

Богомолова Ольга Борисовна,
Усенков Дмитрий Юрьевич

«ЕГЭ, № 20»: АНТОЛОГИЯ ЗАДАЧ ПО ПРОГРАММИРОВАНИЮ НА ВЫДЕЛЕНИЕ И ОБРАБОТКУ ЦИФР

Задачи ЕГЭ под номером 20 нацелены на проверку знаний по алгоритмике и программированию и навыков чтения и понимания программ. В последние годы их условие бывает почти «типовым»: анализируемая программа принимает на вход целое число, в цикле производится «разборка» этого числа на отдельные цифры, и эти цифры каким-либо образом обрабатываются. В результате выводятся два (реже – три) некоторых значения, по которым необходимо определить возможное входное число (наибольшее либо наименьшее). В целом такие задачи не сложны. Но разработчики стараются всеми силами их усложнить, например, за счет использования недесятичных систем счисления.

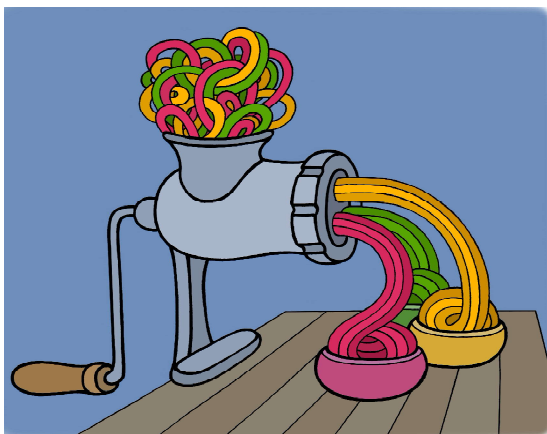
Данная статья представляет собой «антологию» разбора задач этого типа за последние 5–7 лет. Рекомендуется поочередно разобрать их решения (задачи приведены по порядку «от простого к более сложному», а решения вначале расписаны подробно, а в последующих задачах более кратко), обращая внимание на выделенные в рамках методические комментарии.

Задача 1. Получив на вход число x , программа печатает два числа a и b . Укажите наибольшее из таких чисел x , при вводе которых программа напечатает сначала число 3, а потом число 35.

```
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. Программа должна первым напечатать значение 3. Поскольку первым стоит оператор вывода `writeln(a)`, по завершении выполнения цикла переменная a должна быть



равна 3. Изначально $a = 0$. Изменение этой переменной в цикле производится оператором $a := a+1$, то есть значение a определяет количество проходов цикла.

2. Таким образом, цикл выполняется 3 раза. При этом второе печатаемое число (значение переменной b) равно 35.

Анализируем цикл целиком:

```
while x>0 do begin
  a := a+1;
  b := b*(x mod 10);
  x := x div 10;
end;
```

Конструкция, включающая в себя сначала вычисление остатка от деления числа x на 10, а затем меняющая значение x на результат целочисленного деления x на 10, в цикле пока $x > 0$, является типовой и реализует последовательное выделение цифр числа x .

Таким образом, данный цикл выполняет поочередное (справа налево – от младших разрядов к старшим) выделение цифр числа x и вычисление в переменной b произведения этих цифр.

3. Итак, нам нужно найти наибольшее значение числа x , у которого произведение цифр равно 35, а количество цифр равно трем (так как цикл выделения каждой цифры выполнялся трижды).

Число 35 можно разложить на множители $5 \cdot 7$, причем эти множители простые. Чтобы получить три множителя, можно добавить к ним единицу: $35 = 5 \cdot 7 \cdot 1$.

Таким образом, для получения заданных результатов (чисел 3 и 35) нам годятся значения x , равные всем возможным комбинациям цифр 1, 5 и 7. Наибольшее же из таких чисел равно, очевидно, 751.

Ответ: $x = 751$.

Важно не забывать, что при определении исходного значения x надо учитывать не только разложение второго из заданных чисел (здесь – 35) на простые множители, но и количество проходов цикла, определяемое первым из заданных чисел (здесь – 3).

Задача 2. Получив на вход число x , программа печатает два числа a и b . Укажите наименьшее из таких чисел x , при вводе которых программа напечатает сначала число 4, а потом число 24.

```
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. Программа должна первым напечатать значение 4, значит, по завершении выполнения цикла переменная a должна быть равна 4. Изначально $a = 0$, изменение этой переменной в цикле производится оператором $a := a+1$, значит, цикл должен быть выполнен 4 раза. При этом второе печатаемое число (значение переменной b) равно 24.

2. Анализируем цикл целиком:

```
while x>0 do begin
  a := a+1;
  b := b*(x mod 10);
  x := x div 10;
end;
```

Конструкция, включающая в себя сначала вычисление остатка от деления числа x на 10, а затем меняющая значение x на результат целочисленного деления x на 10, в цикле пока $x > 0$, является типовой и реализует последовательное выделение цифр числа x .

Таким образом, данный цикл выполняет поочередное (справа налево – от младших разрядов к старшим) выделение цифр числа x и вычисление в переменной b произведения этих цифр.

3. Итак, нам нужно найти наименьшее значение числа x , у которого произведение цифр равно 24, а количество цифр равно четырем (так как цикл выделения каждой цифры выполнялся четыре раза).

