ЦИКЛИЧЕСКИЕ СТРУКТУРЫ.

. Понятие цикла.

Для того, чтобы наглядно увидеть необходимость в использовании в программах циклических структур, рассмотрим пример.

Задача .

Вывести на экран квадраты первых N натуральных чисел.

Разберем решение задачи, например, для N=5.

```
CLS
PRINT 1^2
PRINT 2^2
PRINT 3^2
PRINT 4^2
PRINT 5^2
END
```

Для небольших значений N написать и набрать эту программу не составит труда — можно воспользоваться копированием и вставкой строк, а потом только исправить числа. Но представьте себе в какого “монстра” превратится эта программа, если нам придется писать ее для N=1000 !

Как же быть? Оказывается все очень просто. Внимательно рассмотрев программу, мы увидим, что 5 раз повторяется один и тот же оператор, отличие только в том, что каждый раз вычисляется квадрат числа на единицу больше предыдущего.

Таким образом при многократном повторении одних и тех же действий речь может идти о цикле. А что такое цикл? Когда Вы говорите о человеке — “зациклился”? Когда он много-много раз говорит об одном и том же или делает одно и то же.

Итак,

ЦИКЛ — это многократное повторение одних и тех же действий.

Цикл в общем виде должен содержать:

Установку начальных значений переменных цикла.

Тело цикла — действия, повторяемые многократно.

Изменение переменных цикла.

Проверку условия окончания (продолжения) цикла.

В случае выполнения условия продолжения цикла, переход к пункту 2.

В данном пособии мы рассмотрим две циклические структуры — счетный цикл FOR ... NEXT и цикл “пока” WHILE ... WEND.

Счетный цикл FOR...NEXT (Цикл с параметром)

Общий вид:

```
FOR счетчик = нач.знач TO кон.значение STEP шаг
        ТЕЛО ЦИКЛА
NEXT счетчик
```

где счетчик — это имя переменной, изменяющейся в цикле;

нач.значение и кон.значение — числовые константы;

шаг изменения — числовое значение, определяющее величину изменения переменной цикла при последующем исполнении.

Служебное слово FOR называют заголовком цикла, NEXT — концом цикла.

Оператор FOR...NEXT выполняется следующим образом:

1) Переменной цикла присваивается начальное значение.

2) Проверка допустимости значения переменной цикла, т.е. не превышает ли оно конечного Если не превышает, то переход к п.3, иначе — конец цикла и переход на оператор, следующий за оператором NEXT.

3) Выполняются операторы тела цикла.

4) Переменная цикла получает новое значение, измененное на величину шага.

значения.

Итак, как же должна выглядеть наша программа?

Так как число n заранее неизвестно, используем ввод с клавиатуры INPUT.

```
CLS                      :REM  Очистка экрана
INPUT “n=“;N             :REM  Ввод числа n пользователем
FOR I=1 TO n STEP 1      :REM   Организация цикла
  PRINT I^2              :REM  Тело цикла
NEXT I                   :REM Конец цикла
END
```

ПОЯСНЕНИЕ: I — переменная цикла, которое будет изменяться от 1 до n. Тело цикла — вывод на печать квадрата переменной цикла.

Вложенные циклы.

В программе могут быть циклы, имеющие в теле цикла другие циклы. В таких случаях нужно помнить:

1) Переменные циклов должны иметь разные имена.

2) Сначала должен быть закрыт внутренний цикл, а потом внешний.

Например, нижеприведенная программа выводит на экран таблицу умножения:

```
FOR I=1 TO 9 STEP 1
.  FOR J=1 TO 9 STEP 1
.  . PRINT I; "*";J;"=";I*J
.  NEXT J: PRINT
NEXT I
```

Решение задач

Задача 1.

Подсчитать N! (факториал числа N), то есть произведение первых N натуральных чисел. Число N ввести с клавиатуры.

ПОЯСНЕНИЕ. Для подсчета нам необходимо завести переменную памяти, в которой будут "накапливаться" произведения. Пока она не задействована, ей присваивают значение 1. Переменная цикла будет последовательно принимать значения натуральных чисел от 1 до N. В теле цикла ячейке-"копилке" будут присваиваться произведения ее старого значения на переменную цикла.

```
CLS
INPUT "Введите натуральное число: ";N
P=1
FOR I=1 TO N STEP 1
  P=P*I
NEXT I
PRINT  N; "! = "; P
END
```

Задача 2.

Подсчитать сумму дробей вида $\frac{1}{n+1}$ от 1 до n (n ввести с клавиатуры).

ПРИМЕЧАНИЕ. В данной задаче нам тоже понадобится переменная-"копилка", в которую мы будем складывать дроби. Первоначальное значение переменной должно быть равно 0. Округление произвести с помощью функции USING.

```
CLS
INPUT ""Введите натуральное число "; N
S=0
FOR I=1 TO N STEP 1
  S=S+ 1/(I+1)
NEXT I
PRINT " Сумма дробей вида 1/(n+1) для n от 1 до ";N; "равна ";
PRINT  USING"##.##";S
END
```

Результат для N=4:

```
Введите натуральное число? 4↵
Сумма дробей вида 1/(n+1) для n от 1 до 4 равна 1.28
```

Задача 3.

Написать программу переворачивающую слово.

Решение.

```
CLS
INPUT ""Введите слово: ";B$
A$=""
FOR I= LEN(B$) TO 1 STEP -1
  A$=A$+ MID$(B$, I, 1)
NEXT I
PRINT "Перевернутое слово —  "; A$
END
```

Цикл “ ПОКА ” — WHILE ... WEND (Цикл с предусловием)

Если в счетном цикле FOR ... NEXT число повторений фиксировано, то цикл WHILE ... WEND используют в том случае, если количество повторений заранее неизвестно, но известно условие, определяющее конец работы цикла.

ОБЩИЙ ВИД:

```
WHILE ⌴ УСЛОВИЕ
  Т Е Л О  Ц И К Л А
WEND
```

Тело цикла выполняется до тех пор, пока условие истинно. Если условие сразу не выполняется, то тело цикла не выполняется ни разу.

Например:

```
INPUT X
WHILE X< 100
  X=X ^ 2
  PRINT X
WEND
```

Пояснение: Ввели с клавиатуры переменную X. Пока X<100, возводим ее в квадрат и печатаем.

Пусть X=3. Получим:

3<100? да
X=9
9<100? да
X=81
81<100? да
X=6561
6561<100 нет

```
? 3↵
9

81

6561

ок
```

С другой стороны цикл может никогда не завершиться, то есть произойдет зацикливание. Например, если с клавиатуры будет введено число, меньше единицы, то переменная X будет, уменьшаясь, стремиться к нулю, а значит условие будет всегда выполняться.

Задача.

Вычислить квадратный корень числа, введеного с клавиатуры.

ПРИМЕЧАНИЕ: квадратный корень вычисляется только из неотрицательного числа, поэтому необходимо организовать в программе проверку ввода.

```
 CLS
INPUT “Введите число ”; N
WHILE N <0
PRINT “Ошибка! Повторите ввод!“
INPUT “Введите число ”; N
WEND
PRINT “Корень этого числа равен  “; SQR(N)
END
```