

## 20 (повышенный уровень, время – 5 мин)

### Пример задания:

**P-08.** Получив на вход число  $x$ , этот алгоритм печатает число  $M$ . Известно, что  $x > 100$ . Укажите наименьшее такое (т.е. большее 100) число  $x$ , при вводе которого алгоритм печатает 26.

```
var x, L, M: integer;
begin
  readln(x);
  L := x;
  M := 65;
  if L mod 2 = 0 then M := 52;
  while L <> M do { * }
    if L > M then { * }
      L := L - M { * }
    else { * }
      M := M - L; { * }
  writeln(M);
end.
```

### Решение:

- 1) видим, что в последней строке выводится на экран переменная **M**
- 2) ключевой момент решения: нужно узнать в строках программы, отмеченных знаком \* в комментариях, АЛГОРИТМ ЕВКЛИДА для вычисления наибольшего общего делителя (НОД) чисел, записанный в переменные **M** и **L**
- 3) введённое значение **x** записывается в переменную **L** и участвует в поиске НОД
- 4) в переменную **M** до начала цикла записывается 65, но если было введено чётное ( $L \bmod 2 = 0$ ) значение **x** (оно же **L**), значение **M** заменяется на 52
- 5) сначала предположим, что замены не было, и в **M** осталось значение 65; поскольку по условию алгоритм печатает 26, тогда получается, что  $\text{НОД}(x, 65) = 26$ ; этого явно не может быть, потому что 65 не делится на 26
- 6) делаем вывод, что введено чётное значение **x** и произошла замена **M** на 52
- 7) итак, нужно найти чётное число **x**, большее 100, такое, что  $\text{НОД}(x, 52) = 26$
- 8) первое число, большее 100, которое делится на 26 – это 104, но оно не подходит, потому что делится ещё и на 52, так что  $\text{НОД}(x, 52) = 52$
- 9) поэтому берём следующее число, которое делится на 26:  $104 + 26 = 130$
- 10) Ответ: **130**.

### Ещё пример задания:

**P-07.** Ниже записан алгоритм. Укажите минимальное число , при вводе которого алгоритм печатает сначала 26391.

```
var x, K, A, B: integer;
begin
  readln(x);
  K:=1; A:=0; B:=0;
  while x>0 do begin
    if (x mod 10) mod 2 = 0 then
      A:=A*10+x mod 10
    else begin
      K:=K*10;
      B:=B*10 + x mod 10
    end;
    x:=x div 10
  end;
  A:=A*K + B;
  writeln(A)
end.
```

### Решение:

- 1) видим, что в последней строке выводится на экран переменная **A**, которая вычисляется в предыдущей строке по формуле **A:=A\*K+B**
- 2) определим, сколько раз выполняется цикл **while**; условие его продолжения –  $x > 0$ , с переменной **x** выполняется единственная операция – деление на 10 нацело:  

```
while x>0 do begin
    ...
    x:=x div 10
end;
```

отсюда делаем вывод, что цикл выполняется столько раз, сколько цифр в десятичной записи введённого числа **x**
- 3) теперь посмотрим, что происходит внутри цикла: выбор варианта действия зависит от выполнения условия  
 $(x \bmod 10) \bmod 2 = 0$   
здесь  $x \bmod 10$  – это последняя цифра  $x$ , в этом условии проверяется её чётность (делимость на 2)
- 4) итак, если последняя цифра числа чётная, выполняется оператор  
**A:=A\*10+x mod 10**  
то есть, предыдущее значение **A** умножается на 10 и к результату добавляется последняя цифра **x**; таким образом переменная **A** составляется из чётных цифр числа **x**, причём в обратном порядке, потому что новая цифра добавляется в конец числа, а предыдущие (которые были ближе к концу в записи числа **x**) продвигаются влево, в старшие разряды
- 5) теперь смотрим, как строится **B**: здесь всё то же самое, только нечётные цифры собираются в обратном порядке; например, если исходное число было 12345, после окончания цикла мы получим **A=42** и **B=531**
- 6) но есть ещё переменная **K**, её начальное значение – 1, и с каждой найденной нечётной цифрой она умножается на 10, то есть **K=10** в степени, равной количеству нечётных цифр!  
для числа 12345 получим  $K=1000$
- 7) в предпоследней строке по формуле **A:=A\*K+B** собирается итоговое значение **A**; для нашего примера (12345) мы получим **A:=42\*1000+531=42531**, то есть **K** служит для того, чтобы сдвинуть комбинацию чётных цифр в начало числа
- 8) итак, нам задано число 26391, поэтому в искомом числе есть чётные цифры (по порядку, слева направо) {6, 2} и нечётные цифры {1, 9, 3} (тоже по порядку)
- 9) как же расположить эти цифры, чтобы получилось минимальное число? для этого сравниваем первые числа в списках чётных и нечётных чисел, и записываем в ответ меньшее из них; эту операцию повторяем, пока числа в обоих списках не кончатся; помним, что менять порядок чётных и нечётных чисел нельзя!
- 10) в данном случае получается {1, 6, 2, 9, 3} = 16293.
- 11) Ответ: **16293**.