Табл. 1

Цифры	3, 8, 1, 1	4, 6, 1, 1	4, 3, 2, 1	2, 2, 6, 1	2, 2, 2, 3
Наименьшее возможное число	1138	1146	1234	1226	2223

Число 24 можно разложить на четыре однозначных множителя различными способами:

$$24 = 3 \cdot 8 \cdot 1 \cdot 1 = 4 \cdot 6 \cdot 1 \cdot 1 = 4 \cdot 3 \cdot 2 \cdot 1 = 2 \cdot 2 \cdot 6 \cdot 1 = 2 \cdot 2 \cdot 2 \cdot 3.$$

Таким образом, в качестве исходного числа x подходят все возможные комбинации цифр: 3, 8, 1, 1 или 4, 6, 1, 1 или 4, 3, 2, 1 или 2, 2, 6, 1 или 2, 2, 2, 3.

Найдем наименьшие возможные значения x , которые можно получить из таких цифр, в каждой из полученных групп, и сравним их (см. табл. 1).

Тогда наименьшее возможное значение x , удовлетворяющее условиям задачи, равно 1138.

Ответ: $x = 1138$.

Задача 3. Получив на вход число x , программа печатает два числа a и b . Укажите наибольшее из таких чисел x , при вводе которых программа напечатает сначала число 4, а потом число 7.

```
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. Программа должна первым напечатать значение 4, значит, по завершении выполнения цикла переменная a должна быть равна 4. Изначально $a = 0$, изменение этой переменной в цикле производится оператором $a := a+1$, значит цикл должен быть выполнен 4 раза.

2. Второе печатаемое число (значение переменной b) должно быть равно 7.

Анализируем цикл целиком:

```
while x>0 do begin
  a := a+1;
  b := b+(x mod 10);
  x := x div 10;
end;
```

Конструкция, включающая в себя сначала вычисление остатка от деления числа x на 10, а затем меняющая значение x на результат целочисленного деления x на 10, в цикле пока $x > 0$, является типовой и реализует последовательное выделение цифр числа x .

Таким образом, данный цикл выполняет поочередное (справа налево – от младших разрядов к старшим) выделение цифр числа x и вычисление в переменной b суммы этих цифр.

3. Итак, нам нужно найти наибольшее значение числа x , у которого сумма цифр равна 7, а количество цифр равно четырем (так как цикл выделения каждой цифры выполнялся четыре раза).

Число 7 можно представить как суммы различных цифр, например:

$$7 = 1 + 1 + 1 + 4;$$

$$7 = 1 + 2 + 3 + 3;$$

$$7 = 2 + 2 + 2 + 1;$$

и т. д.

Однако, в общем-то, нетрудно догадаться, что наибольшее из возможных четырехзначных значений x , сумма цифр которого была бы равна 7, равно 7000 (сумма цифр: $7 + 0 + 0 + 0$).

Ответ: $x = 7000$.

Задача 4. Получив на вход число x , программа печатает два числа a и b . Укажите наибольшее из таких чисел x , при вводе которых программа напечатает сначала число 3, а потом число 10.

```
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. Программа должна первым напечатать значение 3, значит, по завершении выполнения цикла переменная a должна быть равна 3. Изначально $a = 0$, изменение этой переменной в цикле производится оператором $a := a+1$, значит, цикл должен быть выполнен 3 раза.

2. Второе печатаемое число (значение переменной b) должно быть равно 10.

Анализируем цикл целиком:

```

while x>0 do begin
  a := a+1;
  b := b+(x mod 10);
  x := x div 10;
end;

```

Конструкция, включающая в себя сначала вычисление остатка от деления числа x на 10, а затем меняющая значение x на результат целочисленного деления x на 10, в цикле пока $x > 0$, является типовой и реализует последовательное выделение цифр числа x .

Таким образом, данный цикл выполняет поочередное (справа налево – от младших разрядов к старшим) выделение цифр числа x и вычисление в переменной b суммы этих цифр.

3. Итак, нам нужно найти наибольшее значение числа x , у которого сумма цифр равна 10, а количество цифр равно трем (так как цикл выделения каждой цифры выполнялся три раза).

Очевидно, что воспользоваться той же идеей, что и в предыдущей задаче (когда сумма цифр формировалась как само значение суммы плюс требуемое количество нулей), здесь нельзя: 10 – это не цифра. Поэтому нам все же придется искать представление числа 10 как суммы трех цифр, таких, что составленное из них число – максимально возможное.

Нетрудно догадаться, как выполнить такое разложение: $10 = 9 + 1 + 0$.

Ответ: $x = 910$.

Задача 5. Получив на вход число x , алгоритм печатает два числа a и b . Укажите *наибольшее* из таких чисел x , при вводе которых алгоритм печатает сначала 4, а потом 20.

```

var x, a, b: integer;
begin
  readln(x);
  a := 0; b := 0;
  while x > 0 do
  begin
    a := a + 1;
    if x mod 2 = 0 then
      b := b + x mod 10;
    x := x div 10;
  end;
  writeln(a); write(b);
end.

```

Решение

Анализируем алгоритм.

Очевидно, что переменная a представляет собой счетчик количества проходов цикла.

В самом цикле сначала выполняется проверка четности числа (то есть фактически – четности его последней цифры). Если число (а следовательно, и последняя его цифра) четно, то последняя цифра числа выделяется ($x \bmod 10$) и прибавляется к переменной b . После этого, уже вне условного оператора, из числа x отбрасывается последняя цифра ($x \div 10$). Таким образом, в программе выполняется подсчет суммы четных цифр введенного числа.

