разрабатывали учебные программы объемом 200-400 строк BASICа по различным учебным дисциплинам. Созданные программы были использованы на уроках по этим предметам.

Работа предназначена для учащихся, изучающих программирование при наличии учителя и доступа к ЭВМ "YAMAHA". Однако идеи, способ изложения материала и часть упражнений могут быть использованы и в других условиях.

Автор выражает благодарность М.Н.Семёновенкову за большую работу по редактированию рукописи, а также Э.Э.Шнолю, М.А.Ройтбергу, И.И.Фридрих за высказанные замечания и пожелания.

### 1. Введение в программирование

#### 1.1. Знакомство с линейными программами

На наших занятиях мы будем учиться программировать. Перед тем, как приступить к написанию программы, мы должны хорошо представлять себе, ЧТО нам нужно сделать и КАКИМИ СРЕДСТВАМИ мы располагаем для достижения цели. Для обозначения того, что нам нужно сделать, мы введем понятие ЗАДАЧА; для обозначения средств, помогающих нам добиваться результата – понятие ИСПОЛНИТЕЛЬ. Исполнителем может быть человек, прибор, устройство, организация. Исполнитель умеет производить некоторые ДЕЙСТВИЯ и распознавать приказы на выполнение этих действий. Один такой приказ мы будем называть ПРЕДПИСАНИЕМ, а всю их совокупность для одного исполнителя – СИСТЕМОЙ ПРЕДПИСАНИЙ данного исполнителя. Ввести в

рассмотрение новый исполнитель – это значит указать его систему предписаний и пояснить, какие именно действия происходят при выполнении предписаний.

Простейшая ПРОГРАММА представляют собой перечисленные друг за другом предписания, которые нужно выдать исполнителю для решения задачи. Такие программы, где после выполнения очередного предписания всегда выполняется следующее, называются ЛИНЕЙНЫМИ.

Иногда имеющийся в распоряжении исполнитель выполняет нашу задачу сразу, целиком: нужен каравай, хлопнули в ладоши – и "лежит хлеб, изукрашен разными хитростями, по бокам узоры печатные, сверху города с заставами". В таком случае программа состоит из одного предписания. Чаще возможности исполнителя значительно скромнее наших потребностей. Это порождает программы в сотни, тысячи, десятки тысяч предписаний.

Рассмотрим задачу о передвижении коня из одного угла шахматной доски в противоположный. Систему предписаний исполнителя КОНЬ можно описать следующим образом:

- шаг вверх, 2 шага вправо
- шаг вверх, 2 шага влево
- шаг вниз, 2 шага вправо
- шаг вниз, 2 шага влево
- шаг вправо, 2 шага вверх
- шаг вправо, 2 шага вниз
- шаг влево, 2 шага вверх
- шаг влево, 2 шага вниз

Поставленная задача может быть решена, например, в результате выполнения следующей программы:

```
шаг вправо, 2 шага вверх  
шаг вверх, 2 шага вправо  
шаг вправо, 2 шага вверх  
шаг вниз, 2 шага вправо  
шаг влево, 2 шага вверх  
шаг вверх, 2 шага вправо
```

Заметим, что программа существенно упростится, если вместо коня будет использована ладья, и станет совсем легкой для слона и ферзя (если их система предписаний будет соответствовать шахматным правилам).

Итак, запомним:

1. О программе можно говорить лишь тогда, у нас имеется задача и исполнитель. Обычная цепочка: ЗАДАЧА – ИСПОЛНИТЕЛЬ – ПРОГРАММА.
2. Может существовать не единственная программа, выполняющая данную задачу при помощи данного исполнителя.

3. С изменением задачи для одного и того же исполнителя, как правило, меняется программа. Меняется программа также и при смене используемого исполнителя для одной и той же задачи.

**ЗАДАНИЯ.**

1. Опишите систему предписаний исполнителей ПЕШКА, КОРОЛЬ.
2. Сколько предписаний содержится в системе предписаний исполнителя ФЕРЗЬ?
3. Напишите другие программы для решения задачи о перемещении коня в противоположный угол доски.

## 1.2. Знакомство с нелинейными программами

А теперь представим себе исполнитель ЛУНОХОД. Можно ли заранее, на Земле, предписать исполнителю такую программу, которая позволит ему в неизвестных условиях выполнять определенные задания, например, обойти прямоугольное препятствие по часовой стрелке? Какими возможностями должен обладать исполнитель для этого?

Пусть ЛУНОХОДу доступны предписания

сделать шаг  
поворнуться направо

и возможность делать проверку  
справа занято?

Заметим, что обход препятствия по часовой стрелке складывается из прохода вдоль четырех стен:

пройти вдоль северной стены  
пройти вдоль восточной стены  
пройти вдоль южной стены  
пройти вдоль западной стены

Такое разбиение задачи на части будет справедливым, если каждое предписание будет включать в себя

движение вдоль стены, пока справа занято  
поворот на 90 градусов для ориентации вдоль следующей стены  
шаг в новом направлении

Выпишем условия начального и конечного состояния ЛУНОХОДа для предписания "пройти вдоль северной стены"

**ДАНО:** ЛУНОХОД в начале северной стены, ориентирован на восток, справа занято  
**ПОЛУЧИТЬ:** ЛУНОХОД в начале восточной стены, ориентирован на юг, справа занято

Заметим, что аналогичная информация для других стен отличается только сторонами света. Попробуем переформулировать эти условия:

**ДАНО:** ЛУНОХОД в начале стены, ориентирован

по движению вдоль нее, справа занято  
ПОЛУЧИТЬ: ЛУНОХОД в начале следующей стены, ориентирован  
по движению вдоль нее, справа занято

В таком виде условия "ДАНО" и "ПОЛУЧИТЬ" полностью одинаковы для  
всех четырех предписаний, и вместо них можно использовать одно,  
повторив его 4 раза:

ЦИКЛ 4 РАЗА ВЫПОЛНИТЬ  
. пройти вдоль стены  
КОНЕЦ ЦИКЛА

При движении вдоль стены ЛУНОХОДу все время нужно проверять, за-  
нято ли справа. Если да, то нужно передвигаться вперед, а в про-  
тивном случае прекращать движение вдоль стены (достигнут конец  
стены). Приказ на выполнение такой последовательности действий и  
проверок записывается так:

ЦИКЛ ПОКА справа занято  
. выполнять  
. сделать шаг  
КОНЕЦ ЦИКЛА

После такого цикла ЛУНОХОД ориентирован вдоль пройденной стены;  
справа свободно. Нужно повернуть его направо и передвинуть на  
один шаг. Тогда он займет исходное положение для движения вдоль  
следующей стены.

Итак вся программа обхода препятствия записывается так:

ПРОГРАММА обход препятствия  
. ДАНО: ЛУНОХОД в начале стены, ориентирован по  
. движению вдоль нее, справа занято  
. ПОЛУЧИТЬ: ЛУНОХОД обошел препятствие по часовой стрелке  
. и вернулся в исходную точку  
. ЦИКЛ 4 РАЗА ВЫПОЛНИТЬ  
. . пройти вдоль стены  
. КОНЕЦ ЦИКЛА  
КОНЕЦ ПРОГРАММЫ

ПРЕДПИСАНИЕ пройти вдоль стены  
. ДАНО: ЛУНОХОД в начале стены, ориентирован по  
. движению вдоль нее, справа занято  
. ПОЛУЧИТЬ: ЛУНОХОД в начале следующей стены, ориентирован  
. по движению вдоль нее, справа занято  
. ЦИКЛ ПОКА справа занято  
. . выполнять  
. . сделать шаг  
. КОНЕЦ ЦИКЛА  
. УТВЕРЖДЕНИЕ. справа свободно

- повернуться направо
  - сделать шаг
- КОНЕЦ ПРЕДПИСАНИЯ

Разберем теперь более сложную задачу (при первом чтении ее можно опустить). Пусть нужно обойти препятствие произвольной формы, составленное из прямоугольников. Для этого нам понадобится более сложный исполнитель, например, МАРСОХОД с предписаниями

сделать шаг  
поворнуться направо  
поворнуться налево  
отметить точку  
справа занято?  
справа свободно?  
впереди свободно?  
впереди занято?  
точка отмечена?

Нужно отметить точку, где исполнитель находится в начальный момент, потом выйти из нее и двигаться вдоль стены до тех пор, пока проверка "точка отмечена?" не даст положительный ответ. При движении вдоль стены нужно все время проверять 2 условия — свободно ли впереди (чтобы не врезаться в стену) и занято ли справа (чтобы не оторваться от препятствия). Если нарушается условие "впереди свободно", то нужно поворачиваться налево до тех пор, пока это условие не выполнится. Если нарушается условие "справа занято" (угол), то нужно поворачиваться направо и делать шаг.

#### ПРОГРАММА обход препятствия произвольной формы

- ДАНО: МАРСОХОД рядом с препятствием произвольной формы, составленным из прямоугольников
- ПОЛУЧИТЬ: МАРСОХОД обошел препятствие и вернулся в исходную точку
- отметить точку
- перейти в неотмеченную точку
- ЦИКЛ ПОКА точка не отмечена
- . . . . . выполнять
- . . . ЦИКЛ ПОКА впереди свободно и справа занято
- . . . . . выполнять
- . . . . . сделать шаг
- . . . КОНЕЦ ЦИКЛА
- . . УТВЕРЖДЕНИЕ. впереди занято или справа свободно
- . . ЕСЛИ впереди занято
- . . . ТО ЦИКЛ ПОКА впереди занято
- . . . . . выполнять

```
    . . . . повернуться налево
    . . . . КОНЕЦ ЦИКЛА
    . . . . УТВЕРЖДЕНИЕ. впереди свободно
    . . . . ИНАЧЕ повернуться направо
    . . . . сделать шаг
    . . . . КОНЕЦ ЕСЛИ
    . . . . КОНЕЦ ЦИКЛА
    . . . . УТВЕРЖДЕНИЕ. точка отмечена
КОНЕЦ ПРОГРАММЫ
```

В этой программе встретилось одно предписание, которого нет в системе предписаний МАРСОХОДа - "перейти в неотмеченную точку". Нужно раскрыть его через доступные исполнителю предписания:

```
ПРЕДПИСАНИЕ перейти в неотмеченную точку
    . ДАНО: МАРСОХОД в отмеченной точке рядом с
        препятствием
    . ПОЛУЧИТЬ: МАРСОХОД в неотмеченной точке, ближайшей к
        отмеченной при движении по часовой стрелке
    . ЦИКЛ ПОКА впереди занято или справа свободно
    . . ВЫПОЛНЯТЬ
    . . . . повернуться налево
    . . . . КОНЕЦ ЦИКЛА
    . . . . УТВЕРЖДЕНИЕ. впереди свободно и справа занято
    . . . . сделать шаг
КОНЕЦ ПРЕДПИСАНИЯ
```

Познакомившись с этими программами, мы увидели, что

1. В программах очень часто встречаются нарушения последовательности выполнения операций.
  2. Эти нарушения связаны с необходимостью выполнить либо разветвление в зависимости от каких-то условий, либо многократно повторить часть программы.
  3. Повторять можно либо определенное число раз, либо до указанного условия.
  4. Исполнитель должен уметь выполнять проверки, встречающиеся в разветвлениях и повторах. Возможна комбинация таких условий.
  5. В программах можно использовать предписания, не принадлежащие исполнителю. В таком случае их нужно раскрыть через доступные исполнителю предписания.
- ЗАДАНИЕ. Смог ли бы МАРСОХОД справиться с поставленной задачей, если бы у него отсутствовало предписание "поворнуться налево"? А если бы отсутствовала проверка "точка отмечена"?

## 2. BASIC-машина как исполнитель

### 2.1. Понятие BASIC-машины

ЭВМ представляет собой исполнитель с очень богатыми возможностями. Она может служить помощником при вычислениях, обработке сложных текстов, рисовании графиков. Как и все исполнители, запросы на выполнение действий она принимает только в строго фиксированной форме, и мы можем говорить о системе предписаний ЭВМ. В этой работе мы будем говорить о том, какие предписания способна выполнять конкретная ЭВМ "Ямаха" в конкретной ситуации — без дисковода. В таких условиях мы имеем дело с BASIC-машиной.

BASIC-машина — это исполнитель, который позволяет создавать, исправлять и выполнять программы на языке BASIC. В систему предписаний BASIC- машины входят как предписания, которые употребляются внутри программ, так и предписания для работы с программами. Информацию, поступающую к ней, BASIC-машина пытается расшифровать (проприетаризовать) как одно из своих предписаний. Если такая попытка оказывается удачной, действие выполняется, в противном случае выдается сообщение об ошибке.

Система предписаний BASIC-машины очень велика. В этой работе мы рассмотрим 100 из них; список их приведен в приложении 1. Мы будем изучать эти предписания постепенно, сохраняя сквозную их нумерацию на протяжении всего текста. При вводе нового предписания будут указываться его номер по порядку, содержательная запись предписания и его запись на BASICe. Это будет выглядеть так

--- <номер> <предписание> ---  
<запись на BASICe>

или так

--- <номер> <предписание> \*\*\*  
<запись на BASICe>

Звездочками будут отмечены те предписания, которые рекомендуется запомнить; остальными можно пользоваться, заглядывая в справочник.

Например:

--- 20. Нарисовать точку с координатами (X,Y) цветом С \*\*\*  
PSET (X,Y),C

В программе можно употребить предписание PSET (10,20),15 для рисования белой точки с координатами (10,20), либо PSET (A,B),1 для рисования черной точки с координатами (A,B).

Признаком того, что BASIC-машина готова принимать очередное предписание, является наличие на экране текста OK, под которым

располагается курсор. Иногда машина по непонятным причинам долго не переходит в такое состояние. В этом случае нужно

--- 1. Прекратить выполнение текущего предписания XXX  
CTRL/STOP

(здесь и далее записи такого типа означают "нажать клавишу CTRL и, не отжимая ее - клавишу STOP").

Обмен информацией между BASIC-машиной и человеком происходит при помощи дисплея. Текст для человека выводится на экран дисплея, с той позиции, где находится курсор. Информацию, предназначенную для BASIC-машины, человек набирает на клавиатуре; при этом он видит набираемый текст на экране. Информация поступает BASIC-машине по нажатию клавиши "RETURN" (другие названия - "возврат каретки", BK, "ввод", ENTER, CR). При этом вводится та строка, в которой находился курсор.

## 2.2. Работа BASIC-машины в режиме калькулятора.

Работа с BASIC-машиной в режиме калькулятора - это такой способ работы, когда каждая программа состоит из одного предписания. Например, чтобы вычислить значение выражения  $2+2$ , нужно набрать на экране строку

?2+2

и нажать "RETURN". При этом появится текст

4

OK

Попробуйте вычислить значение другого выражения. Запомните правила записи выражений:

1. Допускаются следующие знаки арифметических операций: сложение-плюс (+), вычитание - тире (-), умножение - звездочка (\*), деление - косая черта (\ для деления нацело и / в других случаях), возведение в степень - галочка (^).
2. В вещественных числах в качестве разделителя между целой и дробной частью используется точка (а не привычная нам запятая). Для очень больших или очень маленьких чисел пользуются такой формой записи: 2.5E15, что означает  $2,5 \times 10^{15}$ ; при этом показатель степени при 10 может быть и отрицательным.
3. Скобки используются только круглые; количество открывающих скобок должно быть равно количеству закрывающих.
4. При переносе длинного арифметического выражения с одной строки на другую не требуется никаких дополнительных знаков.

Заметим, что наряду со знаком "?" для печати значения арифметического выражения можно употреблять слово PRINT; для BASIC-машины они являются синонимами. Мы чаще будем пользоваться именно второй возможностью.

Итак, мы познакомились с новым предписанием BASIC-машины

--- 2. Напечатать значение выражения \*\*\*  
PRINT <выражение>

Его называют также оператором печати.

