# 8 (базовый уровень, время – 3 мин)

**Тема**: Анализ программы.

**Что нужно знать**:

* основные конструкции языка программирования:
	+ объявление переменных
	+ оператор присваивания
	+ оператор вывода
	+ циклы
* уметь выполнять ручную прокрутку программы
* уметь выделять переменную цикла, от изменения которой зависит количество шагов цикла
* уметь определять количество шагов цикла
* уметь определять переменную, которая выводится на экран
* формулу для вычисления -ого элемента арифметической прогрессии:



* формулу для вычисления суммы первых  членов арифметической прогрессии:



где  – -ый элемент последовательности,  – шаг (разность) последовательности

### Пример задания:

**Р-02.** *При каком наибольшем введенном числе d после выполнения программы будет напечатано 55?*

**var n, s, d: integer;**

**begin**

 **readln(d);**

 **n := 0;**

 **s := 0;**

 **while s <= 365 do begin**

 **s := s + d;**

 **n := n + 5**

 **end;**

 **write(n)**

**end.**

 **Решение:**

1. из программы видно, что начальные значения переменных **s** и **n** равны нулю
2. шаг изменения переменной **n** равен 5, а шаг изменения переменной **s** равен неизвестному значению **d**
3. для того, чтобы значение **n** стало равно 55, нужно увеличить его на 5 (с нуля) ровно 11 раз, поэтому цикл выполнится ровно 11 раз
4. следовательно, s увеличится на d тоже 11 раз и станет равно 0 + 11·d = 11·d
5. чтобы цикл остановился на 11-м шаге, нужно выполнить условие 11·d > 365, при этом он не должен остановиться на 10-м шаге, то есть, 10·d ≤ 365, поэтому получаем два неравенства:





1. в итоге значение d – целое число – ограничено отрезком [34; 36], наибольшее из подходящих чисел равно 36
2. Ответ: 36.

### Ещё пример задания:

**Р-02.** *Запишите число, которое будет напечатано в результате выполнения программы.*

**var s, n: integer;**

**begin**

 **s := 33;**

 **n := 1;**

 **while s > 0 do begin**

 **s := s – 7;**

 **n := n \* 3**

 **end;**

 **writeln(n)**

**end.**

 **Решение:**

1. из программы видно, что начальные значения переменных **s** и **n** равны соответственно 33 и 1
2. цикл заканчивается, когда нарушается условие **s > 0**, то есть количество шагов цикла определяется изменением переменной **s**
3. после окончания цикла выводится значение переменной **n**
4. таким образом, задача сводится к тому, чтобы определить число шагов цикла, необходимое для того, чтобы значение **s** стало меньше или равно 0
5. с каждым шагом цикла значение **s** уменьшается на 7, а значение **n** увеличивается в 3 раза, так что **n=3k**, где **k** – это число шагов цикла
6. поскольку **s** уменьшается на 7, конечное значение **s** должно быть равно **33-7\*k** , причём первое значение, меньшее или равное 0, достигается при **k=5** (и **s=33–7\*5=-2**)
7. тогда **n=3k=35=243**
8. Ответ: 243.

### Ещё пример задания:

**Р-01.** *Определите, что будет напечатано в результате работы следующего фрагмента программы:*

**var k, s: integer;**

**begin**

 **s:=0;**

 **k:=0;**

 **while s < 1024 do begin**

 **s:=s+10;**

 **k:=k+1;**

 **end;**

 **write(k);**

**end.**

 **Решение:**

1. из программы видно, что начальные значения переменных **k** и **s** равны нулю
2. цикл заканчивается, когда нарушается условие **s < 1024**, то есть количество шагов цикла определяется изменением переменной **s**
3. после окончания цикла выводится значение переменной **k**
4. таким образом, задача сводится к тому, чтобы определить число шагов цикла, необходимое для того, чтобы значение **s** стало не меньше 1024
5. с каждым шагом цикла значение **s** увеличивается на 10, а значение **k** – на единицу, так что фактически **k** – это счётчик шагов цикла
6. поскольку **s** увеличивается на 10, конечное значение **s** должно быть кратно 10, то есть это 1030 > 1024
7. для достижения этого значения переменную **s** нужно 103 раза увеличить на 10, поэтому цикл выполнится 103 раза
8. так как **k** – это счётчик шагов цикла, конечное значение **k** будет равно 103
9. Ответ: 103.

|  |
| --- |
| **Возможные ловушки и проблемы**:* можно перепутать переменную, которая выводится на экран (внимательно смотрим на оператор вывода)
 |

### Ещё пример задания:

**Р-00.** *Определите, что будет напечатано в результате работы следующего фрагмента программы:*

**var k, s: integer;**

**begin**

 **k:=5;**

 **s:=2;**

 **while k < 120 do begin**

 **s:=s+k;**

 **k:=k+2;**

 **end;**

 **write(s);**

**end.**

 **Решение:**

1. начальные значения переменных **k** и **s** равны соответственно 5 и 2
2. цикл заканчивается, когда нарушается условие **k** **<** **120**, то есть количество шагов цикла определяется изменением переменной **k**
3. после окончания цикла выводится значение переменной **s**
4. с каждым шагом цикла значение **s** увеличивается на **k**, а затем значение **k** – на 2, так что к начальному значению **s** добавляется сумма членов арифметической прогрессии с начальным значением  и разностью 
5. поскольку начальное значение **k** равно 5 и с каждым шагом оно увеличивается на 2, переменная **k** принимает последовательно нечётные значения: 5, 7, 9, …
6. цикл заканчивается, когда значение **k** становится не меньше 120; поскольку **k** всегда нечётное, конечное значение **k** равно 121
7. поскольку значение **k** увеличивается после того, как увеличивается значение s, значение 121 уже не входит в сумму, то есть последний элемент последовательности :



1. количество  членов последовательности, которые входят в сумму, можно вычислить: чтобы из 5 получить 119 нужно 57 раз добавить шаг 2, поэтому общее число элементов последовательности равно (на один больше)
2. теперь используем формулу для вычисления суммы членов арифметической прогрессии:



1. к этой сумме нужно добавить начальное значение переменной **s**, равное 2:



1. Ответ: 3598.

|  |
| --- |
| **Возможные ловушки и проблемы**:* попытка делать ручную трассировку, скорее всего, приведет к вычислительной ошибке, потому что число шагов слишком велико
* легко забыть, что начальные значения переменных **s** и **k** не равны нулю
* нужно помнить, что количество членов арифметической прогрессии на 1 больше, чем количество шагов, которые необходимы для перехода от первого значения к последнему
 |