Поскольку в результате на экран выводится пара чисел 4 и 20, делаем вывод, что количество цифр в исходном числе (определяющее количество проходов цикла) равно 4, а сумма четных цифр в этом числе равна 20.

Остается найти наибольшее из возможных четырехзначных чисел, в котором сумма четных цифр равна 20.

Вспоминаем правило: в большем числе цифры расположены по убыванию.

Поэтому нам «выгодно» выделить в составе суммы 20 одно число – четное и соответствующее наибольшему из возможных значению четной десятичной цифры (оче-

видно, это 8). Оставшуюся часть суммы (12) проверяем: может ли она соответствовать какой-либо четной цифре. В нашем случае она превышает максимально возможное значение четной цифры (8). Поэтому мы снова выделяем из этой суммы четное и наибольшее из возможных значение 8. Остается число 4. Таким образом, мы разбиваем сумму 20 на три слагаемых: $8 + 8 + 4$.

Таким образом, мы определили три четных цифры, входящих в исходное число. Оставшаяся четвертая цифра должна быть нечетной. И поскольку нам требуется наибольшее исходное число из возможных, а нечетные цифры на подсчет суммы не влияют, мы можем выбрать эту цифру равной 9.

Итак, мы имеем четыре цифры: 8, 8, 4 и 9. Остается расположить их по убыванию, чтобы получить число 9884. Оно и будет наибольшим из возможных, которое в результате работы указанной программы даст на экране значения 4 и 20.

Ответ: 9884.

Задача 6. Получив на вход число x , алгоритм печатает два числа a и b . Укажите *наименьшее* из таких чисел x , при вводе которых алгоритм печатает сначала 6, а потом 18.

```
var x, a, b: integer;
begin
  readln(x);
  a := 0; b := 0;
  while x > 0 do
    begin
      a := a + 1;
      if x mod 2 = 0 then
        b := b + x mod 10;
      x := x div 10;
    end;
  writeln(a); write(b);
end.
```

Решение

Как и в предыдущей задаче, алгоритм – тот же самый: переменная a – это счетчик количества проходов цикла и, следовательно, количество цифр в исходном числе (6 проходов – шестизначное число), а в переменной b подсчитывается сумма четных цифр числа. Однако теперь нужно найти наименьшее из возможных шестизначных чисел, в котором сумма четных цифр равна 18.

Вспоминаем правило: в меньшем числе цифры расположены по возрастанию, при этом самая первая цифра должна быть ненулевая, наибольшие цифры должны быть «сгруппированы» в конце числа (справа), а начальные цифры (кроме самой первой) «выгодно» заполнять нулями.

Наша сумма (18) превышает возможное наибольшее значение четной цифры (8). Пробуем выделить из этой суммы слагаемое, равное 8 (которое мы возьмем в качестве последней цифры). Остается 10. Это число тоже превышает возможное наибольшее значение четной цифры, поэтому из него мы снова выделяем слагаемое 8 (это будет предпоследняя цифра числа). Остается число 2, которое может являться четной десятичной цифрой. Таким образом, мы разбили сумму 18 на три слагаемых: $8 + 8 + 2$.

Три цифры исходного числа нам известны. Осталось определить еще три. Одну из них берем ненулевую наименьшую возможную нечетную – это единица. Она в создаваемом числе будет самой первой, а на подсчет суммы не влияет. А еще две цифры «заполняем» нулями.

Остается записать полученные цифры (8, 8, 2, 1, 0, 0) в таком порядке, чтобы получилось наименьшее из возможных шестизначных чисел. Это число 100288.

Ответ: 100288.

Задача 7. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наименьшее из чисел x , при вводе которого алгоритм напечатает 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

Проанализируем алгоритм. В цикле, который выполняется, пока значение x остается больше нуля, сначала на единицу увеличивается значение переменной a , затем к переменной b прибавляется значение $x \bmod 10$, а потом значение x меняется на $x \operatorname{div} 10$.

Очевидно, что a – это счетчик количества проходов цикла. В переменной b накапливается сумма остатков от деления текущего значения числа x на 10. Но каждый такой остаток – ни что иное как последняя цифра десятичного числа. Следовательно, переменная b накапливает сумму цифр числа x . Оператор же $x := x \operatorname{div} 10$ (что тоже вполне очевидно) отбрасывает в числе x уже отработанную последнюю цифру.

При таком алгоритме каждый проход цикла обрабатывает и отбрасывает одну цифру исходного числа, и так до тех пор, пока не будет отброшена самая первая цифра исходного числа. Значит, количество проходов цикла, вычисленное в переменной a , совпадает с количеством цифр в исходном числе.

Согласно условию задачи, количество цифр в числе (a) равно 2 (двухзначное число x), а сумма цифр числа (b) равна 12.

Какие возможны двухзначные числа, сумма цифр которых равна 12? Проще всего начать перебор вариантов с цифры 9, а затем уменьшать значение первой цифры в каждой возможной паре, пока это возможно и пока не начнут появляться повторяющиеся «симметричные» варианты: $12 = 9 + 3 = 8 + 4 = 7 + 5 = 6 + 6$ (далее получаются «симметричные» варианты – $5 + 7, 4 + 8$ и $3 + 9$, которые уже можно не рассматривать).

Этим вариантам соответствуют возможные значения x , равные: 93, 39, 84, 48, 75, 57 и 66. Из них наименьшим является число 39.

Ответ: 39.