### Пример задания:

**P-06.** Ниже записан алгоритм. Укажите наименьшее пятизначное число , при вводе которого алгоритм печатает сначала 4, а потом 2.

```
var x, y, a, b: longint;
begin
  a := 0;
  b := 0;
  readln(x);
  while x > 0 do begin
    y := x mod 10;
    if y > 3 then a := a + 1;
    if y < 8 then b := b + 1;
```

```

    x := x div 10
end;
writeln(a);
writeln(b)
end.

```

**Решение:**

- 12) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 13) перед началом цикла переменные **a** и **b** обнуляются
- 14) на каждом шаге цикла при выполнении некоторых условий переменные **a** и **b** увеличиваются на 1, то есть представляют собой счётчики
- 15) увеличение переменных зависит от значения  $y = x \bmod 10$ , то есть от последней цифры числа
- 16) если последняя цифра числа больше 3, увеличивается счётчик **a**, если меньше 8 – счётчик **b**;
- 17) в конце каждого шага цикла операция  $x := x \text{ div } 10$  отсекает последнюю цифру в десятичной записи числа
- 18) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 19) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр, больших 3, в десятичной записи числа, а в переменной **b** – количество цифр, меньших 8
- 20) если было выведено 4 и 2, то в числе 4 цифры больше 3 и 2 цифры меньше 8
- 21) так как число пятизначное, есть  $4 + 2 - 5 =$  одна цифра, которая больше 3 и меньше 8 одновременно; она должна быть минимальной, поэтому эта цифра **4**
- 22) для того чтобы число было минимальным, ещё одна цифра должна быть минимальной и меньшей 3 – это старшая **1**, и три цифры минимальные из цифр, больших или равных 8, то есть **три цифры 8**
- 23) ответ: **14888**.

**Ещё пример задания:**

**P-05.** Ниже записан алгоритм. Сколько существует таких чисел , при вводе которых алгоритм печатает сначала 2, а потом 12?

```

var x, a, b: integer;
begin
  readln(x);
  a:=0; b:=0;
  while x>0 do begin
    a:=a + 1;
    b:=b + (x mod 10);
    x:=x div 10;
  end;
  writeln(a); write(b);
end.

```

**Решение:**

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменные **a** и **b** обнуляются
- 3) на каждом шаге цикла при выполнении некоторого условия переменная **a** увеличивается на 1, а **b** увеличивается на  $x \bmod 10$ , то есть, на остаток от деления **x** на 10 – это последняя цифра десятичной записи числа **x**
- 4) в конце каждого шага цикла операция  $x := x \text{ div } 10$  отсекает последнюю цифру в десятичной записи числа

- 5) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 6) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр в десятичной записи числа, а в переменной **b** – их сумма
- 7) если было выведено 2 и 12, то в числе 2 цифры, и их сумма равна 12; таким образом, нам нужно найти все двузначные числа, в котором сумма значений цифр равна 12
- 8) число 12 может быть разложено на два слагаемых, меньших 10, как  $12 = 3 + 9 = 4 + 8 = 5 + 7 = 6 + 6 = 7 + 5 = 8 + 4 = 9 + 3$ , нам подходят числа 39, 48, 57, 66, 75, 84 и 93
- 9) всего таких чисел - 7
- 10) ответ: **7**.