Нередко бывает полезно увидеть рядом значения нескольких выражений. Для этого можно пользоваться одним оператором PRINT, перечислив эти выражения через точку с запятой, например

PRINT 2\*2; 3\*3; 4\*4

Выражения выважут не только арифметические. Примером не арифметического выражения служит текст, заключенный в двойные кавычки:

PRINT "2\*2=4"

PRINT "Земля вращается вокруг Солнца"

#### ЗАДАНИЯ.

1. Вычислить сумму натуральных чисел от 1 до 10.
2. Вычислить площадь квадрата, построенного на гипотенузе прямоугольного треугольника с катетами 5 и 13.
3. Вычислить произведение суммы первых 5 четных чисел на сумму первых 5 нечетных.
4. Вычислить частное от деления суммы кубов первых 5 натуральных чисел на сумму их квадратов.
5. Что больше – 2 в десятой степени или 10 во второй?
6. Радиус Земли 6378 км, плотность 5517 кг/м<sup>3</sup>. Вычислить массу Земли.
7. Представьте себе, что большим канатом опоясали Землю по экватору. Затем канат увеличили на 1 м; между ним и поверхностью образовался просвет. Сможет ли в него проскользнуть мышь высотой 2 см?
8. Площадь суши 149 млн. кв. км. Какую часть поверхности составляет Мировой океан?
9. Диаметр Венеры 12100 км. Ее масса составляет 81,5% массы Земли. Во сколько раз плотность Земли больше плотности Венеры?
10. Масса Венеры составляет 1:408400 массы Солнца. Диаметр Солнца 1392000 км. Вычислите среднюю плотность солнечного вещества.

При решении этих задач вам могут понадобится следующие сведения: длина окружности вычисляется по формуле  $2\pi R$ , площадь поверхности –  $4\pi R^2$ , объем шара –  $\frac{4}{3}\pi R^3$ ; плотность – это масса вещества в единице объема.

Часто возникает необходимость запомнить результаты промежуточных вычислений. Рассмотрим, например, задачу – вычислить площадь фигуры, состоящей из прямоугольника  $5 \times 4$  и примыкающего к нему равнобедренного прямоугольного треугольника со стороной 4, а также площадь составляющих ее частей.

Мы можем поступить так:

```
PRINT 5*4  
PRINT 4*4/2  
PRINT 5*4+4*4/2
```

При этом некоторые вычисления будут выполнены дважды. Было бы эффективнее вычислить и запомнить площади частей, а потом воспользоваться этими сведениями при вычислении площади всей фигуры.

**В Н И М А Н И Е** – для того, чтобы воспользоваться результатами в дальнейшем, мы должны, запомнив их, присвоить им имена – иначе мы не сможем никак на них сослаться.

--- 3. Запомнить значение выражения под указанным именем XXX  
`<имя>=<выражение>`

Это предписание называется также оператором присваивания.

Выберем для площади прямоугольника имя SP, а для площади треугольника – имя ST. Тогда можно действовать так:

```
SP=5*4  
ST=4*4/2  
PRINT SP  
PRINT ST  
PRINT SP+ST
```

**ЗАПОМНИМ:**

1. Имя должно содержать только латинские буквы и цифры (например, не может быть имени МАША)
2. Имя должно начинаться с буквы
3. Имена различаются по первым двум буквам (имена STREUG и STRAP не различаются; после выполнения операторов `STREUG=10`  
`PRINT STRAP` будет напечатано 10).

В качестве выражений мы будем на первых порах использовать

арифметические выражения. Они могут содержать не только числа и знаки операций, но и имена. Для имен в записи арифметического выражения действуют те же правила, что и для чисел; недопустимы записи типа  $2A$  или  $4BC+7XY$  (нельзя опускать знак умножения). При подсчете значения такого выражения используются значения, которые имена получили при помощи операторов присваивания.

После оператора

$SP=4\times 5$

под именем  $SP$  запомнено значение  $20$ ,

после  $SP=6\times 5$  –  $30$ ,

после  $SP=8\times 5$  –  $40$ .

Как видим, эти значения легко меняются; часто при решении задачи значения эти меняются многократно. Поэтому объекты, имеющие имена, называются ПЕРЕМЕННЫМИ. По существу переменная – это некоторая область памяти ЭВМ, а имя – это этикетка, наклейка, адрес, т.е. средство сослаться именно на это место памяти.

Отметим одну специфическую черту оператора присваивания: запись  $A=A+1$ ,

с точки зрения алгебры всегда являющаяся противоречивой, здесь означает:

- взять старое значение  $A$
- привавить 1
- сохранить увеличенное значение под именем  $A$ .

#### ЗАДАНИЯ.

1. Присвойте переменной  $A$  значение суммы чисел от 1 до 10.
2. Присвойте переменной  $B$  значение суммы квадратов этих чисел.
3. Напечатайте рядом значение переменных  $A$ ,  $B$ , их суммы и произведение.
4. Присвойте переменной  $C$  значение разности чисел  $A$  и  $B$ .
5. Попробуйте поменять местами значения переменных  $A$  и  $B$  (переменной  $A$  присвоить значение  $B$ , а переменной  $B$  присвоить значение  $A$ ). Изменится ли при этом значение  $C$ ?
6. Получите в переменной  $D$  среднее арифметическое  $A$  и  $B$ .
7. Получите в переменной  $E$  сумму квадратов  $A$  и  $B$ .
8. Получите в переменной  $F$  разность квадратов  $A$  и  $B$ .
9. Получите в переменной  $G$  произведение суммы  $A$  и  $B$  на их разность.
10. Напечатайте рядом значения переменных  $F$  и  $G$ .

Выполняя эти задания, можно заметить, что не всегда удобно каждое новое предписание писать с новой строки. Иногда удобна более компактная запись. BASIC допускает наличие нескольких предписаний в одной строке; разделителем между предписанием при этом служит двоеточие. Так, задания 8, 9 и 10 могут быть выполнены так:

```
F=A^2-B^2: B=(A+B)*(A-B): PRINT F;B
```

### 2.3. Средства работы с программами.

До сих пор мы работали с ЭВМ в своеобразном режиме: мы выдавали ей предписание, она немедленно выполняла его и сообщала "Ok" в знак готовности выполнять следующее. Но если у нас задача сложная, нам придется многократно повторять ввод предписаний при исправлении ошибок и работе с различными вариантами данных. Конечно, в этом случае такой режим работы неудобен. Появляется необходимость выполнить все предписания целиком, увидеть их на экране, напечатать на бумаге. Все это можно проделать с предписаниями, оформленными в виде программы. Программа на BASICе – это пронумерованная последовательность строк (напомним, строка BASICа может содержать одно или несколько предписаний).

Пример программы:

```
10 A=1+2+3  
20 B=1^2+2^2+3^2  
30 C=1^3+2^3+3^3  
40 PRINT A;B;C
```

BASIC-машина предоставляет следующие возможности работы с программами:

- 4. Начать работу с новой программой XXX  
NEW
- 5. Вывести текст программы на экран XXX  
LIST
- 6. Вывести текст программы на печать ---  
(выполняется только при включенном принтере)  
LLIST
- 7. Выполнить программу XXX  
RUN
- 8. Прекратить выполнение программы XXX  
CTRL/STOP

В командах LIST, LLIST, RUN могут быть указаны номера. Это будет означать вывод (распечатку) одной строки или выполнение программы

с указанной строки. Кроме того, можно вывести (распечатать) группу строк, указав номера первой и последней строк в группе через тире.

Примеры:

```
LIST 100-200    (вывести на экран строки с 100 по 200)
LLIST 50-      (вывести на принтер строки с 50' до конца)
RUN 30         (выполнить программу со строки 30)
```

Кроме того, для исправления ошибок в программах бывает нужно

- исправить строку
- удалить строку
- вставить строку

Для того, чтобы исправить строку, нужно получить ее на экране (командой LIST или LIST <номер строки>), сделать исправления и **ОБЯЗАТЕЛЬНО НАЖАТЬ КЛАВИШУ "RETURN"!** Если исправления значительны, можно заново набрать строку с таким же номером. При этом старая строка с таким номером заменится на новую.

Для удаления строки можно использовать предписание

--- 9. Удалить строку или группу строк ---

DELETE <номер строки>

или

DELETE <номер первой строки>-<номер последней строки>

Можно поступить и иначе: заменить эту строку на пустую, состоящую только из номера, набрав этот номер и нажав "RETURN".

Для того, чтобы вставить строку между двумя существующими, необходимо набрать ее с номером, лежащим в промежутке между номерами этих строк. Набирать можно в любом месте экрана. После этого можно выполнить команду LIST и убедиться, что строка "встала на свое место".

Иногда бывает невозможно указать промежуточный номер, т.к. номера "раздвигаемых" строк идут подряд. В таком случае программу нужно перенумеровать.

--- 10. Перенумеровать программу ---

RENUM

После выполнения этого предписания строки программы получат новые номера, идущие с 10 через 10.

Набирать номера строк вручную не очень удобно. Можно поручить это машине, воспользовавшись предписанием

--- 11. Начать режим автоматической нумерации строк \*\*\*  
AUTO

Набрав слово AUTO и нажав "RETURN", мы получим на экране число 10; набрав первую строку программы и нажав "RETURN", получим число 20 и т.д.

--- 12. Выйти из режима автоматической нумерации \*\*\*  
CTRL/C или CTRL/STOP.

Слова AUTO, LIST, RUN не обязательно набирать самому. Для этого можно пользоваться клавишами F2, F4, F5 соответственно.

ЗАДАНИЕ. Имеется программа:

```
10 PRINT "Наша таня громко плачет"  
20 PRINT "Уронила вречку мячик"  
30 PRINT "Уронила в речку мячик"  
40 PRINT "Не утонет вречке мяч"  
50 PRINT "Тише, Танечка, не плачь"
```

Исправьте ее так, чтобы в результате ее работы печатался правильный текст.

Наши программы, чем дальше, тем будут становиться все длиннее и сложнее. Договоримся о двух правилах оформления программ. Во-первых, в начале программы мы будем всегда писать некоторое пояснение для себя и преподавателя. Чтобы BASIC-машина не пытась работать со строкой, предназначенной для человека, в начале такой строки мы будем ставить специальный знак - апостроф ('), расположенный на клавиатуре под цифрой 7. Далее, мы всегда будем отмечать конец выполнения программы.

--- 13. Дальнейшую информацию в строке считать пояснением \*\*\*  
<строка на BASICe> ' <информация>

Пример.

```
5 ' стихотворение А.Барто  
40 PRINT "20 сентября" 'печатать даты
```

--- 14. Закончить выполнение программы \*\*\*  
END

Эти предписания имеют разновидности:

--- 15. Всю строку считать пояснением ---

<номер> REM <информация>

--- 16. Закончить выполнение программы и сообщить номер последнего выполненного оператора ---  
STOP

Впоследствии мы увидим, что операторы STOP и END могут располагаться в середине текста программы.

#### 2.4. Графические возможности BASIC-машины

До сих пор, работая с ЭВМ, мы имели дело только с буквами и цифрами. Такой режим работы дисплея ЭВМ называется алфавитно-цифровым. У некоторых дисплеев это единственный возможный режим работы. Наш дисплей может работать еще и в графическом режиме. При работе в таком режиме все точки экрана имеют две координаты; координата X может изменяться от 0 до 255, возрастая слева направо; Y – от 0 до 191, возрастая сверху вниз; можно зажечь или погасить любую точку, указав ее координаты.

При работе с экраном различают цвет фона, рисунка, бордюра. Понятие фона не отличается от общепринятого. Чтобы рисунок был виден, его нужно нарисовать цветом, отличным от цвета фона. Степерь рисунок – это значит нарисовать его фоновым цветом. Кроме того, вокруг всего поля, по которому рисуют, можно сделать рамку цветом, отличным от цвета фона – бордюр. Буквы в алфавитно-цифровом режиме тоже пишутся цветом рисунка. Сразу после включения BASIC-машины устанавливается цвет фона – синий, рисунка – белый.

Все цвета перенумерованы. Соответствие между цветом и его номером следующее:

|                    |                    |
|--------------------|--------------------|
| 1 – черный         | 9 – светло-красный |
| 2 – зеленый        | 10 – желтый        |
| 3 – светло-зеленый | 11 – светло-желтый |
| 4 – синий          | 12 – темно-зеленый |
| 5 – светло-синий   | 13 – малиновый     |
| 6 – коричневый     | 14 – серый         |
| 7 – голубой        | 15 – белый         |
| 8 – красный        |                    |

Для работы в графическом режиме существуют следующие предписания:

--- 17. Почистить экран ---  
CLS  
--- 18. Установить цвет рисунка/фона/бордюра ---  
COLOR C1,C2,C3

Напомним правила записи предписаний: на месте С1, С2, С3 здесь могут встречаться как числовые значения, так и переменные с другими именами, например COLOR 2,4,6 или COLOR CR, CF, CB.

- 19. Установить графический режим работы экрана \*\*\*  
SCREEN 2,2
- 20. Нарисовать точку с координатами (Х,Y) цветом С \*\*\*  
PSET(X,Y),C
- 21. Нарисовать точку с координатами (Х,Y) фоновым цветом ---  
PRESET(X,Y)
- 22. Провести отрезок из точки (Х1,Y1) в (Х2,Y2) цветом С \*\*\*  
LINE(X1,Y1)-(X2,Y2),C
- 23. Провести границу прямоугольника со сторонами, параллельными границам экрана, и противоположными углами в (Х1,Y1) и (Х2,Y2), цветом С ---  
LINE(X1,Y1)-(X2,Y2),C,B
- 24. То же. Прямоугольник закрасить цветом С ---  
LINE(X1,Y1)-(X2,Y2),C,BF
- 25. Нарисовать окружность с центром в точке (Х,Y), радиуса R, цветом С \*\*\*  
CIRCLE(X,Y),R,C
- 26. Нарисовать эллипс, у которого вертикальная полуось в N раз длиннее горизонтальной ---  
CIRCLE(X,Y),R,C,,,N
- 27. Закрасить цветом С область, ограниченную замкнутым контуром цвета С, начиная от внутренней точки (Х,Y) \*\*\*  
PAINT(X,Y),C
- 28. Получить цвет точки с координатами (Х,Y) в переменной С  
С=POINT(X,Y)
- 29. Разрешить вывод текста в графическом режиме ---  
OPEN "GRP:" AS #1
- 30. Вывести алфавитно-цифровую информацию на экран в графическом режиме ---  
PRINT #1,<выражение>

**ЗАДАЧА.** Нарисовать горизонтальный отрезок, проходящий через середину экрана:  
10 'горизонтальный отрезок через середину  
20 SCREEN 2,2  
30 LINE (0,95)-(255,95)  
40 LINEINPUT A\$

50 END

Новый оператор в строке 40 – LINEINPUT A\$ – мы употребили для того, чтобы задержать экран в графическом режиме до тех пор, пока не будет нажата клавиша "RETURN". В противном случае сразу после рисования отрезка программа завершает работу, и экран возвращается в алфавитно-цифровой режим – мы не успеваем рассмотреть рисунок.

**ЗАДАНИЯ.**

1. Нарисовать вертикальный отрезок.
2. Нарисовать биссектрису левого верхнего угла экрана.
3. Нарисовать биссектрису правого верхнего угла экрана.
4. Нарисовать треугольник.
5. Нарисовать открытый конверт.
6. Нарисовать окружность с центром в середине экрана.
7. Нарисовать треугольник, квадрат, окружность. Рисунки подписать.
8. Нарисовать 5 олимпийских колец.
9. Нарисовать снеговика.
10. Нарисовать робота.

**2.5. Типы переменных. Работа со строками**

**Сравнивай операторы**

PRINT 2

и

PRINT "Зима",

мы видим, что машина умеет работать как с числовыми данными, так и с текстами. Но значение 2 можно присвоить какой-нибудь переменной. Можно ли присвоить некоторой переменной значение "Зима"? Переменную со значением 2 можно увеличивать и уменьшать, возводить в квадрат и т.п. Какие операции можно применить к строке "Зима"?

Переменные бывают разного типа. Определить тип переменной – это значит указать,

- какие значения она может принимать,
- какие операции можно с ней выполнять.