Задача 8. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наибольшее из чисел x , при вводе которого алгоритм напечатает сначала 2, а потом 26.

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

```
while x > 0 do
  begin
    a := a+1;
    b := b + (x mod 100) ;
    x := x div 100;
  end;
writeln(a); write(b);
end.
```

Решение

Вновь проанализируем алгоритм – что в нем изменилось по сравнению с предыдущей задачей? Теперь в операциях `mod` и `div` вторым операндом является число 100, а не 10. Следовательно, теперь в цикле обрабатываются не отдельные цифры исходного числа, а числа, составленные из пар таких цифр.

Значение переменной a равно 2. Значит, в исходном числе (раз оно разбивается на пары) может быть или четыре цифры, или три (вторая по счету выделяемая пара – неполная). Нам, по условию, нужно искать наибольшее из возможных чисел, значит, нужно пытаться брать в рассмотрение число из наибольшего количества цифр из четырех.

Сумма числа, которое составлено из двух последних цифр, и числа, составленного из первых двух цифр, равна 26. Нам нужно получить как можно большее возможное значение x , а значит, наибольшие цифры должны стоять в нем в начале числа. Предположим, что эти первые цифры – и есть число 26. Тогда второе число из пары цифр будет равно 00. Задача решена.

Ответ: 2600.

Задача 9. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать *наименьшее* из чисел x , при вводе которого алгоритм напечатает сначала 2, а потом 14.

```
var x, a, b: integer;
begin
  readln(x);
  a := 0; b := 0;
  while x > 0 do
    begin
      a := a+1;
      b := b + (x mod 100) ;
      x := x div 100;
```

```
end;  
writeln(a); write(b);  
end.
```

Решение

В цикле вновь обрабатываются не отдельные цифры исходного числа, а числа, составленные из пар таких цифр. Значение переменной a равно 2. Значит, в исходном числе может быть или четыре цифры, или три (вторая по счету выделяемая пара – неполная). Нам нужно искать наименьшее из возможных чисел, значит, нужно брать в рассмотрение число из наименьшего количества цифр – из трех.

Сумма числа, которое составлено из двух последних цифр, и числа, составленного из первых двух цифр, равна 14. Как ее «раскидать» на два двузначных числа наиболее «выгодным» для нас способом?

Очевидно, первая цифра числа x должна быть как можно меньшей, но нулем она быть не может (иначе цикл выполнялся бы только один раз, и значение a было бы равно 1, а не 2). А наименьшая возможная цифра – это 1. Ее мы и попробуем выделить из нашей суммы и поставить на первое место в числе. Оставшиеся две последние цифры числа x легко вычислить: $14 - 1 = 13$. Тогда всё число x (наименьшее из возможных) равно 113.

Ответ: 113.

Задача 10. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наименьшее из таких чисел x , при вводе которого алгоритм напечатает сначала 2, а потом 3.

```
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); write(b);  
end.
```

Решение

Здесь алгоритм совершенно другой. Подсчет количества проходов цикла не ведется вообще. В переменную c записывается остаток от деления текущего значения числа x на 2, а затем если $c = 0$, то на единицу увеличивается значение a , а если $c = 1$, то на единицу увеличивается значение b (которые в итоге и выводятся на экран). А после этого у исходного числа x отбрасывается последняя цифра.

Нетрудно догадаться, что данный алгоритм, последовательно выделяя из исходного числа отдельные цифры, определяет четность каждой такой цифры и подсчитывает в переменной a количество четных, а в переменной b – количество нечетных цифр. По условию, четных цифр в этом числе – 2, а нечетных – 3.

Не менее очевидно, что сумма итоговых значений a и b дает нам общее количество цифр в числе: $2 + 3 = 5$.

Итак, нам нужно найти наименьшее из возможных 5-значных чисел, в котором две четных цифры и три нечетных. Какие это могут быть цифры и как они расположены в числе?

Очевидно, что для получения наименьшего возможного числа надо брать и наименьшие возможные цифры. Наименьшая нечетная цифра – это единица. А наименьшая четная? Если учесть, что остаток от деления нуля на 2 тоже равен нулю, то в качестве наименьшей четной цифры у нас в таких задачах выступает нуль.

Итак, в числе x должно быть три единицы и два нуля. В каких разрядах их расположить? Самой первой цифрой, очевидно, должна стоять ненулевая, то есть в старшем разряде будет единица. А остальные цифры располагаем по правилу: в наименьшем возможном числе цифры по разрядам нужно располагать по возрастанию слева направо (чтобы в старших разрядах, «вес» которых максимален, оказывались самые маленькие цифры). Значит, после «лидирующей» обя-

зательной единицы (которая определяет пятизначность числа) сначала надо записать оба нуля, а в конце – две оставшиеся единицы. Получим число 10011 (обратим внимание – десятичное).

Ответ: 10011.

А что будет, если нечетных цифр нет вообще?

Задача 11. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наименьшее из таких чисел x , при вводе которого алгоритм напечатает сначала 4, а потом 0.

```
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); write(b);
end.
```

Решение

Первая часть решения та же, что и в предыдущем случае. Не будем повторять ее, а просто констатируем, что в искомом числе x имеются четыре четных цифры и ни одной нечетной и это число – четырехзначное.

Как сформировать такое число, чтобы оно было и четырехзначным и наименьшим?