### Ещё пример задания:

**P-04.** Ниже записан алгоритм. Укажите наименьшее из таких чисел , при вводе которых алгоритм печатает сначала 2, а потом 15.

```
var x, a, b: integer;
begin
  readln(x);
  a:=0; b:=1;
  while x>0 do begin
    a:=a+1;
    b:=b*(x mod 10);
    x:= x div 10
  end;
  writeln(a); write(b)
end.
```

#### Решение:

- 1) видим, что в последней строке выводятся на экран переменные **a** и **b**, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла переменная **a** обнуляется, а переменная **b** равна 1
- 3) на каждом шаге цикла при выполнении некоторого условия переменная **a** увеличивается на 1, а **b** умножается на  $x \bmod 10$ , то есть, на остаток от деления **x** на 10 – это последняя цифра десятичной записи числа **x**
- 4) в конце каждого шага цикла операция  $x := x \text{ div } 10$  отсекает последнюю цифру в десятичной записи числа
- 5) цикл заканчивается, когда перестаёт выполняться условие  $x > 0$ , то есть, когда все цифры исходного числа отброшены
- 6) таким образом, делаем вывод: после завершения цикла в переменной **a** находится количество цифр в десятичной записи числа, а в переменной **b** – их произведение
- 7) если было выведено 2 и 15, то в числа 2 цифры, и их произведение равно 15; таким образом, нам нужно найти минимальное двузначное число, в котором произведение значений цифр равно 15
- 8) поскольку число 15 может быть разложено на два сомножителя, меньших 10, только как 3·5, минимальное подходящее число – 35.
- 9) ответ: **35**.

### Ещё пример задания:

**P-03.** Ниже записан алгоритм. Укажите наименьшее из таких чисел , при вводе которых алгоритм печатает сначала 3, а потом 2.

```
var x, a, b, c: integer;
begin
  readln(x);
  a:= 0; b:= 0;
```

```

while x > 0 do begin
  c:= x mod 2;
  if c = 0 then a:= a + 1
  else b:= b + 1;
  x:= x div 10;
end;
writeln(a);
writeln(b);
end.

```

**Решение:**

- 1) видим, что в последний строках выводятся на экран переменные a и b, поэтому сначала нужно определить, что они обозначают в программе
- 2) перед началом цикла обе переменные обнуляются
- 3) на каждом шаге цикла при выполнении некоторого условия переменная a увеличивается на 1, а если это условие не выполняется, то на 1 увеличивается b; таким образом, обе переменных – счётчики
- 4) теперь посмотрим на условие  $c = 0$ : в предыдущей строке в переменную c записывается остаток от деления числа x на 2, то есть, переменная c определяет четность числа или, что равносильно, **чётность его последней цифры**
- 5) если последняя цифра чётная, то увеличивается счётчик a, а если нечётная – увеличивается счётчик b
- 6) в конце каждого шага цикла операция  $x:=x \text{ div } 10$  отсекает последнюю цифру в десятичной записи числа
- 7) таким образом, делаем вывод: после завершения цикла в переменной a находится количество чётных цифр в десятичной записи числа, а в переменной b – количество нечётных цифр
- 8) если было выведено 3 и 2, то в числа 5 цифр, из них 3 чётных и 2 нечётных; таким образом, нам нужно найти минимальное пятизначное число, в котором 3 чётные и 2 нечётные цифры
- 9) минимальная чётная цифра – это 0, минимальная нечётная – 1; 0 не может стоять на первом месте, поэтому число начинается с 1
- 10) для получения минимального числа после 1 должны идти нули и последняя цифра – снова 1
- 11) ответ: **10001**

**Ещё пример задания:**

**P-02.** Ниже записан алгоритм. После выполнения алгоритма было напечатано 3 числа. Первые два напечатанных числа – это числа 9 и 81. Какое наибольшее число может быть напечатано третьим?

```

var x, y, z: integer;
    r, a, b: integer;
begin
  readln(x, y);
  if y > x then begin
    z:= x; x:= y; y:= z;
  end;
  a:= x; b:= y;
  while b > 0 do begin
    r:= a mod b;
    a:= b;
    b:= r;
  end;
  writeln(a);
  writeln(x);
  write(y);

```

end.