Мы будем различать переменные следующих типов:

- целые
- вещественные
- строчные

Переменные двух первых типов кажутся нам привычными. Они принимают числовые значения и могут быть использованы в одном и том же

арифметическом выражении. По умолчанию (при отсутствии явного указания типа) переменные считаются вещественными. Чтобы объявить переменную целой, нужно либо указывать знак "%" после ее имени, либо использовать оператор DEFINT (DEFINE INTEGER - определить целой).

--- 31. Объявить целыми все переменные, имена которых начинаются с указанной буквы ---  
DEFINT <буква>

Например, оператор DEFINT I объявляет целыми все переменные, имена которых начинаются на букву I. Примеры.

|                 |                 |
|-----------------|-----------------|
| 10 'иллюстрация | 10 'иллюстрация |
| 20 PI=3.14      | 20 DEFINT I     |
| 30 I%=PI        | 30 PI=3.14      |
| 40 PRINT I%     | 40 I=PI         |
| 50 END          | 50 PRINT I      |
|                 | 60 END          |

Примеры демонстрируют работу с целыми переменными. В обоих случаях будет напечатано число 3.

С целыми переменными ЭВМ работает быстрее.

Примером нечисловых переменных в BASICе могут служить переменные типа строка.

--- 32. Объявить строками все переменные, имена которых начинаются с указанной буквы \*\*\*  
DEFSTR <буква>

Другой возможный способ объявить это - указывать знак "\$" после имени переменной:

|                 |
|-----------------|
| 10 DEFSTR A     |
| 20 A="Петя"     |
| 30 B\$="Иванов" |

Для работы со строками существуют следующие предписания:

--- 33. Объединение строк A\$ и B\$ запомнить под именем C\$ \*\*\*  
C\$=A\$+B\$

Если A\$="Петя", B\$="Иванов", то после выполнения C\$=A\$+" "+B\$ C\$ получит значение "Петя Иванов"

--- 34. В переменной L получить длину строки A\$ \*\*\*  
L=LEN(A\$)

В нашем случае после операторов L1=LEN(A\$); L2=LEN(B\$); L3=LEN(C\$) переменные L1, L2, L3 получат значения 4, 6, 11 соответственно

--- 35. В переменной A\$ получить N левых символов строки B\$ ---  
A\$=LEFT\$(B\$,N)

После выполнения D\$=LEFT\$(C\$,6) D\$ получит значение "Петя И"

--- 36. В переменной A\$ получить N правых символов строки B\$ ---  
A\$=RIGHT\$(B\$,N)

После выполнения D\$=RIGHT\$(C\$,5) D\$ получит значение "ванов"

--- 37. В переменной A\$ получить значение L символов строки B\$, начиная с позиции N \*\*\*  
A\$=MID\$(B\$,N,L)

После выполнения оператора D\$=MID\$(C\$,6,4) переменная D\$ получит значение "Иван". В частном случае, когда второе число опущено, выбирается часть строки с указанной позиции до конца: после выполнения E\$=MID\$(C\$,6) E\$ получит значение "Иванов".

#### ЗАДАНИЯ.

1. В переменной DA\$ содержится дата в виде месяц-число-год, например 02-25-1987. Получить в переменной D2\$ дату в виде число-месяц-год.
2. В переменной BR\$ содержится имя и отчество так, что под имя отведено 10 позиций, например "Николай Степанович". В переменной IM\$ содержится имя его сестры. Получить в переменной SE\$ имя и отчество сестры.
3. В переменной SL\$ содержится существительное первого склонения, оканчивающееся на "а". Напечатать это слово во всех падежах.
4. В переменной PR\$ содержится прилагательное мужского рода, заканчивающееся на "ый". Напечатать женский и средний род этого прилагательного.
5. В строке IO\$ длиной 30 содержатся имя, отчество и фамилия так, что каждая часть занимает 10 позиций, например "Антон Сергеевич Степанов ". Напечатать фамилию и инициалы.

6. В переменных A\$ и B\$ содержатся слова, состоящие только из корней, например "лес" и "воз". Напечатать все сложные слова, которые можно получить из них при помощи соединительных гласных "о" или "е".
7. Известно, что сложное слово A\$ состоит из двух частей одинаковой длины и соединительной гласной. Напечатать отдельно обе части.
8. Длины сторон треугольника заданы в переменных A, B, C. Напечатать фразу "Рассмотрим треугольник со сторонами ... см, ... см, ... см", где вместо многоточий указаны соответствующие значения.
9. В переменной A\$ содержится прилагательное с окончанием "ий", в переменной B\$ — с окончанием "ый" (например, A\$="синий", B\$="красный"). Напечатать все возможные сложные прилагательные (например, "красно-синий").
10. В переменной A\$ задано предложение, состоящее из 4 слов одинаковой длины, разделенных пробелами. Напечатать каждое слово.

#### 2.6. Ввод-вывод.

До сих пор мы определяли значения переменных до начала работы программы. Значения других переменных получались в процессе выполнения на основании определенных нами переменных и различных операций (например, умножения или объединения строк). Эти полученные значения сообщались нам при помощи оператора PRINT. Мы не могли вмешаться и изменить значения переменных в процессе выполнения. Это, конечно, резко снижает ценность программы: гораздо полезнее программа, которая может просклонять любое задаваемое слово, чем та, которая это делает только с одним заранее заданным словом.

Мы познакомимся сейчас с несколькими предписаниями, предназначенными для ввода информации в процессе выполнения. Поработав с ними, мы увидим, что они обладают рядом отличий. Эти различия и будут, в конце концов, определять выбор одного из этих предписаний в конкретной ситуации.

- ```

--- 38. Набранную на клавиатуре до клавиши "RETURN" последовательность символов запомнить под именем A в виде, соответствующем типу переменной A XXX
      INPUT A
--- 39. Набранную на клавиатуре до клавиши "RETURN" последовательность символов запомнить в строке A$ XXX
      
```

```
LINEINPUT A$  
--- 40. Н набранных символов запомнить в строке A$ ---  
A$=INPUT$(N)
```

**ЗАМЕЧАНИЕ.** При использовании оператора INPUT\$(N) после набранных N символов не следует нажимать "RETURN".

**ЗАДАНИЕ.**

1. Спросить имя пользователя и поздороваться с ним по имени.
2. Спросить имя ученика, потом отчество, потом фамилию. После этого полученную информацию напечатать в одну строку.
3. Ввести с клавиатуры 3 числа и подсчитать их среднее арифметическое.
4. Ввести длины 3 сторон и вычислить объем и площадь поверхности соответствующего прямоугольного параллелепипеда.
5. Ввести радиус. Вычислить длину окружности, площадь поверхности и объем шара.
6. Вывести на экран условие примера и получить в переменной ANS значение ответа, введенного пользователем. Значение ANS напечатать.
7. Просклонять слово, введенное с клавиатуры. Известно, что это существительное 1 склонения с окончанием "я".
8. С клавиатуры ввести 2 слова: прилагательное с окончанием "ый" и существительное женского рода. Напечатать согласованное словосочетание.
9. В графическом режиме принять от пользователя пятибуквенное слово. Слово печатать в левом верхнем углу экрана по мере поступления букв.
10. Проверить, можно ли в графическом режиме пользоваться оператором LINEINPUT для ввода текста.

### 3. Организация данных

#### 3.1. Массивы

Нам неоднократно приходилось иметь дело с однотипными данными. До сих пор мы хранили такие данные независимо друг от друга. Например, 6 падежных окончаний можно хранить в шести разных переменных – 01\$, 02\$, 03\$, 04\$, 05\$, 06\$. Но логичнее объединить их в группу и работать с элементами этой группы по заранее определенным правилам. BASIC предоставляет возможность организовывать данные в массивы. Массив – это группа однотипных данных с общим

именем. К элементу осуществляется доступ по его номеру, который называется также индексом. Для работы с массивом используется предписание

--- 41. Объявить А массивом длины N  
DIM A(N)

После такого объявления мы можем работать с элементами массива А с номерами от 1 до N, как с обычными переменными; номер элемента массива указывается при этом в круглых скобках после имени. Например, для задачи с падежными окончаниями начало программы будет выглядеть так:

```
DIM OK$(6)
OK$(1)="а":OK$(2)="ы":OK$(3)="е":OK$(4)="у":OK$(5)="ой":
OK$(6)="е"
```

Для склонения введенного слова нужно "отщипнуть" последнюю букву и 6 раз напечатать корень вместе с соответствующим окончанием.

```
10 "склонение
20 DIM OK$(6)
30 OK$(1)="а":OK$(2)="ы":OK$(3)="е":OK$(4)="у"
40 OK$(5)="ой":OK$(6)="е"
50 INPUT SL$
60 K0$=LEFT$(SL$,LEN(SL$)-1)
70 PRINT K0$+OK$(1)
80 PRINT K0$+OK$(2)
90 PRINT K0$+OK$(3)
100 PRINT K0$+OK$(4)
110 PRINT K0$+OK$(5)
120 PRINT K0$+OK$(6)
130 END
```

Позже мы научимся записывать такие программы более удобно.

Такой массив является примером одномерного массива. Существуют и данные с более сложной организацией. Они имеют не один, а несколько индексов. Положение фигур на шахматной доске, например, естественно описывать с помощью двумерного массива, где один из индексов задает номер вертикали, второй — номер горизонтали: DIM CB\$(8,8). Этот массив содержит 64 элемента с индексами (1,1), (1,2), ..., (1,8), (2,1), ..., (8,8). CB\$(5,1) содержит информацию о поле Е1, CB\$(8,8) содержит информацию о поле НВ.

**ЗАДАНИЯ.**

1. Сколько массивов и какой размерности понадобится, чтобы хранить данные из следующей таблицы:

|                    |  |  |
|--------------------|--|--|
| Джомолунгма   8848 |  |  |
| Чогори   8611      |  |  |
| Дхаулагири   8221  |  |  |
| Нангапарбат   8126 |  |  |

2. В массиве FA\$(5) задать фамилии первых по списку 5 учеников класса. Фамилии напечатать.
3. Проспрягать глагол, заканчивающийся на "ать", используя массив окончаний.
4. В массиве TE(7) задать температуру 7 дней недели. Вычислить среднюю температуру за неделю.
5. В массиве SL\$(3) задать 3 слова. В массиве LE(3) получить их длины. Слова и их длины напечатать.
6. Сколько массивов и какой размерности понадобится, чтобы хранить данные из следующей таблицы:

| I                                                  | II | III | IV | V | VI | VII |
|----------------------------------------------------|----|-----|----|---|----|-----|
| Азия   Эверест   8848   Чогори   8611              |    |     |    |   |    |     |
| Африка   Кения   5199   Пик Маргерита   5109       |    |     |    |   |    |     |
| Южная Америка   Аконкагуа   6960   Ильимани   6882 |    |     |    |   |    |     |

7. Заполнить массив NO(2,8), в котором должны находиться порядковые номера химических элементов в зависимости от периода (первый индекс) и группы (второй индекс). Ненулевые значения массива напечатать вместе со значением индексов.
8. Откройте учебник с периодической системой Менделеева. Для семи элементов от Na до Cl заполните массив EL(7,3), где указано количество электронов на трех электронных оболочках. Напечатайте данные в виде
- Na - 2 8 1
9. Последовательность РНК состоит из нуклеотидов У, Ц, А, Г. При считывании белка с РНК одному триплету (участку из трех нуклеотидов) соответствует одна аминокислота. Здесь в таблице приведено соответствие аминокислот триплетам РНК для 16 триплетов из 64 возможных:

| I | I | У   | I | Ц   | I | А   | I | Г   | I |   |
|---|---|-----|---|-----|---|-----|---|-----|---|---|
| I | I | Фен | I | Сер | I | Тир | I | Цис | I | У |
| I | У | Фен | I | Сер | I | Тир | I | Цис | I | Ц |
| I | I | Лей | I | Сер | I | XXX | I | XXX | I | А |
| I | I | Лей | I | Сер | I | XXX | I | Trp | I | Г |

Из таблицы, например, видно, что триплету УУУ соответствует аминокислота, обозначаемая Фен; УЦА – Сер; УГГ – Трп; трем триплетам никакая аминокислота не соответствует, они выполняют специальную функцию. В соответствии с таблицей присвойте значения 16 элементам трехмерного массива AM\$(4,4,4) (AM\$(1,1,1)= "Фен").

10. "Прессмейстер сделал ход "E2-E4". Какие элементы массива CB\$ изменил Остап Ибрагимович?

### 3.2. Последовательности

Другим способом организации данных является последовательность.

--- 42. Создать последовательность ---

<номер> DATA <элементы последовательности через запятую>

Разделителем в последовательности могут служить либо запятые, либо следующий оператор DATA. Так, записи

10 DATA понедельник, вторник, среда

20 DATA четверг, пятница, суббота

можно рассматривать либо как две последовательности из 3 элементов, либо как одну из 6.

--- 43. Встать в начало последовательности ---

RESTORE <номер>

--- 44. Прочитать очередной элемент последовательности

в переменную А ---

READ A

После выполнения этого предписания очередным становится следующий элемент последовательности.

Проиллюстрируем использование этих средств на примере программы склонения имен существительных:

```
10 ' склонение '
20 DATA а,ы,е,у,ой,е
30 INPUT SL$
40 K0$=LEFT$(SL$,LEN(SL$)-1)
50 RESTORE 20
60 READ OK$: PRINT K0$+OK$
70 READ OK$: PRINT K0$+OK$
80 READ OK$: PRINT K0$+OK$
90 READ OK$: PRINT K0$+OK$
100 READ OK$: PRINT K0$+OK$
110 READ OK$: PRINT K0$+OK$
120 END
```

Обратите внимание, что, в отличие от массива, в последовательности

- невозможен доступ к элементу по номеру (чтобы воспользоваться пятым элементом последовательности, нужно прочитать предыдущие 4);
- значения элементов последовательности можно задавать только в момент создания; невозможно изменять значения элементов уже созданной последовательности.

При попытке прочитать элементов больше, чем содержится в последовательности, происходит отказ

OUT OF DATA

#### ЗАДАНИЯ.

1. Задайте последовательность названий крупных рек Европы и напечатайте их.
2. Задайте последовательность данных и напишите программу печати для следующих сведений:  
Волга - 3531  
Дунай - 2857  
Урал - 2428  
Днепр - 2200
3. Задайте последовательность центров 5 окружностей и нарисуйте их.
4. Задайте последовательность центров и радиусов 5 окружностей и нарисуйте их.
5. Задайте последовательность радиусов 5 концентрических окружностей и нарисуйте их.
6. Задайте последовательность вершин пятиугольника и нарисуйте его.
7. Имеется последовательность из 6 чисел, представляющих дневные

--- 50. Переменной A присвоить значение функции Y(X) в точке X---  
A=FNY(X)

Пример. В результате выполнения участка программы

```
10 DEF FNY(X)=XXX  
20 FOR X=11 TO 13  
30 PRINT X;"-";FNY(X)
```

```
40 NEXT  
будет напечатано  
11 - 121  
12 - 144  
13 - 169
```

## 7. Дополнительные возможности

### 7.1. Работа со строками

Продолжим знакомство с переменными типа строка. Назовем подстрокой часть строки.

--- 51. В переменной N получить, с какой позиции начинается подстрока B\$ в строке A\$ при просмотре с позиции P ---  
N=INSTR(P,A\$,B\$)

При C\$="Петя Иванов" после выполнения оператора N=INSTR(1,C\$,"в") переменная N получит значение 7, после N=INSTR(8,C\$,"в") – значение 11.

--- 52. Создать строку A\$ из N одинаковых символов S\$ ---  
A\$=STRING\$(N,S\$)

После выполнения A\$=STRING\$(7,"x") A\$ получит значение "xxxxxxx". Частным случаем этого предписания является следующее:

--- 53. Создать строку A\$ из N пробелов ---  
A\$=SPACE\$(N)

Существуют строки, состоящие из цифр, например, A\$="12345". В зависимости от ситуации с ними удобнее работать то как с числами, то как со строками. Для обеспечения таких преобразований существуют два взаимно обратных предписания.

```
FOR I=N1 TO N2 STEP N3
    .
    .
NEXT
```

Если не указан шаг приращения параметра, он полагается равным 1.