Наименьшая четная цифра, как мы ранее выяснили, – это нуль. Но ведь число не может состоять только из одних нулей! Значит, хотя бы одна (самая старшая) цифра должна быть ненулевой. А какая у нас самая маленькая четная цифра, но не нуль? Правильно – двойка. Тогда наименьшее возможное число x можно записать так: 2000.

Ответ: 2000.

А если нужно искать не наименьшее число, а наибольшее?

Задача 12. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наибольшее из таких чисел x , при вводе которого алгоритм напечатает сначала 3, а потом 4.

```
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); write(b);
end.
```

Решение

Первая часть решения тоже такая же, как и ранее. Нетрудно найти, что в искомом числе x имеются три четных цифры и четыре нечетных, а само число – семизначное.

Теперь нам требуется найти наибольшее возможное число, значит, и цифры для него надо выбирать наибольшие из возможных. Самая большая нечетная цифра – это 9, а самая большая четная цифра – 8.

Остается только правильно записать эти три восьмерки и четыре девятки. В наибольшем числе цифры записываются слева направо по убыванию (чтобы в старших разрядах, «вес» которых максимален, оказывались самые большие цифры). Тогда искомое число – 9999888.

Ответ: 9999888.

А теперь еще усложним условие задачи: перейдем к недесятичным системам счисления.

Задача 13. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наибольшее из таких чисел x , при вводе которого алгоритм напечатает сначала 2, а потом 3.

```
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 8
end;
writeln(a); write(b);
end.

```

Решение

Смотрим на алгоритм. Он почти такой же, как и в предыдущих задачах, но вместо оператора $x := x \text{ div } 10$, который, как мы знаем, отбрасывает в десятичном числе последнюю цифру, у нас записано $x := x \text{ div } 8$.

Первая идея, которая приходит в голову, – это пытаться отслеживать кратность при каждом очередном делении на 8 и, исходя из этого, пытаться определить количество цифр. Но, как легко догадаться, такая интерпретация представленного алгоритма слишком сложна и мало наглядна.

А что если мысленно представить исходное число x так, чтобы и при выполнении операции $\text{div } 8$ в числе отбрасывалась последняя цифра? Это вполне возможно, если число x понимать как записанное в восьмеричной системе счисления!

Вот это и есть «ключ» к решению любых задач такого типа (где в операторе $x := x \text{ div } \dots$ стоит не 10, а другое число). Остальная же часть решения подобна рассмотренным выше.

По условию, в числе x (которое мы рассматриваем в восьмеричном формате!) две четные цифры и три нечетные. Значит, всего их пять. Число требуется наибольшее из возможных – значит, цифры надо брать наибольшие из возможных и располагать их в числе по правилу «большие – слева».

Самая большая четная восьмеричная цифра – это 6. Самая большая нечетная восьмеричная цифра – 7. Тогда наибольшее пятизначное число, составленное из этих цифр – это 77766.

Остается только вспомнить, что мы рассматривали и «расписывали» число в

восьмеричном виде, а в задаче-то имелось в виду десятичное число x . Значит, нужно восьмеричное значение 77766 преобразовать в десятичный формат: $77766_8 = 32758_{10}$.

Ответ: 32758.

Вполне очевидно: если бы требовалось найти наименьшее возможное значение x , то рассуждения были бы аналогичными. Но только мы выбирали бы наименьшие возможные восьмеричные цифры (четная – 0, нечетная – 1) и записали бы одну единицу в старшем разряде (так как в начале числа должна стоять хотя бы одна ненулевая цифра), потом записали бы все имеющиеся нули, а в конце числа – оставшиеся единицы: 10011. А после этого осталось бы только преобразовать полученное восьмеричное число в десятичное: $10011_8 = 4105_{10}$.

Задача 14. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать сумму наибольшего и наименьшего из таких чисел x , при вводе которых алгоритм напечатает сначала 2, а потом 3.

```

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 2
  end;
  writeln(a); write(b);
end.

```

Решение

Смотрим на алгоритм. В операции div записана двойка – значит, будем работать в двоичной системе счисления.

По условию, в числе x (рассматриваемом в двоичном формате) две четные цифры и три нечетные – всего пять. В двоичной системе всё однозначно: единственная нечетная цифра – единица, а единственная четная – нуль. Значит, наименьшее или наибольшее

значение числа x определяется исключительно расположением таких цифр в его разрядах.

Наименьшее возможное двоичное пятизначное число из трех единиц и двух нулей – это $10011_2 = 19_{10}$. Наибольшее возможное такое число – $11100_2 = 28_{10}$. Ну, а искомая их сумма тогда равна $19 + 28 = 47$.

Ответ: 47.