Решение:

- 1) сложность этой задачи состоит в том, чтобы разобраться в алгоритме
- 2) сначала вводятся два числа и переставляются так, чтобы в переменной **x** было наибольшее число, а в переменной **y** – наименьшее из двух:

```
if y > x then begin
  z:= x; x:= y; y:= z;
end;
```

- 3) затем исходные значения копируются в переменные **a** и **b** и с ними выполняется следующий алгоритм

```
while b > 0 do begin
  r:= a mod b;
  a:= b;
  b:= r;
end;
```

его суть сводится к тому, что меньшее из двух чисел, **a** и **b**, каждый раз заменяется на остаток от деления большего на меньшее до тех пор, пока этот остаток не станет равен нулю;

- 4) делаем вывод, что это классический [Алгоритм Евклида](#), который служит для вычисления наибольшего общего делителя (НОД) двух чисел; это делитель в результате оказывается в переменной **a**
- 5) смотрим, что выводится на экран: сначала значение переменной **a** (наибольший общий делитель исходных чисел, НОД( $x,y$ )), затем значение **x** (большее из исходных чисел) и значение **y** (меньшее из исходных чисел)
- 6) по условию первое число – 9, второе – 81, поэтому третье число должно быть меньше, чем 81, и  $\text{НОД}(81,y) = 9$
- 7) наибольшее число, которое меньше 81 и делится на 9, равно 72 (обратите внимание, что исходные числа не могут быть равны, потому что в этом случае их НОД был бы равен 81)
- 8) ответ: **72**

Ещё пример задания:

**P-01.** Ниже записана программа. Получив на вход число  $x$ , эта программа печатает два числа,  $L$  и  $M$ . Укажите наибольшее из таких чисел  $x$ , при вводе которых алгоритм печатает сначала 3, а потом 7.

```
var x, L, M: integer;
begin
  readln(x);
  L:=0; M:=0;
  while x > 0 do begin
    L:=L+1;
    if M < (x mod 10) then begin
      M:=x mod 10;
    end;
    x:= x div 10;
  end;
  writeln(L); write(M);
end.
```

Решение:

- 1) для решения задачи необходимо понять, что делает эта программа
- 2) если это не видно сразу, можно выполнить ручную прокрутку для какого-то простого числа, например, для числа 251:

оператор	условие	x	L	M
readln(x);		251	?	?

L:=0; M:=0;			0	0
while x > 0 do...	251 > 0? да			
L:=L+1;			1	
if M<(x mod 10) then...	M<(251 mod 10)? да			
M:=x mod 10;				1
x:=x div 10;		25		
while x > 0 do...	25 > 0? да			
L:=L+1;			2	
if M<(x mod 10) then...	M<(25 mod 10)? да			
M:=x mod 10;				5
x:=x div 10;		2		
while x > 0 do...	2 > 0? да			
L:=L+1;			3	
if M<(x mod 10) then...	M<(2 mod 10)? нет			
x:=x div 10;		0		
while x > 0 do...	0 > 0? нет			
writeln(L); write(M);			3	5

- 3) можно догадаться, что в результате работы программы в переменной L окажется число цифр числа, а в переменной M – наибольшая цифра, но это предположение нужно постараться доказать
- 4) нужно вспомнить (и запомнить), что для целого числа остаток от деления на 10 ( $x \bmod 10$ ) – это последняя цифра в десятичной записи числа, а целочисленное деление ( $x \div 10$ ) отсекает последнюю цифру, то есть из 123 получается 12
- 5) рассмотрим цикл, число шагов которого зависит от изменения переменной **x**:
- ```
while x > 0 do begin
    ...
    x:= x div 10;      { отсечение последней цифры }
end;
```
- здесь оставлены только те операторы, которые влияют на значение **x**
- 6) из приведенного цикла видно, что на каждом шаге от десятичной записи **x** отсекается последняя цифра до тех пор, пока все цифры не будут отсечены, то есть **x** не станет равно 0; поэтому **цикл выполняется столько раз, сколько цифр в десятичной записи введенного числа**
- 7) на каждом шаге цикла переменная **L** увеличивается на 1:
- ```
L:=L+1;
```
- других операторов, меняющих значение **L**, в программе нет; поэтому после завершения цикла **в переменной L действительно находится количество цифр**
- 8) теперь разберемся с переменной **M**, которая сначала равна 0; оператор, в котором она меняется, выглядит так:
- ```
if M < (x mod 10) then begin
    M:=x mod 10;
end;
```
- учитывая, что  $x \bmod 10$  – это последняя цифра десятичной записи числа, получается что если эта цифра больше, чем значение **M**, она записывается в переменную **M**;
- 9) этот оператор выполняется в цикле, причем выражение  $x \bmod 10$  по очереди принимает значения всех цифр исходного числа; поэтому после завершения цикла **в переменной M окажется наибольшая из всех цифр**, то есть наша догадка подтверждается
- 10) итак, по условию задачи фактически требуется найти наибольшее трехзначное число, в котором наибольшая цифра – 7; очевидно, что это 777.

11) ответ: **777**.