Выполняется эта конструкция следующим образом:

1. Параметр полагается равным N1.
2. Выполняются все операторы до слова "NEXT".
3. Параметр увеличивается на N3.
4. Происходит проверка – значение параметра не превысило N2?
5. Если превысило, то выполнение конструкции прекращается; выполняется следующий за словом "NEXT" оператор. Если нет, то происходит переход к п.2.

Проиллюстрируем использование оператора цикла:

```
10 'печатать четных чисел
20 FOR I=0 TO 20 STEP 2
30 PRINT I
40 NEXT
50 END
10 'склонение
20 DIM OK$(6)
30 OK$(1)="а":OK$(2)="ы":OK$(3)="е":OK$(4)="ы"
40 OK$(5)="ой":OK$(6)="е"
50 INPUT SL$
60 KO$=LEFT$(SL$,LEN(SL$)-1)
70 FOR I=1 TO 6
80 PRINT KO$+OK$(I)
90 NEXT
```

Между операторами FOR и NEXT могут встретиться любые операторы, в том числе и другая конструкция цикла. Например, в результате выполнения программы

```
10 'вложенные циклы
20 FOR I=1 TO 2
30   FOR J=1 TO 3
40     PRINT "Выполняем цикл"
50   NEXT
60 NEXT
```

```
70 END
```

слова "Выполняем цикл" будут напечатаны 6 раз. Нужно внимательно следить, чтобы операторы внутри цикла не изменяли значения параметра цикла, а именно, чтобы имя параметра цикла не встречалось в операторах присваивания слева или не было параметром вложенных циклов. Проверьте, сколько раз будет напечатан текст в результате выполнения такой программы

```
10 'вложенные циклы
20 FOR I=1 TO 2
30   FOR I=1 TO 3
40     PRINT "Выполняем цикл"
50   NEXT
60 NEXT
70 END
```

#### ЗАДАНИЯ.

1. Напечатать числа от 3 до 30, кратные 3.
2. Напечатать числа от 0 до 50, дающие при делении на 5 остаток 1.
3. Напечатать квадраты целых чисел от 1 до 10.
4. Вычислить и напечатать сумму чисел; сумму квадратов; сумму кубов от 1 до 10.
5. Вычислить и напечатать произведение целых чисел от 1 до 10.
6. Нарисовать 5 горизонтальных отрезков.

Один из возможных вариантов решения:

```
10 'горизонтальные отрезки
20 SCREEN 2
30 FOR X=0 TO 200 STEP 50
40   LINE (X,100)-(X+30,100)
50 NEXT
60 LINEINPUT A$
70 END
```

7. Нарисовать 5-угольник.
8. Провести в 5-угольнике все диагонали, выходящие из одной точки.
9. Нарисовать 5 окружностей.
10. Заданное слово напечатать по одной букве в строке.

#### 4.2. Конструкция ветвления

Другим случаем, когда нарушается естественная последовательность выполнения предписаний в программе, являются ветвления —

необходимость выполнить ту или иную часть программы в зависимости от условий, которые заранее не известны.

Пример.

ЕСЛИ в именительном падеже окончание "а"  
. ТО в родительном "ы"

- в винительном "у"
- в творительном "ой"
- ИНАЧЕ в родительном "и"
- в винительном "ю"
- в творительном "ей"

КОНЕЦ ЕСЛИ

После слова "ЕСЛИ" здесь должно располагаться выражение, отвечающее на вопрос "верно ли?". Такие выражения называются логическими. Примеры логических выражений:

2 < 5  
8 >= 12  
"север" <> "юг"

Значения этих выражений "да", "нет", "да".

В логических выражениях могут быть использованы отношения

|                  |           |
|------------------|-----------|
| больше           | >         |
| меньше           | <         |
| больше или равно | >= или => |
| меньше или равно | <= или =< |
| равно            | =         |
| не равно         | <> или >< |

При этом первые 4 отношения используются только с числовыми величинами, а последние два - как с числовыми, так и со строчными.

Пара простых логических выражений может быть соединена в сложные при помощи логических операций. Существует несколько логических операций; наиболее часто встречаются "и" (AND) и "или" (OR).

Правила вычисления логических выражений, содержащих логические операции, следующие:

| AND | да  | нет |
|-----|-----|-----|
| да  | да  | нет |
| нет | нет | нет |

| OR  | да | нет |
|-----|----|-----|
| да  | да | да  |
| нет | да | нет |

Т.е. выражение с AND верно только тогда, когда верны обе части; с OR – когда верна хотя бы одна из частей.

Пример.

Выражение  $2 < 5$  AND  $3 > 10$  принимает значение "нет"  
 $1 < 4$  OR "a" = "b" "да"

Отметим, что сложные логические выражения также могут соединяться логическими операциями. Если выражение содержит несколько логических операций, они выполняются следующим образом: сначала получают значения все операции сравнения; затем AND слева направо, затем OR. Скобки используются по аналогии с арифметическими выражениями.

Пример.

$1 > 5$  OR "A"  $\neq$  "B" AND  $2 \times 2 = 4$   
нет OR да AND да  
нет OR да  
да

--- 47. ЕСЛИ ... ТО ... ИНАЧЕ ... КОНЕЦ ЕСЛИ \*\*\*

IF ... THEN ... ELSE

Здесь после IF может находиться любое логическое выражение.

Пример.

ПРОГРАММА поиск максимального элемента  
• присвоить значения элементам массива  
• максимальный := значение первого элемента  
• ЦИКЛ меняя номер элемента от 2 до количества элементов  
• . . . ВЫПОЛНЯТЬ  
• . . . ЕСЛИ текущий элемент больше максимального  
• . . . ТО максимальный := значение текущего элемента  
• . . . КОНЕЦ ЕСЛИ  
• КОНЕЦ ЦИКЛА  
• напечатать максимальный  
КОНЕЦ ПРОГРАММЫ

```
10 ' поиск максимального элемента
20 DIM A(10)
30 FOR I=1 TO 10: INPUT A:NEXT
40 M=A(1)
50 FOR I=2 TO 10
60   IF A(I)>M THEN M=A(I)
70 NEXT
80 PRINT "Максимальный элемент=";M
```

**90 END**

Конструкции ветвления могут быть вложенными, т.е. после THEN или после ELSE также может встретиться IF. При разворачивании конструкций следует иметь в виду, что ELSE относится к последнему из возможных THEN. Так, конструкция

```
IF A<5 THEN IF A<2 THEN PRINT "Вася" ELSE PRINT "Петя"  
ELSE IF A>8 THEN PRINT "Коля" ELSE PRINT "Сережа"
```

эквивалентна следующей

```
ЕСЛИ A<5  
    . ТО ЕСЛИ A<2  
        . ТО печатать "Вася"  
        . ИНАЧЕ печатать "Петя"  
    . КОНЕЦ ЕСЛИ  
    . ИНАЧЕ ЕСЛИ A>8  
        . ТО печатать "Коля"  
        . ИНАЧЕ печатать "Сережа"  
    . КОНЕЦ ЕСЛИ  
КОНЕЦ ЕСЛИ
```

#### **ЗАДАНИЯ.**

1. Найти минимальный элемент массива.
2. Подсчитать количество букв "а" в заданном слове.
3. Подсчитать количество слов в заданной фразе.
4. Подсчитать количество удвоенных букв в заданной фразе.
5. Нарисовать квадрат или круг по запросу пользователя.
6. Определить число (единственное или множественное) предъявленного существительного.
7. Определить, является ли предъявленное существительное существительным 3-его склонения.
8. Определить род предъявленного прилагательного.
9. Просклонять существительное 1-го склонения (с окончанием "а" или "я").
10. Написать тренажер умножения на 2 с реакцией "Верно" или "Неверно".

#### **4.3. Цикл "пока"**

Рассмотрим задачу: напечатать все числа, квадрат которых не превышает 200. Для решения этой задачи нужно присвоить числу минимальное значение, допустимое по характеру задачи, затем многократно его увеличивать и производить проверку: квадрат получившегося числа не больше 200? При решении задач такого типа использу-

ется конструкция

ЦИКЛ ПОКА <условие>

- ВЫПОЛНЯТЬ
- <действия>

КОНЕЦ ЦИКЛА

В BASICе такая конструкция реализуется при помощи конструкции ветвления и оператора перехода:

```
10 'квадраты чисел до 200
20 A=1
30 IF A^2 > 200 THEN GOTO 70
40 PRINT A; A^2
50 A=A+1
60 GOTO 30
70 'превышает
```

Заметим, что использование оператора перехода в других ситуациях нарушает структуру программы и не поощряется.

#### ЗАДАНИЯ.

1. Вычислить и напечатать все степени двойки, не превышающие 33000.
2. От левого края экрана до правого нарисовать расположенные вплотную окружности с радиусами, увеличивающимися вдвое. Начать с радиуса 2.
3. На всю высоту экрана нарисовать лесенку с увеличивающимися по высоте ступеньками.
4. Нарисовать ветку грабовы  $y=x^2$  при помощи отдельных точек, соответствующих целым значениям  $x$ . Рисование закончить, когда ордината превысит граничное значение.
5. Написать тренажер таблицы умножения на 2. Правильность ответа проверять. После третьего неверного ответа подряд сообщать подсказку и переходить к следующему примеру.
6. На нарисованном горизонтальном отрезке отметить середину; на левом отрезке снова отметить середину и т.д. Процесс закончить, когда разница между соседними отметками станет равной 1.
7. Найти максимальное целое число, куб которого не превышает 33000.
8. Для борьбы с жуками юннаты выпустили в охраняемый участок леса дятлов. После этого численность жуков стала ежедневно уменьшаться.

- шаться не менее чем наполовину. Через сколько дней будет уничтожен последний жук, если в начале этой операции их насчитывалось 10000?
9. В задаваемых пользователем словах подсчитывать количество букв "а". Признаком окончания сеанса работы с пользователем считать слово "xxx".
  10. Найти максимальное число, квадрат которого не превышает 2 (с точностью до 0.01).

## 5. Задачи

### Циклы в вычислительных задачах

1. Вычислить  $1+2+4+8+\dots+2^{10}$
  2. Вычислить  $1\times 2+2\times 3+3\times 4+\dots+10\times 11$
  3. Вычислить  $(1+2)\times(1+2+3)\times\dots\times(1+2+\dots+10)$
  4. Вычислить  $1 + 1\times 2 + 1\times 2 \times 3 + \dots + 1\times 2 \times \dots \times 10$
  5. Вычислить  $1\times 2/(3\times 4) + 2\times 3/(4\times 5) + \dots + 10\times 11/(12\times 13)$
  6. Числами Фибоначчи называют числа следующей последовательности:  
1, 1, 2, 3, 5, 8...
- Здесь каждое следующее число равно сумме двух предыдущих.
- Вычислить 20-е число Фибоначчи.
7. Вычислить сумму 10 чисел Фибоначчи.
  8. Сколько членов суммы  $1+1/2+1/3+\dots$  нужно взять, чтобы результат превысил 1.5?
  9. То же для суммы  $1+1/4+1/9+\dots$
  10. Сколько членов суммы  $1+1/2+1/4+1/8+1/16+\dots$  нужно взять, чтобы результат оказался не менее 1.999? не менее 2? превысил 2?

### Циклы в графических задачах

11. Провести 5 горизонтальных отрезков, расположенных друг под другом.
12. Провести 6 вертикальных отрезков, верхние концы которых лежат на одной горизонтальной прямой.
13. Провести 8 горизонтальных отрезков увеличивающейся длины, расположенных друг под другом.
14. Провести 10 горизонтальных отрезков одинаковой длины, левые концы которых расположены по биссектрисе левого верхнего угла экрана.
15. Провести 5 вертикальных отрезков одинаковой длины, верхние концы которых расположены по биссектрисе правого верхнего угла

16. Нарисовать 10 рядов по 5 горизонтальных отрезков.
17. Нарисовать 6 рядов по 20 вертикальных отрезков.
18. Нарисовать 7 прямоугольников, обозначенных границами, друг под другом.
19. Нарисовать 9 закрашенных прямоугольников увеличивающейся высоты, верхние стороны которых лежат на одной прямой.
20. Нарисовать границы 8 квадратов одинакового размера, центры которых лежат на биссектрисе левого верхнего угла экрана.
21. Нарисовать 12 закрашенных квадратов увеличивающегося размера, левые верхние углы которых лежат на биссектрисе левого верхнего угла экрана.

Одно из возможных решений:

```

10 'увеличивающиеся квадраты вдоль главной диагонали
20 SCREEN 2,2
30 X=0
40 L=10
50 FOR N=1 TO 12

60 LINE (X,X)-(X+L,X+L),,BF
70 X=X+L 'очередной левый верхний угол
80 L=L+2 'размер очередного квадрата
90 NEXT
100 LINEINPUT A$
110 END

```

22. Нарисовать 5 рядов по 6 квадратов.
23. Нарисовать 15 квадратов с одинаковым центром.
24. Нарисовать "разбегающийся" квадрат (сначала маленький, потом, вместо него — чуть побольше, с тем же центром, и т.д.)
25. Нарисовать 5-угольник. Провести в нем все диагонали.
26. Нарисовать 5-угольник. Нарисовать другой 5-угольник с вершинами в серединах сторон исходного.
27. Нарисовать 5 окружностей с центрами на одной горизонтальной прямой.
28. Нарисовать 7 окружностей увеличивающегося радиуса с центрами на одной вертикальной прямой.
29. Нарисовать 8 закрашенных кругов одинакового радиуса с центрами на биссектрисе правого верхнего угла.

Одно из возможных решений:

```

10 'круги вдоль второй диагонали
20 SCREEN 2,2

```

```
30 FOR X=20 TO 160 STEP 20
40 CIRCLE (255-X,X),9
50 PAINT (255-X,X)
60 NEXT
70 LINEINPUT A$
80 END
```

30. Нарисовать 11 концентрических окружностей.  
31. Нарисовать "разбегающуюся" окружность.  
32. Нарисовать закрашенный прямоугольник, "разбегающийся" вправо  
33. Нарисовать "мерцающий" квадрат, который поочередно становится  
то белым, то синим.  
34. Нарисовать точку, "перебегающую" из левой части экрана в пра-  
вую  
35. Движение точки по окружности с центром в  $(X_0, Y_0)$  и радиусом  $R$   
можно изобразить так:

```
FOR A=0 TO 6.28 STEP 0.05
    X=X0+R*X*SIN(A)
    Y=Y0-R*X*COS(A)
    PSET (X,Y)
NEXT
```

Изобразить движение Луны вокруг Земли, которая, в свою оче-  
редь, движется вокруг Солнца.

#### Работа с массивами

36. Провести 6 отрезков, один конец которых – точка  $(0,0)$ , а коор-  
динаты других концов заданы.  
37. Нарисуйте треугольник с вершинами в заданных точках.  
38. За последнюю неделю температура изменялась следующим образом:  
 $-3, +5, -2, -7, +1, +4, +2$ . Изобразите на экране график, при-  
няв за нулевой уровень горизонтальный отрезок, проходящий че-  
рез середину экрана.  
39. Постойте пятиугольник с вершинами в заданных точках. Точку,  
расположенную внутри пятиугольника, соедините с серединами  
сторон.  
40. Некоторое государство находится на круглом острове. Столица  
располагается в центре острова. 4 крупных города расположены  
строго на север, юг, запад и восток от столицы на полпути к  
морю. Нарисуйте карту государства. Затем по очереди "помигай-  
те" кружками, изображающими города.

### Циклы и массивы при работе со строками

41. В переменной D содержится число, в переменной M\$ – месяц, в переменной Y – год. Получить в переменной DA\$ дату в виде "20 сентября 1987 года". Известно, что дата осенняя или зимняя.

42. Данное слово напечатать в обратном порядке, например "машина – анишам".