Задача 15. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наибольшее из таких чисел x , при вводе которого алгоритм напечатает два числа 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 5
    end;
  writeln(a); write(b);
end.
```

Решение

В операции **div** записана пятерка – значит, наша рабочая система счисления – пятеричная.

По условию, в числе x (рассматриваемом в пятеричном формате) две четные и две нечетные цифры – всего четыре, значит, число четырехзначное.

Требуется наибольшее возможное число – значит, выбираем наибольшие возможные четные и нечетные пятеричные цифры – нечетная 3 и четная 4, а в числе записываем их по убыванию слева направо: 4433.

Остается перевести полученное пятеричное число в десятичный формат:

$$4433_5 = 4 * 5^3 + 4 * 5^2 + 3 * 5 + 3 = 618_{10}.$$

Ответ: 618.

Как эта задача решалась бы, если бы требовалось найти наименьшее возможное значение x , наверное, понятно без дополнительных объяснений.

Итак, если в условии подобной задачи с четностью и нечетностью в операции **div** вместо 10 стоит другое число, то не пугаемся. Это число, стоящее в **div**, берем в качестве основания системы счисления, затем решаем задачу поиска числа в ней (как обычно, только помним, что наибольшие цифры не могут превышать основание системы счисления), а в конце – самое главное! – не забываем преобразовать полученное число из «рабочей» системы счисления в десятичную.

Не сложнее и задачи, в которых, вместо определения четности последней цифры, производится ее сравнение с какими-либо значениями и подсчет в переменных a и b найденных таких случаев.

Задача 16. Имеется алгоритм. Получив на вход число x , он печатает три числа: d , a и b . Нужно указать наибольшее из таких чисел x , при вводе которого алгоритм напечатает числа 6, 2 и 1.

```
var x, a, b, c: integer;
begin
  readln(x);
  a := 0; b := 0;
  while x > 0 do
    begin
      d := d + 1;
      c := x mod 10
      if c < 3 then a := a + 1
      if c > 7 then b := b + 1
      x := x div 10
    end;
  writeln(d); writeln(a);
  write(b);
end.
```

Решение

В переменной d подсчитывается количество проходов цикла, а значит, в данном случае и количество цифр в числе x , – в нашем случае 6.

Далее выделяется последняя цифра числа. Если она меньше 3, то увеличивается значение a , а если она больше 7, то увеличивается значение b . (Очевидно, для цифр из диапазона [3...7] ничего не происходит). Затем последняя цифра числа отбрасывается.

Ищется наибольшее возможное число x , то есть наибольшее возможное десятичное шестизначное число, в котором есть две цифры меньше 3 и одна цифра больше 7. (Очевидно, остальные три цифры должны быть из диапазона от 3 до 7.)

В наибольшем возможном числе нужно брать наибольшие возможные цифры, удовлетворяющие условию. Значит, мы можем взять одну девятку ($9 > 7$), три семерки (это те цифры, которые не вошли в подсчеты количеств в a и b) и две двойки ($2 < 3$).

Кроме того, в наибольшем возможном числе надо вначале записать самые большие цифры. Значит, наше число – 977722.

Ответ: 977722.

Нетрудно догадаться, что если в алгоритме вместо `mod 10` и `div 10` будет вместо 10 записано какое-то другое число, то решаться такая задача будет так же, но в указанной этим числом системе счисления. При этом к условиям, записанным в операторах `if`, добавится еще условие, что никакая цифра не больше основания системы счисления, а полученное в итоге число надо будет не забыть перевести из «рабочей» системы счисления в десятичную.

Задача 17. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наибольшее положительное пятизначное число x , при вводе которого алгоритм напечатает сначала 3, а потом 4.

```
var x, a, b, c: integer;
begin
  readln(x);
  a := 0; b := 0;
  while x > 0 do
  begin
    c := x mod 10
    if c > 5 then a := a + 1
    if c < 8 then b := b + 1
    x := x div 10
  end;
  writeln(a); write(b);
end.
```

Решение

Здесь, в отличие от предыдущей задачи, количество проходов цикла не подсчитыва-

ется, зато в условии сказано, что число пятизначное.

Сначала в нём выделяется последняя цифра. Если она больше 5, то увеличивается значение переменной a . Если она меньше 8, то увеличивается значение переменной b . После этого последняя цифра числа отбрасывается.

Исходя из условия задачи, в числе есть три цифры, больших 5, и четыре цифры, меньших 8. А всего в числе 5 цифр, и это число должно быть наибольшим. Значит, и цифры нужно стараться выбирать как можно большими.

Если в числе 4 цифры меньше 8, то значит, в нём есть только одна цифра больше 8. Выберем самую большую из таких цифр – девятку.

Но это как раз одна из цифр, больших 5. Значит, в нашем числе, кроме уже найденной девятки, может быть только две цифры больше 5, но меньше 8. Самая большая цифра, которая удовлетворяет этим условиям, – это семерка. Значит, в нашем числе мы нашли одну девятку и две семерки и этим исчерпали всё количество цифр, больших 5.

В числе осталось еще две цифры. Очевидно, они могут быть только не больше 5 (поскольку лимит цифр больше 5 мы уже исчерпали). Наибольшая же цифра, не большая 5, – это и есть сама пятерка.

Итак, мы нашли, что в нашем числе могут быть: одна девятка, две семерки и две пятерки. Остается записать их в числе так, чтобы число было наибольшим из возможных. Следовательно, надо записывать цифры по убыванию: 9, 7, 7, 5, 5, а получим число 97755.

Ответ: 97755.

Задача 18. Имеется алгоритм. Получив на вход число x , он печатает два числа a и b . Нужно указать наибольшее четырехзначное число x , при вводе которого алгоритм напечатает сначала 5, а потом 7.

