

УПРАВЛЯЮЩИЕ СТРУКТУРЫ.

Ветвление.

Вычислительный процесс называется разветвляющимся, когда он, в зависимости от выполнения определенных условий, реализуется по одному из заранее предусмотренных, возможных направлений.

Структура выбора в Бейсике реализуется с помощью оператора условного перехода, который позволяет выполнять входящий в него один из двух операторов, в зависимости от выполнения условия.

Оператор условного перехода.

ОБЩИЙ ВИД:

IF < УСЛОВИЕ > THEN < Действие 1 > ELSE < Действие 2 >

Оператор условного перехода работает следующим образом:

ЕСЛИ условие выполняется,

ТО выполняем Действие 1 и выходим из оператора.

ИНАЧЕ выполняем Действие 2 и выходим.

Это полная форма оператора.

Например: Даны два числа. Вывести на печать квадрат большего из них.

РЕШЕНИЕ.

CLS

INPUT "A=";A

INPUT "B=";B

IF A>B THEN PRINT A^2 ELSE PRINT B^2

END

Оператор условного перехода можно записать в блочной форме:

IF условие 1 THEN

 БЛОК ДЕЙСТВИЙ 1

ELSEIF условие 2 THEN

 БЛОК ДЕЙСТВИЙ 2

.....

ELSE

 БЛОК ДЕЙСВИЙ n

END IF

В этом случае программа из предыдущего примера запишется так:

CLS

INPUT "A=";A

INPUT "B=";B

IF A>B THEN

 PRINT A^2

ELSE

 PRINT B^2

END IF

END

Оператор IF может не иметь конструкции ELSE. Такая форма оператора называется сокращенной.

IF <input type="checkbox"/> < УСЛОВИЕ > <input type="checkbox"/> THEN <input type="checkbox"/> < Действие >

Этот оператор выполняется следующим образом:

ЕСЛИ условие выполняется,

ТО выполняем Действие 1 и выходим из оператора.

ИНАЧЕ выходим из команды.

В блочной форме:

IF условие THEN

 БЛОК ДЕЙСТВИЙ

END IF

Пример: Дано число. Заменить его значение на корень арифметический из этого числа, если оно больше или равно нулю.

CLS

INPUT "A="; A

IF A>=0 THEN PRINT A=SQR(A)

PRINT "A=";a

END

Простые и составные условия.

В условных операторах и в операторе цикла WHILE в качестве условия можно использовать отношения равенства и неравенства: $a>0$; $c\leq 0$ и т.д. Такие условия называют простыми.

Из простых условий в Бейсике разрешается строить более сложные. Для этих построений существуют определенные правила.

Прежде всего требуется, чтобы все отношения, которые используются при построении, заключались в скобки: $(a=b+1)$, $(c>=0)$, $(s>t)$ и т.д.

Далее, пусть A и B — некоторые условия. Рассмотрим связки, которыми они могут быть соединены.

1. **A and B** (чит. A и B) — это условие соблюдается тогда и только тогда, когда соблюдаются оба условия.

2. **A or B** (чит. A или B) — это условие соблюдается тогда и только тогда, когда соблюдается хотя бы одно из условий A и B.

3. **not A** (чит. не A) — это условие соблюдается тогда и только тогда, когда условие A не соблюдается.

При выполнении логических операций необходимо соблюдать следующий приоритет:

1) NOT; 2) AND; 3) OR.

Решение задач.

Задача 1.

Определить, принадлежит ли число X отрезку [3, 7].

РЕШЕНИЕ.

CLS

INPUT "X=";X

IF (X<3)or (X>7) THEN

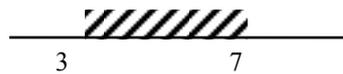
PRINT "Не принадлежит "

ELSE

PRINT "Принадлежит "

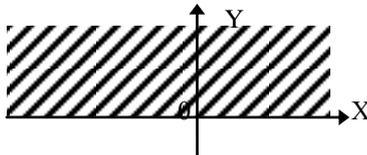
END IF

END



Задача 2.