43. Заданы названия зверей в виде

```
10 DATA тигр  
20 DATA зубр  
30 DATA кит  
40 DATA волр
```

Напечатать рядом названия взрослых зверей и их детенышей:  
"тигр-тигренок"

44. Даны названия детенышей: гусенок, лосенок, рысенок. Написать названия взрослых зверей путем отбрасывания суффикса "енок" и добавления мягкого знака.

45. Даны пары слов:

```
10 DATA Александр Невский  
20 DATA Дмитрий Донской  
30 DATA Илья Муромец  
40 DATA Алеша Попович  
50 DATA Довриня Никитич
```

Напечатать отдельно первые, затем отдельно вторые слова из каждой пары.

46. В одном массиве заданы цвета – красный, синий, белый, желтый, в другом – названия игрушек: кубик, грузовик, самолет. Напечатать все возможные сочетания, например, "синий кубик".

47. Последовательность ДНК состоит из нуклеотидов 4 видов: А, Т, Г, Ц. Известно, что особое внимание при изучении ДНК уделяется триплетам – участкам из трех нуклеотидов. Напечатайте все возможные триплеты ДНК.

48. Имеется список, содержащий фамилии, имена, отчества в виде:

```
Сидоров Иван Петрович  
Козлович Константин Олегович  
Кузнецов Игорь Владимирович  
Смирнов Сергей Максимович
```

На основании этого списка получить список имен отцов.

49. Вспомним задачу о государстве на острове (N40). Назовем города "Центральный", "Южный", "Северный", "Западный", "Восточный". Требуется, показывая город, выводить в левый верхний угол экрана его название. Одно из возможных решений:

```

10 *города
20 DATA Центральный,Северный,Южный,Западный,Восточный
30 DEFSTR T
40 C1=1:CF=15:COLOR C1,CF:SCREEN 2,2:OPEN "GRP:" AS #1
50 DIM X(5),Y(5)
60 X(1)=128:Y(1)=96:R=50:R1=2 *центр, радиус острова, радиус города
70 X(2)=X(1):X(3)=X(1):X(4)=X(1)-R/2:X(5)=X(1)+R/2
80 Y(2)=Y(1)-R/2:Y(3)=Y(1)+R/2:Y(4)=Y(1):Y(5)=Y(1)
90 CIRCLE (X(1),Y(1)),R *нарисовали остров
100 RESTORE 20
110 FOR I=1 TO 5
120 LINE (0,0)-(255,7),CF,BF *стерли предыдущую надпись
130 READ T
140 PRESET (0,0):PRINT #1,T *вывели название
150 FOR J=1 TO 10
160 CIRCLE (X(I),Y(I)),R1,CF
170 CIRCLE (X(I),Y(I)),R1,C1
180 NEXT *помигали
190 NEXT *перебрали все города
200 LINEINPUT A$
210 END

```

50. Нарисуйте свою карту и присвойте городам свои названия

#### Циклы и ветвления при работе со строками

51. Нарисовать большой круг, большой квадрат, маленький круг, маленький квадрат в зависимости от запроса пользователя.
52. Подсчитать количество гласных в заданном слове.
53. Определить, какая гласная встречается в заданной фразе наиболее часто.
54. Первые 10 элементов таблицы Менделеева имеют следующие знаки:  
 водород - H, гелий - He, литий - Li, бериллий - Be, бор - B,  
 углерод - C, азот - N, кислород - O, фтор - F, неон - Ne.  
 Написать тренажер знаков химических элементов.
55. Заданы английские слова и их перевод на русский язык в виде

```

10 DATA mother,мама
20 DATA father,папа

```

Написать тренажер по переводу слов.

56. Усложнить программу так, чтобы считался правильным любой из нескольких возможных переводов. Данные задавать в виде

10 DATA mother, "мать, мама"  
20 DATA father, "отец, папа"

57. Вспомните программу для склонения существительных 1-го склонения, задаваемых пользователем. Напишите программу-тренажер по проверке правильности склонения заданного набора существительных 1-го склонения.
58. Укажите множественное число предъявленного существительного.
59. Напишите тренажер по проверке умения образовать множественное число предъявляемых ученику существительных.
60. Определить длину предъявленного слова. Напечатать "В этом слове N букв". Употребить форму "буква", "буквы" или "букв" в зависимости от N.
61. В заданной нити ДНК найти и напечатать все участки из 4-х одинаковых нуклеотидов подряд.
62. Две нити ДНК обладают тем свойством, что нуклеотиды на соответствующих местах образуют пары А-Т или Г-Ц, например

АТГЦААТТГГ  
ТАЦГTTAAЦЦ

По заданной одной нити построить вторую.

63. РНК состоит из нуклеотидов А, У, Г, Ц. Особый интерес представляют участки РНК, в которых равноотстоящие от концов нуклеотиды комплементарны (А-У, Г-Ц). Для каждого из заданных участков РНК выяснить, обладает ли он таким свойством.
64. Известно, что процесс считывания белка с РНК начинается с триплета-инициатора АУГ и заканчивается одним из терминаторов УАА, УАГ или УГА. При считывании одному триплету соответствует одна аминокислота. В заданной РНК найдите все участки, откуда может быть считан белок:
- АУАУАУГАААУГУГГЦАГАУААУГУУАГАЦАУГАААУГАУУУ
65. При считывании белка каждому триплету РНК соответствует одна аминокислота. Зависимость эта приведена в таблице. Из нее, например, следует, что триплет УЦА кодирует серин (Сер), а триплет ЦУГ - лейцин (Лей). Напишите программу, которая строит белок по заданной последовательности РНК.

|  |   |     |     |     |     |     |     |     |     |   |   |
|--|---|-----|-----|-----|-----|-----|-----|-----|-----|---|---|
|  |   | У   |     | Ц   |     | А   |     | Г   |     |   |   |
|  |   | Фен |     | Сер |     | Тир |     | Цис |     | У |   |
|  | У |     | Фен |     | Сер |     | Тир |     | Цис |   | Ц |
|  |   | Лей |     | Сер |     | *** |     | *** |     | А |   |
|  |   | Лей |     | Сер |     | *** |     | Трп |     | Г |   |

| I | I | У   | I   | Ц   | I   | А   | I   | Г   | I   | I |
|---|---|-----|-----|-----|-----|-----|-----|-----|-----|---|
|   |   |     |     |     |     |     |     |     |     |   |
|   |   | Лей |     | Про |     | Гис |     | Арг |     | У |
|   | Ц |     | Лей |     | Про |     | Гис |     | Арг |   |
|   |   | Лей |     | Про |     | Гин |     | Арг |     | А |
|   |   | Лей |     | Про |     | Гин |     | Арг |     | Г |
|   |   |     |     |     |     |     |     |     |     |   |
|   |   | Иле |     | Тре |     | Асн |     | Сер |     | У |
|   | А |     | Иле |     | Тре |     | Асн |     | Сер |   |
|   |   | Иле |     | Тре |     | Лиз |     | Арг |     | Ц |
|   |   | Мет |     | Тре |     | Лиз |     | Арг |     | Г |
|   |   |     |     |     |     |     |     |     |     |   |
|   |   | Вал |     | Ала |     | Асп |     | Гли |     | У |
|   | Г |     | Вал |     | Ала |     | Асп |     | Гли |   |
|   |   | Вал |     | Ала |     | Глу |     | Гли |     | Ц |
|   |   | Вал |     | Ала |     | Глу |     | Гли |     | А |
|   |   |     |     |     |     |     |     |     |     |   |

## 6. Разработка программ "сверху вниз"

### 6.1. Подпрограммы

Маленькая программа имеет неоспоримые преимущества перед большой. Но вспомним, что программа определяется как поставленной задачей, так и имеющимся в распоряжении исполнителем (например, программы для передвижения в противоположный конец доски коня и ферзя). Как быть, когда в нашем распоряжении нет исполнителя, позволяющего написать короткую программу?

В этом случае нам нужно разбить задачу на части и написать программу в предположении, что подходящий исполнитель для этих частей имеется. Программа получится короткой и наглядной. А теперь расплатимся за то, что воспользовались несуществующим предписанием: опишем, как они могут быть выполнены с помощью другого, менее сложного исполнителя. Получится текст примерно такой структуры:

```

ПРОГРАММА
  < вызов предписания 1>
  < вызов предписания 2>
  < вызов предписания 3>
  . . .
КОНЕЦ ПРОГРАММЫ

```

```
< описание предписания 1>
< описание предписания 2>
< описание предписания 3>
```

Предположим пока для простоты, что описания предписаний 1, 2, 3 не содержат ничего, недоступного имеющемуся исполнителю. Тогда задача решена. Важно помнить лишь, что вызов созданного нами предписания должен отличаться по внешнему виду от его описания.

Для этой цели BASIC предоставляет следующие возможности:

- 47. Выполнить предписание, описание которого начинается с указанного оператора XXX  
GOSUB <номер оператора>
- 48. Конец предписания XXX  
RETURN

В BASICе и ряде других языков такие создаваемые нами предписания называются подпрограммами.

В BASICе не предусмотрено специального оператора, отмечающего начало описания предписания. Мы всегда будем начинать описание предписания (подпрограмму) с комментария, в котором содержится название предписания. Это поможет нам при работе с большими программами – как своими, так и с программами друг друга.

При необходимости описания предписаний могут содержать вызовы других предписаний, т.е. подпрограммы могут быть вложенными.

#### ЗАДАНИЕ.

Для двух приведенных программ выпишите номера операторов в порядке выполнения:

|                 |                 |
|-----------------|-----------------|
| 10 'программа 1 | 10 'программа 2 |
| 20 GOSUB 100    | 20 GOSUB 100    |
| 30 GOSUB 200    | 30 GOSUB 200    |
| 40 GOSUB 300    | 40 END          |
| 50 END          | 100 '           |
| 100 '           | 110 . . .       |
| 110 . . .       | 120 . . .       |
| 120 . . .       | 130 RETURN      |
| 130 RETURN      | 200 '           |
| 200 '           | 210 GOSUB 300   |
| 210 . . .       | 220 . . .       |
| 220 RETURN      | 230 RETURN      |
| 300 '           | 300 '           |

```
310 . . .
320 . . .
330 RETURN
```

```
310 . . .
320 RETURN
```

А теперь проиллюстрируем, как разрабатывают большие программы. Вспомним задачу о городах на острове (задача №49). Устроим из нее тренажер: пусть на карте миганием отмечается положение города, а ученик вводит его название. Это будет довольно большая программа, содержащая явно выраженные части:

ПРОГРАММА города

- . начать работу (инициализировать массивы)
  - . нарисовать карту
  - . ЦИКЛ 5 РАЗ ВЫПОЛНИТЬ
    - . . указать город
    - . . принять ответ
    - . . проверить ответ
    - . . ЕСЛИ ответ правильный
      - . . . ТО поощрить
      - . . . ИНАЧЕ указать на ошибку
    - . . КОНЕЦ ЕСЛИ
  - . КОНЕЦ ЦИКЛА
  - . кончить работу (сообщить результаты)
- КОНЕЦ ПРОГРАММЫ

На BASICе это записывается так:

```
10 'тренажер "ГОРОДА"
20 GOSUB 1000 'начать работу
30 GOSUB 2000 'нарисовать карту
40 FOR NN=1 TO 5
50   GOSUB 3000 'указать город
60   GOSUB 4000 'принять ответ
70   GOSUB 5000 'проверить ответ
80   IF OP=1 THEN GOSUB 6000
        ELSE GOSUB 7000 'обработка ответа
90 NEXT
100 GOSUB 8000 'кончить работу
110 END
1000 'начать работу
1990 RETURN
2000 'нарисовать карту
2990 RETURN
3000 'указать город
3990 RETURN
```

```

4000 'принять ответ
4990 RETURN
5000 'проверить ответ
5990 RETURN
6000 'обработка правильного ответа
6990 RETURN
7000 'обработка неправильного ответа
7990 RETURN
8000 'кончить работу
8990 RETURN

```

Специфическая особенность BASICa: все переменные по характеру своего использования являются общими для всех подпрограмм. Поэтому в подпрограммах 3000-7000 нельзя изменять значение переменной цикла NN, а в подпрограмме "принять ответ" переменная OR должна получать значение 1 в случае правильного ответа и другое значение – в случае неправильного.

**ЗАДАНИЕ.** Закончить тренажер

Одна из возможных реализаций предписания "Принять ответ"

```

4000 'принять ответ
4010 ANS="" 'почистить переменную для ответа
4020 LINE (0,0)-(255,7),CF,BF 'почистить место на экране
                               для ответа; CF-цвет фона
4030 PRESET (0,0) 'начало выводимого текста
4040 A=INPUT$(1)
4050 IF ASC(A)=13 THEN 4090
4060 ANS=ANS+A
4070 PRINT #1,A; 'вывести очередной символ ответа
4080 GOTO 4040 'цикл пока не нажато "RETURN"
4090 RETURN

```

## 6.2. Функции

Часто при решении задач вычислительного характера бывает полезен исполнитель, обладающий предписанием "вычислить значение указанной функции в указанной точке". Такая возможность позволяет свести к минимуму исправления в программе вычислительного характера при переходе от одной функции к другой. Для этого применяют предписания

```

--- 49. Определить функцию Y(X) ---
DEF FNY(X)=<выражение>

```

--- 50. Переменной A присвоить значение функции Y(X) в точке X---  
A=FNY(X)

Пример. В результате выполнения участка программы

```
10 DEF FNY(X)=X*X  
20 FOR X=11 TO 13  
30 PRINT X;"-";FNY(X)  
40 NEXT
```

будет напечатано

```
11 - 121  
12 - 144  
13 - 169
```

## 7. Дополнительные возможности

### 7.1. Работа со строками

Продолжим знакомство с переменными типа строка. Назовем подстрокой часть строки.

--- 51. В переменной N получить, с какой позиции начинается подстрока B\$ в строке A\$ при просмотре с позиции P ---  
N=INSTR(P,A\$,B\$)

При C\$="Петя Иванов" после выполнения оператора N=INSTR(1,C\$,"в") переменная N получит значение 7, после N=INSTR(8,C\$,"в") – значение 11.

--- 52. Создать строку A\$ из N одинаковых символов S\$ ---  
A\$=STRING\$(N,S\$)

После выполнения A\$=STRING\$(7,"x") A\$ получит значение "xxxxxxx". Частным случаем этого предписания является следующее:

--- 53. Создать строку A\$ из N пробелов ---  
A\$=SPACE\$(N)

Существуют строки, состоящие из цифр, например, A\$="12345". В зависимости от ситуации с ними удобнее работать то как с числами, то как со строками. Для обеспечения таких преобразований существуют два взаимно обратных предписания.

--- 54. В переменной N получить числовое значение строки A\$ ---  
N=VAL(A\$)

В нашем случае числовая переменная N получит значение 12345; это значение можно будет складывать, вычитать и пр.

--- 55. Получить строку A\$, представляющую запись числа N ---  
A\$=STR\$(N)

В нашем случае строка A\$ получит значение "12345"; в этой строке можно будет выделять левые, правые, средние символы и пр.

Каждый символ имеет свой код (соответствующее этому символу число). Например, символ А большое русское имеет код 225. Более того, клавиши, которым никакой символ не соответствует, также имеют свой код (например, клавиша "RETURN" имеет код 13). Существуют два взаимно обратных предписания, позволяющие по значению символа получить его код и наоборот.

--- 56. В переменной N получить код символа A\$ ---  
N=ASC(A\$)

--- 57. В переменной A\$ получить символ с кодом N ---  
A\$=CHR\$(N)

Для работы со строками отводится некоторый ресурс памяти, не превышающий 200 символов. В своей работе мы можем исчерпать этот ресурс, и тогда программа завершится аварийно с сообщением

OUT OF STRING SPACE

Получив такое сообщение, мы должны внимательно посмотреть на программу и постараться сделать ее более эффективной. Однако, если мы не видим такой возможности, можно увеличить соответствующий ресурс при помощи предписания

--- 58. Увеличить ресурс для работы со строками до N символов ---  
CLEAR N

Следует помнить, что увеличение этого ресурса происходит за счет уменьшения других.