```
var x, y, a, b: integer;
begin
  a := 10;
  b := 0;
  readln(x);
```

```

while x > 0 do
  begin
    y := x mod 10;
    x := x div 10
    if y < a then
      a := y;
    if y > b then
      b := y;
  end;
writeln(a);
writeln(b)
end.

```

Решение

И снова здесь другой алгоритм в теле цикла, с которым мы ранее не встречались. Анализируем его.

Вначале в переменную y выделяется последняя цифра из очередного десятичного значения числа, а в исходном числе она отбрасывается.

Затем записано условие: **if $y < a$ then $a := y$** , причем изначально $a = 10$, то есть значению, заведомо большему, чем любая цифра.

Что нам это напоминает? Конечно же, всем известный алгоритмический прием поиска минимума:

– изначально предполагаемый минимум берется больше, чем самое большое возможное значение;

– если очередное значение меньше предполагаемого минимума, то это значение берется в качестве нового предполагаемого минимума.

Следовательно, в переменной a ищется минимальная цифра числа.

Аналогично, условие **if $y > b$ then $b := y$** , как нетрудно догадаться, – это поиск максимальной цифры числа.

Итого имеем четырехзначное число, в котором наименьшая цифра (a) равна 5, а наибольшая цифра (b) равна 7. Это означает, что цифр от 0 до 4 и цифр 8 и 9 в числе быть не может (иначе значения a и b получились бы другие) и что в числе могут быть только цифры 5, 6 и 7, причем количество их может быть любым, главное, чтобы была хотя бы одна пятерка и хотя бы одна семерка.

Мы ищем наибольшее возможное четырехзначное число, состоящее из таких цифр. Значит, надо записывать их по убыванию,

причем взять как можно больше самых больших возможных цифр. Тогда в качестве числа x можно записать: 7775.

Ответ: 7775.

Очевидно, что при поиске наименьшего такого числа нужно было бы брать как можно больше самых маленьких цифр и записать их по возрастанию – мы получили бы число 5557. А если бы выяснилось, что значение a равно нулю, то есть наименьшая цифра равна нулю, то, чтобы получить четырехзначное число и записать в его начале ненулевую цифру, мы выбрали бы одну наименьшую возможную ненулевую цифру из допустимого диапазона (единицу), затем записали бы максимально возможное количество нулей, а в конце – единственную цифру, равную найденному значению максимума: 1007.

А если бы в операциях **mod** и **div** вместо 10 было бы записано другое число, то мы бы решали задачу аналогичным путем, но в соответствующей системе счисления (и, конечно, не забыли бы преобразовать полученное число в десятичную систему счисления)...

Задача 19. Определить наибольшее двузначное натуральное число, при вводе которого программа выведет на экран число 0.

```

var i, n: longint;
begin
  i := 0;
  readln(n);
  while (n > 0) do
    begin
      i := i + n mod 8;
      n := n div 8;
    end;
  writeln(i mod 7);
end.

```

Решение

1. Анализируем алгоритм:

1) судя по числовым значениям в операциях **div** и **mod**, введенное число разбирается на цифры в восьмеричной системе;

2) в переменной i накапливается сумма выделенных из числа восьмеричных цифр;

3) выводится на экран не сама эта сумма, а остаток от ее деления на 7, и, согласно

Табл. 2

Десятичное число	Восьмеричная запись	Сумма восьмеричных цифр	Остаток от деления суммы на 7
99	143	8	1
98	142	7	0

условию, этот остаток должен быть равен нулю.

Итак, требуется найти *наибольшее двузначное десятичное число, у которого сумма цифр в восьмеричной записи кратна 7.*

2. Весь диапазон двузначных десятичных чисел – от 10 до 99. Поскольку нам нужно наибольшее возможное число, просмотр этого диапазона мы начинаем с 99 и далее по убыванию.

Построим таблицу (см. табл. 2).

Итак, мы нашли такое двузначное десятичное число n , для которого сумма цифр восьмеричной записи кратна 7. Это найденное число и есть ответ.

Ответ: 98

Очевидно, что переводить десятичное число в другую систему счисления (в данном случае в восьмеричную) нужно только один раз. Далее мы каждый раз уменьшаем десятичное значение на единицу, и соответствующее восьмеричное значение тоже достаточно уменьшать на 1. Только нужно при этом помнить правила вычитания в соответствующей системе счисления: например в восьмеричной системе $140_8 - 1 = 137_8$.

Задача 20. Определить наименьшее четырехзначное натуральное число, при вводе которого программа выведет на экран число 0.

```
var i, n: longint;
begin
  i := 0;
  readln(n);
  while (n > 0) do
  begin
    i := i + n mod 16;
    n := n div 16;
  end;
  writeln(i mod 15);
end.
```

Решение

Задача похожа на предыдущую.

1. Анализ алгоритма:

1) введенное число разбирается на цифры в шестнадцатеричной системе;

2) в переменной i накапливается сумма выделенных из числа шестнадцатеричных цифр;

3) выводится остаток от деления этой суммы на 15, и, согласно условию, этот остаток должен быть равен нулю.

Вывод: требуется найти *наименьшее четырехзначное десятичное число, у которого сумма цифр в шестнадцатеричной записи кратна 15.*

2. Весь диапазон четырехзначных десятичных чисел – от 1000 до 9999. Нам нужно наименьшее возможное число, поэтому просмотр этого диапазона начинаем с 1000 и далее по возрастанию.

Строим таблицу 3 (помним, чему равны значения шестнадцатеричных цифр: $A = 10$, $B = 11$, $C = 12$, $D = 13$, $E = 14$, $F = 15$).

Табл. 3

Десятичное число	Шестнадцатеричная запись	Сумма шестнадцатеричных цифр	Остаток от деления суммы на 15
1000	3E8	$3 + 14 + 8 = 25$	10
1001	3E9	$3 + 14 + 9 = 26$	11
1002	3EA	$3 + 14 + 10 = 27$	12
1003	3EB	$3 + 14 + 11 = 28$	13
1004	3EC	$3 + 14 + 12 = 29$	14
1005	3ED	$3 + 14 + 13 = 30$	0

Мы нашли такое четырехзначное десятичное число n , для которого сумма цифр шестнадцатеричной записи кратна 15. Это найденное число и есть ответ.

Ответ: 1005.

Задача 21. Определить количество двухзначных натуральных чисел, при вводе которых программа выводит на экран число 0.