Записать условия, которым удовлетворяют точки верхней полуплоскости системы координат:



На графике видно, что заштрихованы все точки, координаты Y у которых неотрицательны, следовательно Ответ: $Y\geq 0$.

Задача 3.

Дано X. Вычислить Y, если

X^3 , при $-1\leq X\leq 1$

$Y=7$, в противном случае

Примечание: При вычислении Y нужно проверить знак X. Сложное неравенство разобьем на 2 простых ($X\geq -1$) и ($X\leq 1$).



Т.к. эти условия должны выполняться одновременно, то мы должны связать их связкой **and**.

Решение.

CLS

INPUT "X=";X

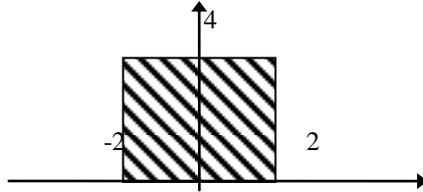
IF (X>=-1) and (X<=1) THEN Y=X^3 ELSE Y=7

PRINT "Y=";Y

END

Задача 4.

Для заданной координатами X и Y точки проверить принадлежит ли она квадрату на рисунке:



Решение:

```
CLS
INPUT "X=";X
INPUT "Y=";Y
IF (ABS(X)>=2) AND (Y<=4) AND (Y>=0) THEN
  PRINT " ТОЧКА ПРИНАДЛЕЖИТ"
ELSE
  PRINT "ТОЧКА НЕ ПРИНАДЛЕЖИТ"
END IF
END
```

Задача 5.

Вычислить значение выражения $y = \frac{\sqrt{x+1}}{x-2}$. Значение x ввести клавиатуры.

Решение.

```
CLS
INPUT "X=";X
IF (x+1 < 0) AND (X=2) THEN
  PRINT "При данном значении x функция не определена"
ELSE
  Y=SQR (X+1) / (X-2)
  PRINT "Значение функции равно "; Y
END IF
END
```

Безусловный переход

Безусловный переход оператором goto предписывает программе свернуть с линейного пути и, беспрекословно повинаясь, перейти к метке, расположенной в любом месте программы. В качестве метки используются натуральные числа (можно с двоеточием после них). Метка указывается только в начале строки (т. е. если в строке программы несколько операторов, то нельзя ставить метку, например, перед вторым). Приведем пример программы, рисующей на экране три символа звездочки по диагонали:

```
CLS
?"?*" ?"??*"
```

Символ ? в данном случае означает пробел.

Добавим теперь метку 1 рядом со второй строкой и оператор goto в конце программы. Получаем:

```
CLS
1: ?"*"
?"?*"
9 "77* "
GOTO 1
```

Казалось бы, изменения незначительные, но запустите программу и посмотрите что происходит! Вы в панике? Вы пытаетесь прекратить действие программы, но ничего не

получается? Программа зациклилась! Что делать? Спокойно нажмите клавишу <Ctrl> и, не отпуская ее, клавишу <Break>. Это называется прервать выполнение программы. Переведите дух и давайте шаг за шагом разберемся, что же произошло.

Программа очищает экран, затем рисует первую звездочку, под ней с отступом в один пробел — вторую, потом ниже третью, а вот затем получает команду `goto 1`. Это приказ, которого невозможно послушаться. Программа ищет метку, находит ее у команды рисования первой звездочки и начинает снова последовательно выполнять команды рисования — первую, вторую, третью и — снова приказ перехода, и так — до бесконечности (вернее, до прерывания программы). А поскольку оператор `print`, не снабженный точкой с запятой, после выполнения всегда переводит курсор на следующую строку, то и получается эффект движения.

Посмотрите на текст следующей программы:

```
x=0  
1: x=x+1: print x  
GOTO 1
```

Как вы думаете, при каком значении `x` программа закончит свою работу? Если у вас достаточно быстродействующий компьютер, можете рискнуть проверить свое предположение на практике.