#### ЗАДАНИЯ.

1. Определить, сколько раз встречается указанное слово в указанном тексте.
2. В строке A\$ содержится предложение. Напечатать длины слов, со-

держащихся в предложении.

3. Из приведенного текста выбрать все слова, в которых встречается буква "т".
4. В строке 10\$ содержатся имя, отчество и фамилия через пробел, например "Антон Сергеевич Степанов". Напечатать фамилию и инициалы.
5. Из приведенного текста выбрать все предложения, в которых встречается слово "стоит". Проверьте правильность работы программы на тексте:  
"На окраине города стоит здание школы. Оно состоит из нескольких частей – младшего корпуса и старшего, а также бассейна и двух спортивных залов. Стоит ли удивляться, что ученики так любят свою школу? На каникулах здесь состоятся немало интересных соревнований."
6. Пользуясь операторами STRING\$ и SPACE\$, напечатать состоящие из звездочек прямоугольник; треугольник; трапецию.
7. Для слова, введенного с клавиатуры, напечатать коды всех его букв. Выполните программу для слов "MARS", "mars", "марс", "МАРС".
8. Введите с клавиатуры некоторое слово, набранное большими латинскими буквами. Напечатайте это слово маленькими латинскими; маленькими русскими; большими русскими буквами.
9. В строке A\$ содержатся сведения о росте учеников в таком виде "Иванов-158, Петров-162, Сидоров-154". Вычислить средний рост учеников.
10. Оператор INPUT A в том случае, когда A является переменной целого или вещественного типа, выдает ошибку при попытке ввести текст. Этой ошибки можно избежать, если вводить не число, а строку, и определять числовое значение введенной строки. Написать тренажер таблицы умножения с использованием этой возможности.

## 7.2. Дополнительные возможности ввода-вывода

Все изученные нами операторы ввода обладали общим свойством: при выполнении их программа переходила в состояние ожидания до тех пор, пока не поступала нужная информация. Существует и другая возможность.

--- 59. Опросить клавиатуру \*\*\*  
A\$=INKEY\$

При выполнении этого предписания переменная A\$ получает либо значение нажатой клавиши, либо значение пустой строки, если не нажата ничего. При этом конструкция

```
10 A$=INKEY$: IF A$="" THEN 10
20 PRINT A$
оказывается эквивалентной
10 A$=INPUT$(1)
20 PRINT A$
```

Оператором INKEY\$ часто пользуются для того, чтобы выполнять некоторые действия в то время, пока идет обдумывание и ввод ответа. ЗАДАЧА. Экран находится в графическом режиме. Пока идет ввод слова, передвигать точку из левого конца экрана в правый. Ввод слова заканчивать либо по нажатию клавиши "RETURN" с сообщением "Слово введено", либо тогда, когда точка достигнет границы экрана, с сообщением "Не успели".

```
10 'ввод слова
20 SCREEN 2,2:OPEN "GRP:" AS #1
30 XT=0: YT=100: PSET (XT,YT)
40 XL=0: YL=0
50 A$=INKEY$: IF A$="" THEN XT=XT+1:PSET (XT,YT):IF XT <
255 THEN GOTO 50 ELSE S$="Не успели":GOTO 80
60 IF ASC(A$)=13 THEN S$="Слово введено":GOTO 80
70 PRESET (XL,YL):PRINT #1,A$:XL=XL+8:GOTO 50
80 'обработка конца
90 PRESET (0,170):PRINT #1,S$:LINEINPUT A$:END
```

#### ЗАДАНИЯ.

1. Изменить программу так, чтобы точка двигалась сверху вниз; по диагонали экрана; чтобы двигались две точки навстречу друг другу.
2. Определить коды клавиш "пробел", BS, HOME, INS, DEL, стрелок.
3. Поставьте точку в середине экрана. Передвигайте ее влево, вправо, вверх, вниз в зависимости от нажатой стрелки.

#### 7.3. Конструкция выбора

Нам нередко приходится сталкиваться с ситуациями, когда выбор нужно делать не из двух, как в конструкции ветвления, а из трех или более возможностей, например

##### ВЫБОР

- окончание "ый" или "ий" => обработать мужской род
- окончание "ая" или "яя" => обработать женский род

. окончание "ое" или "ее" => обработать средний род  
. иначе => напечатать "не прилагательное"  
КОНЕЦ ВЫБОРА

Средства BASICa не позволяют реализовать такую конструкцию в полной мере. Возможно, однако, ветвление в том случае, когда мы рассматриваем несколько пронумерованных возможностей и нам известен номер того условия, которое выполняется в данный момент:

--- 60. Перейти на один из операторов в зависимости от N ---  
ON <N> GOTO <номер1>, <номер2>, <номер3>...

При N=1 происходит переход на оператор с первым номером из указанного списка, при N=2 - на оператор со вторым номером и т.д. Если N<1 или больше количества номеров в списке, происходит переход на следующий оператор.

Наглядным примером подобной задачи является последняя задача предыдущего раздела. Имеются 4 возможности (стрелки вправо, влево, вверх, вниз); коды этих клавиш 28, 29, 30, 31; в качестве номера условия можно пользоваться выражением "<код клавиши>-27".

#### ПРОГРАММА управления стрелками

. поставить точку в середине экрана  
. ЦИКЛ ПОКА не нажата клавиша "RETURN"  
. . ВЫПОЛНЯТЬ  
. . . опросить клавиатуру  
. . . ЦИКЛ ПОКА ничего не нажато  
. . . . ВЫПОЛНЯТЬ  
. . . . ничего не делать  
. . . КОНЕЦ ЦИКЛА  
. . . УТВЕРЖДЕНИЕ. Нажата клавиша  
. . . ЕСЛИ нажата не стрелка или "RETURN"  
. . . . ТО ничего не делать  
. . . . ИНАЧЕ  
. . . . . ВЫБОР  
. . . . . . стрелка вправо => увеличить X  
. . . . . . стрелка влево => уменьшить X  
. . . . . . стрелка вверх => уменьшить Y  
. . . . . . стрелка вниз => увеличить Y  
. . . . КОНЕЦ ВЫБОРА  
. . . . изобразить точку с новыми координатами  
. . . КОНЕЦ ЕСЛИ  
. . КОНЕЦ ЦИКЛА  
КОНЕЦ ПРОГРАММЫ

```

10 'управление стрелками
20 SCREEN 2,2: X=128: Y=96: PSET (X,Y)
30 A$=INKEY$: IF A$="" THEN 30
40 IF ASC(A$)=13 THEN 130
50 IF ASC(A$)<28 OR ASC(A$)>31 THEN 30
60 ON ASC(A$)-27 GOTO 70, 80, 90, 100
70 X=X+1: GOTO 110
80 X=X-1: GOTO 110
90 Y=Y-1: GOTO 110
100 Y=Y+1
110 PSET (X,Y)
120 GOTO 30
130 END

```

**ЗАДАНИЕ.**

1. Для введенного пользователем числа указать

- удвоенное число
- квадрат числа
- сумму цифр числа

в зависимости от номера запроса, указанного пользователем.

2. Задайте в массиве OK\$(6) падежные окончания прилагательного в зависимости от его окончания в именительном падеже. Предполагается, что начало программы следующее:

```

10 'выбор окончаний
20 DIM OK$(6)
30 INPUT SL$
40 OK$(1)=RIGHT(SL$,2)
50 DATA ый,ий,ая,яя,ое,ее
60 RESTORE 50
70 FOR I=1 TO 6
80 READ O1$
90 IF OK$(1)=O1$ THEN 120
100 NEXT
110 PRINT "Это не прилагательное":END
120 N=I

```

3. Выведите на экран в графическом режиме слово и черточку – указатель – под первой буквой. По нажатию стрелки вправо передвигайте указатель вправо на следующую букву до конца слова, по нажатию стрелки влево – влево до конца слова. Старое положение указателя затирать.

4. Нарисуйте на экране 10 пронумерованных прямоугольников разного цвета. Изобразите точку, движением которой можно управлять при помощи стрелок. По нажатию клавиши "RETURN" определяйте и печатайте номер прямоугольника, в котором в данный момент находится точка ( $\emptyset$  – вне прямоугольников).
5. Нарисуйте 20 прямоугольников, 4 ряда по 5. Изобразите точку, отмечающую центры этих прямоугольников следующим образом: в начале программы точка находится в центре левого верхнего прямоугольника; при нажатии стрелки вправо точка перемещается в соседний правый прямоугольник, если такой имеется; если его нет – остается на месте; аналогично и для остальных стрелок. Старое положение точки затирать.

#### 7.4. Построение движущихся изображений

Для построения движущихся изображений нужны средства

- описать рисунок;
  - вывести рисунок в указанное место экрана;
  - вывести рисунок в новое место экрана,
- стерев его на прежнем месте.

BASIC-машина решает эту проблему при помощи спрайтов.

Спрайт – это одноцветный рисунок фиксированного размера, который можно перемещать по экрану. Размер рисунка не превышает 8×8 точек (в режиме SCREEN 2,1) или 16×16 точек (в режиме SCREEN 2,2). В случае, если рисунок превышает указанные размеры, его можно составить из нескольких спрайтов.

##### Описание спрайта

Рисунок кодируется двумерной таблицей 8×8 (или 16×16), каждый элемент которой указывает, принадлежит ли соответствующая точка рисунку или нет. Элемент равен 1 для точки, принадлежащей рисунку, и  $\emptyset$  – для не принадлежащей. Так, рисунок "черные клетки шахматной доски" кодируется следующей таблицей

```

01010101
10101010
01010101
10101010
01010101
10101010
01010101
10101010

```

Образуем последовательность, элементы которой связаны со строками таблицы таким образом:

```
DATA &B01010101,&B10101010,&B01010101,&B10101010,  
&B01010101,&B10101010,&B01010101,&B10101010
```

На основании этой последовательности получим строку по следующим правилам:

```
A$="":FOR I=1 TO 8:READ J:A$=A$+CHR$(J):NEXT
```

Такая строка используется для описания спрайта.

--- 61. Описать спрайт с номером NS XXX  
SPRITE\$ (NS) = <строка>

Здесь строка должна быть связана с рисунком указанным образом.

Так описывается спрайт 8x8. При описании спрайта 16x16 образуются 32 восьмерки нулей и единиц, и последовательность строится по ним следующим образом: сначала рассматриваются все левые восьмерки сверху вниз, затем все правые сверху вниз.

#### Движение спрайтов

Для работы со спрайтами вводится понятие логической плоскости. Их насчитывается 32; считается, что они расположены между пользователем и экраном так, что ближайшая к пользователю имеет номер 0. При описании спрайт получает номер. Спрайт с указанным номером можно вывести в указанную логическую плоскость, в точку с указанными координатами. Этой цели служит предписание

--- 62. Вывести спрайт с номером NS в логическую плоскость NP цветом C так, чтобы левый верхний угол спрайта находился в точке (X,Y) XXX  
PUT SPRITE NP,(X,Y),C,NS

При выводе спрайта в некоторую логическую плоскость изображение любого спрайта, выведенного в нее ранее, исчезает. На этом основан эффект движения: чтобы "привести" спрайт по экрану слева направо, достаточно циклически вывести его в точки одной логической плоскости с возрастающей первой координатой. При движении спрайты частично перекрывают друг друга. При этом спрайты, расположенные "ближе" (в логических плоскостях с меньшим номером), закрывают те, которые расположены "далее"; любой спрайт закрывает рисунок на экране, полученный с помощью операторов PSET, LINE,

CIRCLE (а также DRAW).

#### Сведения о двоичном представлении чисел

Элементами последовательности, которая строится по таблице описания спрайта, являются двоичные числа; они имеют приставку &B перед строкой нулей и единиц. В записи двоичного числа допустимы только две цифры, а каждый следующий разряд в 2 раза больше предыдущего (есть разряды единиц, двоек, четверок и т.д.). Вычисляется значение числа, заданного в двоичном виде, аналогично десятичным числам:

325=3\*100+2\*10+5

&B01010101=0\*128+1\*64+0\*32+1\*16+0\*8+1\*4+0\*2+1 (=85)

&B10101010=1\*128+0\*64+1\*32+0\*16+1\*8+0\*4+1\*2+0 (=170)

Т.е. записи &B01010101 и 85 задают одно и то же число. Этим можно пользоваться при описании спрайта; числа соответствующей последовательности могут быть заданы как в двоичном, так и в десятичном виде. Так, нашу шахматную доску можно задать последовательностью

DATA 85,170,85,170,85,170,85,170

Иногда бывает полезно получить двоичное представление указанного числа. Для этого пользуются функцией BIN\$. При выполнении оператора PRINT BIN\$(85) будет напечатано 01010101.

А теперь попробуйте построить на листе бумаги в клетку спрайты, описанные в приведенной программе. Результат проверьте с помощью ЭВМ:

```
10 DATA 0,60,4,4,28,4,60,0
20 DATA 0,56,40,60,36,36,60,0
30 DATA 0,34,54,42,34,34,34,0
40 DEFSTR A:SCREEN 2,1:RESTORE 10
50 FOR I=1 TO 3:A=""
60 FOR J=1 TO 8:READ K:A=A+CHR$(K):NEXT
70 SPRITE$(I)=A
80 PUT SPRITE I,(I*10,0),,I
90 NEXT
100 LINEINPUT A
110 END
```

#### ЗАДАНИЯ.

1. Провести спрайт по экрану слева направо; сверху вниз.
2. Два одинаковых спрайта провести навстречу друг другу из разных концов экрана.
3. Создать рисунок из двух спрайтов и провести его по экрану.

4. Нарисовать два положения шагающего человека или животного и провести его по экрану с поочередной сменой положений.
5. Определить, можно ли вывести 10 спрайтов в одну горизонтальную линию.
6. Написать программу, управляющую движением спрайта с помощью стрелок.
7. Описать спрайт - указку с острием в левом верхнем углу.
8. В задаче с городами на острове изобразить спрайт в виде указки. В ответ на заданное название города ученик должен подвести указку к нужному городу и нажать "RETURN" (код 13). Если координаты указки при этом отличаются от координат города не более чем на 2, считать ответ верным.
9. Что измениться в программе, если острие указки будет в другом месте спрайта?
10. Изобразить с помощью спрайтов относительное движение Солнца, Земли, Луны.

#### 7.5. Построение сложных контуров

Для построения сложных контуров (не являющихся прямоугольниками и окружностями) используют предписание

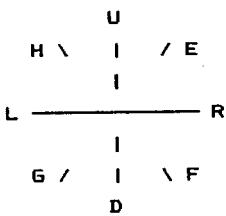
--- 63. Нарисовать сложный контур \*\*\*  
DRAW <строка>

При рисовании используется понятие текущей точки. Перед рисованием текущая точка расположена там, где закончился предыдущий графический оператор (например, если предыдущий оператор был LINE, то текущее положение после него – второй конец отрезка). Стока состоит из команд, записанных подряд (без разделителей). Существуют 4 группы команд. Одни из них (E, M) сводятся к перемещению текущей точки в новое положение; как правило, при этом рисуется траектория. Другие меняют режим выполнения следующей команды (B, N). Команда третьей группы позволяет изменять строку (составлять ее из других, сформированных ранее). Четвертые устанавливают некоторые параметры, которые действуют до их отмены.

Команды первой группы:

1. Переместиться на один шаг

При выполнении этой команды текущая точка перемещается в соседнюю точку экрана в одном из 8 направлений:



Пример. DRAW "E" – рисовать вверх-вправо на один шаг. Один шаг – это расстояние между соседними точками на экране.

2. Переместиться на несколько шагов в указанном направлении  
Пример. DRAW "R5U2" – рисовать 5 шагов вправо, 2 вверх

3. Переместиться в точку с указанными координатами (M)

Пример. DRAW "M100,120" – переместиться в точку (100,120). Координаты без знака считаются абсолютными, со знаком – относительными. DRAW "M+5,-2" – переместиться на 5 шагов правее и 2 шага ниже текущего положения.

Команды второй группы:

4. Следующую команду выполнить, не оставляя следа (B)

Пример. DRAW "R10BR5R10" – нарисовать две черточки по 10 шагов, с промежутком в 5 шагов.

5. Выполнить следующую команду и вернуться в исходное положение (N)

Пример. DRAW "NU5NR5ND5NL5" – нарисовать 4 линии (вверх, вправо, вниз, влево) по 5 шагов каждая из одной точки.

Команда третьей группы:

6. Включить подстроку (X)

Пример. При D\$="R5E2F2" запись "XD\$;XD\$" эквивалентна "R5E2F2R5E2F2". Знак ";" (точка с запятой) используется как разделитель между теми частями строки, которые допускают неоднозначное толкование.

Команды четвертой группы:

7. Задать цвет рисунка (C)

Пример. DRAW "C2U10" – рисовать цветом 2 (зеленым) 10 шагов вверх. Этот цвет будет действовать до следующей команды С.

8. Повернуть изображение против часовой стрелки на угол N\*90 (A)

Пример. DRAW "BM50,50XD\$;BM50,150A2XD\$;A0" – нарисовать два рисунка; один уголком вверх, другой – вниз (между рисунками повернуть изображение на 180 градусов). После второго рисунка вернуться к обычной ориентации.

9. Установить шаг рисования, в N/4 раз больше обычного (S)

Пример. DRAW "BM50,50XD\$;BM150,150S12XD\$;S4" – нарисовать два ри-

сунка; второй больше первого в 3 раза. После второго рисунка вернуться к обычному шагу рисования.

Если вместо константы в строке нужно использовать имя переменной, то перед ним нужно поставить знак "=", а после - ";". Например, при  $K=5$ ,  $L=7$  строка "M=K; , =L;" эквивалентна "M5,7"

#### ЗАДАНИЯ.

1. Нарисуйте карту СССР с границами союзных республик.
2. Сделайте из программы тренажер, задавая республику "наганием" границы и проверяя введенное название.

#### 7.6. Встроенные функции

Для некоторых наиболее часто используемых функций описания уже имеются в доступном BASICu виде. Такие функции называются встроенными; они имеют свои имена, и мы можем обращаться к ним по имени.

- 51. Получить в переменной Y значение ближайшего целого числа, не превышающего X ---  
Y=INT(X)
- 52. Получить в переменной Y значение X без дробной части ---  
Y=FIX(X)

Заметим, что INT(-3.5) равно -4, а FIX(-3.5) равно -3.

- 53. Получить в переменной Y значение абсолютной величины X---  
Y=ABS(X)
- 54. Получить в переменной Y значение квадратного корня из X---  
Y=SQR(X)
- 55. Получить в переменной Y значение синуса X ---  
Y=SIN(X)
- 56. Получить в переменной Y значение косинуса X ---  
Y=COS(X)
- 57. Получить в переменной Y значение тангенса X ---  
Y=TAN(X)
- 58. Получить в переменной Y значение арктангенса X ---  
Y=ATN(X)
- 59. Получить в переменной Y значение  $e(2,7..)$  в степени X ---  
Y=EXP(X)
- 60. Получить в переменной Y натуральный логарифм X ---

```

Y=LOG(X)
--- 61. Получить в Y значение функции SIGN X: -1 для отрицатель-
      ных, 0 для 0, +1 для положительных чисел ---
      Y=SGN(X)
--- 62. Получить в переменной Y остаток от деления A на B ---
      Y=A MOD B

```

**ЗАДАНИЕ.**

Имеется программа, рисующая график функции  $y=2x-5$ :

```

10 'график функции y=2x-5
20 SCREEN 2,2:K=8:S=.1:G=10:X0=128:Y0=100:OPEN "grp:" AS #1
30 LINE(0,Y0)-(255,Y0)
40 LINE(X0,10)-(X0,191) 'нарисовали оси
50 DEF FNY(X)=2*X-5:AF$="2*x-5" 'определен функцию
      и задали название
60 FOR X=-6 TO 6 STEP S
70   X1=X0+K*X
80   Y1=Y0-K*FNY(X)
90   PSET (X1,Y1)
100 NEXT 'поставили точки на графике
110 FOR X=-6 TO 6 STEP 5
120   X1=X0+K*X
130   LINE (X1,Y0-2)-(X1,Y0+2)
140   PRESET (X1-7,Y0+4):PRINT #1,STR$(X)
150 NEXT 'разметили и подписали ось X
160 FOR Y=-6 TO 6 STEP 5
170   Y1=Y0-K*Y
180   LINE (X0-2,Y1)-(X0+2,Y1)
190   IF Y<>0 THEN PRESET (X0+4,Y1-3):PRINT #1,STR$(Y)
200 NEXT 'разметили и подписали ось Y
210 PRESET (0,0):PRINT #1,"y=";AF$ 'вывели название
220 LINEINPUT A$
230 END

```

Используя конструкцию выбора и оператор описания функции, изменить ее таким образом, чтобы она рисовала поочередно графики следующих функций:

1.  $Y(X)=2XX-5$
2.  $Y(X)=INT(X)$
3.  $Y(X)=X-INT(X)$
4.  $Y(X)=(8XX-60)/(X-12)$
5.  $Y(X)=-X^2+8$
6.  $Y(X)=SQR(ABS(X))$

7.  $Y(X) = .1 \times (X+7) \times (X+2) \times (X-4)$   
8.  $Y(X) = \sin(X)$   
9.  $Y(X) = .5 \times \tan(.5 \times X)$   
10.  $Y(X) = X \times \sin(X)$

Одной из функций BASICa является датчик случайных чисел. Познакомимся с этой функцией подробнее.

Нередко возникает необходимость поработать с объектами, не связанными какой-либо закономерностью. Например, нам нужно изобразить ночное звездное небо. Можно вычислить координаты, соответствующие реальной картине; однако задача это трудоемкая и не является необходимой, если нужна только иллюстрация. Достаточно разбросать светлые точки по темному фону "как попало", случайным образом; при наличии какой-либо закономерности картина станет неправдоподобной. Нам понадобится предписание

--- 63. Получить в X случайное число от 0 до 1 ---  
 $X=RND(A)$

В зависимости от знака A результат получается различным. Мы обычно будем использовать предписание  $X=RND(1)$ , а до первого использования "разгонять" датчик вызовом  $X=RND(-TIME)$ .

Чтобы разбросать точки по экрану случайным образом, нам понадобятся случайные координаты X в интервале (0, 255) и Y в интервале (0, 191). 1000 таких точек можно поставить следующим образом:

```
FOR I=1 TO 1000
    X=RND(1)*255
    Y=RND(1)*191
    PSET (X,Y)
NEXT
```

или

```
FOR I=1 TO 1000
    X=RND(1)*10000 MOD 256
    Y=RND(1)*10000 MOD 192
    PSET (X,Y)
NEXT
```

Датчик случайных чисел очень часто используется в тренажерах для того, чтобы сделать работу с ними более разнообразной, чтобы вопросы предлагались в разной последовательности, меняющейся от сеанса к сеансу.

Пример.

10 'тренажер знаков химических элементов

```

20 DATA H,водород
30 DATA He,гелий
40 DATA Li,литий
50 DATA Be,бериллий
60 DATA B,бор
70 DATA C,углерод
80 DATA N,азот
90 DATA O,кислород
100 DATA F,фтор
110 DATA Ne,неон
120 DEFSTR A: KP=0: KN=0: X=RND(-TIME)
130 INTERVAL ON: ON INTERVAL=5*3600 GOSUB 190
140 PRINT "Укажи знак химического элемента"
150 GOSUB 1000 'выбрать элемент
160 LINEINPUT A
170 IF A=AZ THEN PRINT "Верно": KP=KP+1
    ELSE PRINT "Неверно": KN=KN+1
180 GOTO 150
190 'кончилось время
200 PRINT "Правильных ответов - ";KP
210 PRINT "Неправильных ответов - ";KN
220 END
1000 'выбрать элемент
1010 N=(RND(1)*1000 MOD 10)+1
1020 RESTORE 20
1030 FOR I=1 TO N
1040 READ AZ,AN
1050 NEXT
1060 PRINT AN;" - ";
1070 RETURN

```

#### ЗАДАНИЯ.

1. Дополнить приведенный здесь тренажер второй частью, где сообщается знак химического элемента, а запрашивается его название
2. Написать тренажер таблицы умножения, вывирящий сомножители случайным образом.
3. Написать тренажер таблицы вычитания от 1 до 20, причем так, чтобы разность всегда была положительной.
4. Построить нить ДНК из 100 нуклеотидов А, Т, Г, Ц случайным образом.
5. Получить и напечатать 10 случайных дат следующим образом: выбрать случайный номер месяца; для этого месяца выбрать случай-

- ное число; напечатать цифрами число и буквами название месяца.
6. В предыдущей задаче продолжать получение дат до того момента, пока две из них не совпадут. Сколько различных дат вы при этом получили?
  7. Постройте "график" случайной величины  $Y$  в интервале  $(0,191)$  на основании 256 обращений к датчику случайных чисел, увеличивая при каждом обращении  $X$  на 1. Каждую "следующую" точку соединить с "предыдущей" отрезком.
  8. Поместите 1000 точек случайным образом внутрь квадрата  $100 \times 100$ . Подсчитайте, сколько из них располагается внутри круга радиуса 50 с центром в центре квадрата (точка  $(x,y)$  располагается внутри круга радиуса  $r$  с центром в точке  $(a,b)$ , если  $(x-a)^2 + (y-b)^2 < r^2$ ). Намного ли отношение этого количества к 1000 отличается от отношения  $\pi/4$ ? Как этот результат можно объяснить геометрически?
  9. Расположите случайным образом на экране 20 окружностей случайного радиуса от 10 до 20.
  10. Нарисуйте 20 прямоугольников. Случайным образом выберите координаты левого верхнего угла, длину стороны и признак – закрашен прямоугольник или нет.

#### 7.7. Работа с экраном в алфавитно-цифровом режиме

Познакомимся с возможностью управлять положением курсора в алфавитно-цифровом режиме. Он может быть установлен в любую из строк от 0 до максимально возможной, а в строке – в любую из позиций от 0 до WIDTH (ширина экрана). По умолчанию ширина экрана – 39 символов. На экране 24 строки, но в режиме KEY ON 24-ая строка имеет специальное назначение – в нее выведены значения функциональных клавиш; в такую строку нельзя установить курсор.

- 77. Установить алфавитно-цифровой режим экрана ---  
SCREEN 0
- 78. Убрать с экрана значения функциональных клавиш ---  
KEY OFF
- 79. Вернуть значения функциональных клавиш ---  
KEY ON
- 80. Установить ширину экрана ( $1 \leq N \leq 40$ ) ---  
WIDTH N
- 81. Установить курсор в позицию X строки Y ---  
LOCATE X,Y
- 82. Получить в X номер позиции курсора в строке ---

X=POS(a)  
--- 83. Получить в Y номер строки, где стоит курсор ---  
Y=CSRLIN

**ЗАДАНИЯ.**

1. Тренажер таблицы умножения изменить так, чтобы за правильный ответ с первой попытки ученик получал 5 очков, со второй - 4, с третьей - 3; после трех неправильных ответов выводить подсказку. Набранное количество очков после каждого ответа выводить на экран в правый верхний угол.
2. По запросу пользователя рисовать квадрат, круг или треугольник. Список фигур выводит на экран, каждое название - с новой строки. Выбор производить подведением курсора к нужной строке и нажатием клавиши "RETURN".

**7.8. Обработка событий**

BASIC позволяет во время выполнения программы отслеживать некоторые внешние события, такие как нажатие клавиш CTRL/STOP или функциональных клавиш, истечение интервала времени, ошибки в устройстве или при вычислении. Как правило, для правильной обработки таких событий необходимо 2 предписания: включить реакцию на интересуемое событие и указать, как его обработать. По умолчанию в программе события такого sorta не обрабатываются. Кроме того, реакцию на указанные события можно выключать полностью или временно.

- 84. Включить реакцию на нажатие клавиш CTRL/STOP ---  
STOP ON
- 85. Включить реакцию на истечение интервала времени ---  
INTERVAL ON
- 86. Включить реакцию на нажатие функциональной клавиши FN ---  
KEY (N) ON
- 87. Выключить реакцию на нажатие клавиш CTRL/STOP ---  
STOP OFF
- 88. Выключить реакцию на истечение интервала времени ---  
INTERVAL OFF
- 89. Выключить реакцию на нажатие функциональной клавиши FN ---  
KEY (N) OFF
- 90. Временно отменить реакцию на нажатие клавиш CTRL/STOP ---  
STOP STOP
- 91. Временно отменить реакцию на истечение интервала ---

**INTERVAL STOP**

- 92. Временно отменить реакцию на нажатие функциональной клавиши FN ---  
KEY (N) STOP
  - 93. При нажатии CTRL/STOP выполнить подпрограмму с указанного оператора ---  
ON STOP GOSUB <номер>
  - 94. Через N секунд выполнить подпрограмму с указанного оператора ---  
ON INTERVAL=N\$60 GOSUB <номер>
  - 95. При нажатии функциональных клавиш выполнить подпрограммы с указанных операторов ---  
ON KEY GOSUB <номер1>, <номер2>, <номер3>...
  - 96. При ошибке уйти на обработку ошибки ---  
ON ERROR GOTO <номер оператора>
  - 97. Вернуться после обработки ошибки ---  
RESUME
- При этом можно вернуться
- на тот же оператор RESUME
  - на следующий RESUME NEXT
  - на указанный RESUME <номер>
- 98. Спровоцировать ошибку с номером N ---  
ERROR N

**Номера некоторых ошибок:**

- 4 - попытка прочитать больше элементов последовательности, чем в ней содержится
- 6 - переполнение; результат вычислений слишком велик
- 9 - выход за границы массива
- 11 - деление на 0
- 19 - ошибка в устройстве

**ЗАДАНИЯ.**

1. В тренажерах организовать подсказку по нажатию F1
2. Тренажеры заканчивать через 5 минут
3. По CTRL/STOP выдавать ученику количество верных и неверных ответов

**7.9. Музыкальные возможности**

ЭВМ "Янажа" обладает звуковым синтезатором, который позволяет воспроизводить музыкальные звуки. Для этого используют предписа-

ные

--- 99. Воспроизвести 3 последовательности нот на 3 голоса ---  
PLAY <строка1>, <строка2>, <строка3>

Строки 2 и 3 при этом могут отсутствовать.

Каждая из строк, определяющая один голос, должна быть сформирована по определенным правилам. В строке могут содержаться

1. Указания нот

|        |          |
|--------|----------|
| A - ля | E - ми   |
| B - си | F - фа   |
| C - до | G - соль |
| D - ре |          |

2. Указания длительности

L <число>

Число является знаменателем при указании длительности. Так, запись PLAY "L4AL8BL2CD" означает: сыграть четвертную ноту ля, восьмую си, половинные до и ре.

3. Указания октавы

O <номер октавы>

Все октавы пронумерованы от 1 до 8 так, что первой октаве на клавиатуре соответствует О4. Это значение используется по умолчанию.

4. Указания пауз

R <длительность>

По умолчанию используется R4.

5. Указание темпа

T <число>

Число может принимать значение от 32 до 255 и соответствует количеству воспроизводимых четвертных звуков в минуту. По умолчанию используется значение 120.

6. Указание громкости

V <число>

Число может принимать значение от 0 до 15; по умолчанию 8.

7. Указание тембра

S <число>

Используются значения 1, 4, 8, 10, 11, 12, 13, 14; по умолчанию

1. Этот параметр нельзя использовать одновременно с V.

Так, начало детской песенки "Веселые гуси" может быть запрограммировано так:

PLAY "FEDCGGRGR"

Значения параметров L, O, T, V, S продолжают действовать либо до

указания новых значений, либо до оператора подачи звукового сигнала

--- 100. Подать звуковой сигнал ---  
BEEP

**ЗАДАНИЯ.**

1. Воспроизвести гамму до мажор первой октавы.
2. Воспроизвести гамму в восходящем порядке с увеличением силы звука и в нисходящем - с уменьшением.
3. Воспроизвести гамму в восходящем и нисходящем порядке на 4 октавы.
4. Воспроизвести 4 куплета детской песенки: 3 куплета с возрастающей силой звука и четвертый - тихо.
5. Воспроизвести 3 куплета песенки, используя разный тембр.
6. Воспроизвести 4 раза гамму с увеличением темпа.
7. Воспроизвести "расходящуюся" гамму: первый голос - в восходящем порядке, второй - в нисходящем.
8. Воспроизвести все возможные сочетания из двух различных звуков одной октавы.
9. То же - для трезвучий.
10. Произведение состоит из 3 крупных частей. Каждая часть, в свою очередь, также построена по трехчастному принципу, но с повторениями: сначала дважды повторяется первая фраза, затем дважды - вторая и третья фраза вместе; третья фраза совпадает с первой. Предположив, что независимые фразы произведения описаны в строках A\$, B\$, C\$, D\$, напишите, какую структуру должна иметь программа, воспроизводящая это произведение.

Приложение 1. Система предписаний BASIC-машины

- 1. Прекратить выполнение текущего предписания \*\*\*  
CTRL/STOP
- 2. Напечатать значение выражения \*\*\*  
PRINT <выражение>
- 3. Запомнить значение выражения под указанным именем \*\*\*  
<имя>=<выражение>
- 4. Начать работу с новой программой \*\*\* |с  
NEW |р
- 5. Вывести текст программы на экран \*\*\* |е  
LIST |д
- 6. Вывести текст программы на печать --- |с  
(выполняется только при включенном принтере) |т  
LLIST |в
- 7. Выполнить программу \*\*\* |а  
RUN |с
- 8. Прекратить выполнение программы \*\*\* |р  
CTRL/STOP |а п
- 9. Удалить строку или группу строк --- |б р  
DELETE <номер строки> |о о  
или |т г  
DELETE <номер первой строки>—<номер последней строки> |ы р
- 10. Перенумеровать программу --- | а  
RENUM | н
- 11. Начать режим автоматической нумерации строк \*\*\* | м  
AUTO | а
- 12. Выйти из режима автоматической нумерации \*\*\* | м  
CTRL/C или CTRL/STOP | и
- 13. Дальнейшую информацию в строке считать пояснением \*\*\*  
<строка на BASICe> ? <информация>
- 14. Закончить выполнение программы \*\*\*  
END
- 15. Всю строку считать пояснением ---  
<номер> REM <информация>
- 16. Закончить выполнение программы и сообщить номер последне-  
го выполненного оператора ---  
STOP
- 17. Помыть экран ---  
CLS
- 18. Установить цвет рисунка/фона/бордюра ---  
COLOR C1,C2,C3

--- 19. Установить графический режим работы экрана \*\*\*  
SCREEN 2,2

--- 20. Нарисовать точку с координатами (X,Y) цветом С \*\*\*  
PSET(X,Y),C

--- 21. Нарисовать точку с координатами (X,Y) фоновым цветом ---  
PRESET(X,Y)

--- 22. Провести отрезок из точки (X1,Y1) в (X2,Y2) цветом С \*\*\*  
LINE(X1,Y1)-(X2,Y2),C

--- 23. Провести границу прямоугольника со сторонами, параллельными границам экрана, и противоположными углами в (X1,Y1) и (X2,Y2), цветом С ---  
LINE(X1,Y1)-(X2,Y2),C,B

--- 24. То же. Прямоугольник закрасить цветом С ---  
LINE(X1,Y1)-(X2,Y2),C,BF

--- 25. Нарисовать окружность с центром в точке (X,Y), радиуса R, цветом С \*\*\*  
CIRCLE(X,Y),R,C

--- 26. Нарисовать эллипс, у которого вертикальная полуось в N раз длиннее горизонтальной ---  
CIRCLE(X,Y),R,C,,,N

--- 27. Закрасить цветом С область, ограниченную замкнутым контуром цвета С, начиная от внутренней точки (X,Y) \*\*\*  
PAINT(X,Y),C

--- 28. Получить цвет точки с координатами (X,Y) в переменной С  
C=POINT(X,Y)

--- 29. Разрешить вывод текста в графическом режиме ---  
OPEN "GRP:" AS #1

--- 30. Вывести алфавитно-цифровую информацию на экран в графическом режиме ---  
PRINT #1,<выражение>

--- 31. Объявить целыми все переменные, имена которых начинаются с указанной буквы ---  
DEFINT <буква>

--- 32. Объявить строками все переменные, имена которых начинаются с указанной буквы \*\*\*  
DEFSTR <буква>

--- 33. Объединение строк A\$ и B\$ запомнить под именем C\$ \*\*\*  
C\$=A\$+B\$

--- 34. В переменной L получить длину строки A\$ \*\*\*  
L=LEN(A\$)

--- 35. В переменной A\$ получить N левых символов строки B\$ ---  
A\$=LEFT\$(B\$,N)

--- 36. В переменной A\$ получить N правых символов строки B\$ ---  
 A\$=RIGHT\$(B\$,N)  
 --- 37. В переменной A\$ получить значение L символов строки B\$, начиная с позиции N \*\*\*  
 A\$=MID\$(B\$,N,L)  
 --- 38. Набранную на клавиатуре до клавиши "RETURN" последовательность символов запомнить под именем A в виде, соответствующем типу переменной A \*\*\*  
 INPUT A  
 --- 39. Набранную на клавиатуре до клавиши "RETURN" последовательность символов запомнить в строке A\$ \*\*\*  
 LINEINPUT A\$  
 --- 40. N набранных символов запомнить в строке A\$ ---  
 A\$=INPUT\$(N)  
 --- 41. Объявить A массивом длины N \*\*\*  
 DIM A(N)  
 --- 42. Создать последовательность ---  
 <номер> DATA <элементы последовательности через запятую>  
 --- 43. Встать в начало последовательности ---  
 RESTORE <номер>  
 --- 44. Прочитать очередной элемент данных в переменную A ---  
 READ A  
 --- 45. ЦИКЛ МЕНЯЯ параметр от N1 до N2 с ШАГОМ N3  
     . выполнять  
     . . .  
     КОНЕЦ ЦИКЛА \*\*\*  
 FOR I=N1 TO N2 STEP N3  
     . . .  
 NEXT  
 --- 46. ЕСЛИ ... ТО ... ИНАЧЕ ... КОНЕЦ ЕСЛИ \*\*\*  
 IF ... THEN ... ELSE  
 --- 47. Выполнить предписание, описание которого начинается с указанного оператора \*\*\*  
 GOSUB <номер оператора>  
 --- 48. Конец предписания \*\*\*  
 RETURN  
 --- 49. Определить функцию Y(X) ---  
 DEF FNY(X)=<выражение>  
 --- 50. Переменной A присвоить значение функции Y(X) в точке X---  
 A=FNY(X)  
 --- 51. В переменной N получить, с какой позиции начинается подстрока B\$ в строке A\$ при просмотре с позиции P ---

```
N=INSTR(P,A$,B$)
--- 52. Создать строку A$ из N одинаковых символов S$ ---
A$=STRING$(N,S$)
--- 53. Создать строку A$ из N пробелов ---
A$=SPACE$(N)
--- 54. В переменной N получить числовое значение строки A$ ---
N=VAL(A$)
--- 55. Получить строку A$, представляющую запись числа N ---
A$=STR$(N)
--- 56. В переменной N получить значение кода символа A$ ---
N=ASC(A$)
--- 57. В переменной A$ получить символ с кодом N ---
A$=CHR$(N)
--- 58. Увеличить ресурс для работы со строками до N символов ---
CLEAR N
--- 59. Опросить клавиатуру XXX
A$=INKEY$
--- 60. Перейти на один из операторов в зависимости от N ---
ON <N> GOTO <номер1>, <номер2>, <номер3>...
--- 61. Описать спрайт с номером NS XXX
SPRITE$ (NS) = <строка>
--- 62. Вызвести спрайт с номером NS в логическую плоскость NP
цветом С так, чтобы левый верхний угол спрайта находился
в точке (X,Y) XXX
PUT SPRITE NP, (X,Y),C,NS
--- 63. Нарисовать сложный контур XXX
DRAW <строка>
--- 64. Получить в переменной Y значение ближайшего целого числа,
не превышающего X ---
Y=INT(X)
--- 65. Получить в переменной Y значение X без дробной части ---
Y=FIX(X)
--- 66. Получить в переменной Y значение абсолютно величины X---
Y=ABS(X)
--- 67. Получить в переменной Y значение квадратного корня из X---
Y=SQR(X)
--- 68. Получить в переменной Y значение синуса X ---
Y=SIN(X)
--- 69. Получите в переменной Y значение косинуса X ---
Y=COS(X)
--- 70. Получите в переменной Y значение тангенса X ---
Y=TAN(X)
```

- 71. Получить в переменной Y значение арктангенса X ---  
Y=ATN(X)
- 72. Получить в переменной Y значение  $e^{(2,7..)}$  в степени X ---  
Y=EXP(X)
- 73. Получить в переменной Y натуральный логарифм X ---  
Y=LOG(X)
- 74. Получить в Y значение функции SIGN X: -1 для отрицательных, 0 для 0, +1 для положительных чисел ---  
Y=SGN(X)
- 75. Получить в переменной Y остаток от деления A на B ---  
Y=A MOD B
- 76. Получить в X случайное число от 0 до 1 ---  
X=RND(a)
- 77. Установить алфавитно-цифровой режим экрана ---  
SCREEN 0
- 78. Убрать с экрана значения функциональных клавиш ---  
KEY OFF
- 79. Вернуть значения функциональных клавиш ---  
KEY ON
- 80. Установить ширину экрана (  $1 \leq N \leq 40$  ) ---  
WIDTH N
- 81. Установить курсор в позицию X строки Y ---  
LOCATE X,Y
- 82. Получить в X номер позиции курсора в строке ---  
X=POS(a)
- 83. Получить в Y номер строки, где стоит курсор ---  
Y=CSRLIN
- 84. Включить реакцию на нажатие клавиш CTRL/STOP ---  
STOP ON
- 85. Включить реакцию на истечение интервала времени ---  
INTERVAL ON
- 86. Включить реакцию на нажатие функциональной клавиши FN ---  
KEY (N) ON
- 87. Выключить реакцию на нажатие клавиш CTRL/STOP ---  
STOP OFF
- 88. Выключить реакцию на истечение интервала времени ---  
INTERVAL OFF
- 89. Выключить реакцию на нажатие функциональной клавиши FN ---  
KEY (N) OFF
- 90. Временно отменить реакцию на нажатие клавиш CTRL/STOP ---  
STOP STOP
- 91. Временно отменить реакцию на истечение интервала ---

INTERVAL STOP

--- 92. Временно отменить реакцию на нажатие функциональной клавиши FN ---  
KEY (N) STOP

--- 93. При нажатии CTRL/STOP выполнить подпрограмму с указанного оператора ---  
ON STOP GOSUB <номер>

--- 94. Через N секунд выполнить подпрограмму с указанного оператора ---  
ON INTERVAL=N\*60 GOSUB <номер>

--- 95. При нажатии функциональных клавиш выполнить подпрограммы с указанных операторов ---  
ON KEY GOSUB <номер1>, <номер2>, <номер3>...

--- 96. При ошибке уйти на обработку ошибки ---  
ON ERROR GOTO <номер оператора>

--- 97. Вернуться после обработки ошибки ---  
RESUME

--- 98. Спровоцировать ошибку с номером N ---  
ERROR N

--- 99. Воспроизвести 3 последовательности нот на 3 голоса ---  
PLAY <строка1>, <строка2>, <строка3>

--- 100. Подать звуковой сигнал ---  
BEEP

**Приложение 2. Система предписаний исполнителя ДИСПЛЕЙ**

|                                                                                 |                   |
|---------------------------------------------------------------------------------|-------------------|
| перейти к изображению заглавных букв                                            | ! CAPS            |
| перейти к русскому алфавиту                                                     | ! RUS             |
| перейти к изображению заглавных букв или верхнего регистра для клавиш с цифрами | !                 |
|                                                                                 | ! SHIFT           |
|                                                                                 | !                 |
| перестить курсор на 1 позицию вправо                                            | ! Стрелка вправо  |
| перестить курсор на 1 позицию влево                                             | ! Стрелка влево   |
| перестить курсор на 1 позицию вверх                                             | ! Стрелка вверх   |
| перестить курсор на 1 позицию вниз                                              | ! Стрелка вниз    |
| перестить курсор в левый верхний угол                                           | ! HOME            |
| перестить курсор на 8 позиций вправо                                            | ! TAB             |
| перестить курсор в начало следующего слова                                      | ! CTRL/F          |
| перестить курсор в начало предыдущего слова                                     | ! CTRL/B          |
| перестить курсор в конец строки                                                 | ! CTRL/N          |
|                                                                                 | !                 |
| стереть последний символ в строке                                               | ! BS              |
| стереть текущий символ                                                          | ! DEL             |
| стереть все от текущего символа до конца строки                                 | ! CTRL/E          |
| стереть текущую строку                                                          | ! CTRL/U          |
| стереть всю информацию с экрана                                                 | ! CLS             |
|                                                                                 | !                 |
| войти в режим вставки                                                           | ! INS             |
| выйти из режима вставки                                                         | ! INS или стрелка |

| Содержание                                                  | Стр. |
|-------------------------------------------------------------|------|
| Предисловие .....                                           | 3    |
| 1. Введение в программирование                              |      |
| 1.1. Знакомство с линейными программами .....               | 3    |
| 1.2. Знакомство с нелинейными программами .....             | 5    |
| 2. BASIC-машина как исполнитель                             |      |
| 2.1. Понятие BASIC-машины .....                             | 9    |
| 2.2. Работа BASIC-машины в режиме калькулятора .....        | 10   |
| 2.3. Средства для работы с программами .....                | 14   |
| 2.4. Графические возможности BASIC-машины .....             | 17   |
| 2.5. Типы переменных. Строки .....                          | 19   |
| 2.6. Ввод-вывод .....                                       | 22   |
| 3. Организация данных                                       |      |
| 3.1. Массивы .....                                          | 23   |
| 3.2. Последовательности .....                               | 26   |
| 4. Управляющие конструкции                                  |      |
| 4.1. Конструкция цикла .....                                | 28   |
| 4.2. Конструкция ветвления .....                            | 30   |
| 4.3. Цикл "пока" .....                                      | 33   |
| 5. Задачи .....                                             | 35   |
| 6. Разработка программ "сверху-вниз"                        |      |
| 6.1. Подпрограммы .....                                     | 41   |
| 6.2. Функции .....                                          | 44   |
| 7. Дополнительные возможности                               |      |
| 7.1. Работа со строками .....                               | 45   |
| 7.2. Дополнительные возможности ввода-вывода .....          | 47   |
| 7.3. Конструкция выбора .....                               | 48   |
| 7.4. Построение движущихся изображений .....                | 51   |
| 7.5. Построение сложных контуров .....                      | 54   |
| 7.6. Встроенные функции .....                               | 56   |
| 7.7. Работа с экраном в алфавитно-цифровом режиме .....     | 60   |
| 7.8. Обработка событий .....                                | 61   |
| 7.9. Музыкальные возможности .....                          | 62   |
| Приложение 1. Система предписаний BASIC-машины .....        | 65   |
| Приложение 2. Система предписаний исполнителя ДИСПЛЕЙ ..... | 71   |

Подготовлено к печати Научно-исследовательским вычислительным  
центром АН СССР  
7.07.88 г. Уч.-изд. л. 3,4. Усл.-печ. л. 4,5. Тираж 1000 экз.  
Заказ II65Р. Цена 70 к. Изд. № З18.  
Отпечатано с оригинала-макета на ротапринте в Отделе научно-тех-  
нической информации Научного центра биологических исследований  
АН СССР в Пущине