```
var i, n: longint;
begin
  i := 0;
  readln(n);
  while (n > 0) do
  begin
    i := i + n mod 5;
    n := n div 5;
  end;
  writeln(i mod 4);
end.
```

Решение

Эта задача сложнее: потребуется просматривать весь диапазон допустимых чисел.

1. Анализ алгоритма:

1) введенное число разбирается на цифры в пятеричной системе;

2) в переменной i накапливается сумма выделенных из числа пятеричных цифр;

3) выводится остаток от деления этой суммы на 4, и, согласно условию, этот остаток должен быть равен нулю.

Вывод: требуется найти количество двухзначных десятичных чисел, у которых сумма цифр в пятеричной записи кратна 4.

2. Весь диапазон двухзначных десятичных чисел – от 10 до 99. Начнем просмотр этого диапазона с 10 и далее по возрастанию.

Начнем строить таблицу 4, но, в отличие от предыдущих двух задач, найдя нулевой остаток от деления, продолжаем запись.

Нетрудно заметить закономерность: впервые нулевой остаток встречается для $n = 12$, а далее ситуация повторяется для каждого четвертого числа. Остается выписать все такие двухзначные числа в ряд, каждый раз увеличивая значение на 4, и подсчитать их количество:

12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64, 68, 72, 76, 80, 84, 88, 92, 96 (а число 100 уже нам не годится – оно уже трехзначное!).

Количество этих чисел равно 22. Это и есть ответ.

Ответ: 22.

Задача 22. Укажите наименьшее из чисел x , при вводе которого программа выведет на экран пятизначное число.

```
var x, k, m, n: integer;
begin
  readln(x);
  m := x;
  n := 0;
```

Табл. 4

Десятичное число	Пятеричная запись	Сумма пятеричных цифр	Остаток от деления суммы на 4
10	20	2	2
11	21	3	3
12	22	4	0
13	23	5	1
14	24	6	2
15	30	3	3
16	31	4	0
17	32	5	1
18	33	6	2
19	34	7	3
20	40	4	0
21	41	5	1
22	42	6	2
23	43	7	3
24	44	8	0
25	50	5	1
...			

```

while x > 0 do begin
  k := x mod 4;
  n := 10*n + k;
  x := x div 4
end;
n := n + m;
writeln(n);
end.

```

Решение

1. Начнем с анализа алгоритма:

- введенное число x запоминается в переменной m ,
- значение n изначально равно нулю,
- в цикле сначала от числа справа отделяется очередная цифра (причем в *четверичной* системе счисления – см. разбор предыдущих задач), затем эта цифра прибавляется к n , а после этого обработанная цифра четверичной записи числа отбрасывается,
- после завершения цикла к n еще прибавляется ранее запомненное в переменной m исходное число x .

При этом очередная выделенная цифра четверичной записи прибавляется к предыдущему значению n , умноженному на 10. К чему это приводит?

Попробуем рассмотреть работу этого фрагмента на примере.

Пусть $x = 39_{10}$, что в четверичной системе счисления равно 213_4 . Выполняем трассировку алгоритма для четверичной записи числа (см. табл. 5).

Итак, для исходного числа 39_{10} , равного 213_4 , мы получили в итоге n , равное 312_4 .

Следовательно, рассматриваемый алгоритм (три рассмотренных оператора в цик-

ле), по сути, определяет четверичное представление исходного десятичного числа, а затем переписывает эти четверичные цифры в обратном порядке (!) уже как цифры нового десятичного числа n .

2. По условию, нам нужно найти наименьшее возможное число x , которое при таком преобразовании дает пятизначное число n . Но при таком преобразовании наименьшее значение x даст и наименьшее значение n .

Поэтому найдем сначала наименьшее возможное значение n , соответствующее условию задачи, то есть пятизначное, но без учета последнего оператора $n := n + m$.

Можно предположить, что это число 10000. Но вспомним, что рассматриваемый алгоритм «переворачивает» исходное число справа налево (меняет порядок цифр на обратный). И для числа n , равного 10000, исходное четверичное представление числа x тогда оказалось бы равным 00001, или просто 1. А в этом случае значение n не получилось бы пятизначным.

Поэтому минимальное значение n , которое нам подходит, равно 10001. Тогда (учитывая его «симметричность») такое четверичное представление соответствует исходному десятичному числу x , равному: $4^4 + 1 = 257_{10}$.

Итак, для исходного значения x , равного 257, по завершении выполнения цикла n будет равно 10001, и это действительно минимально возможное пятизначное число. Но у нас еще остается не рассмотренным оператор $n := n + m$. После его выполнения $n = 10001 + 257 = 10258$, и это число остается пятизначным.

Табл. 5

		x	n	k
Исходные значения		213	0	–
1-й проход цикла	k := x mod 4	213	0	<u>3</u>
	n := 10*n + k	213	$0*10+2 = \underline{3}$	3
	x := x div 4	<u>21</u>	3	3
2-й проход цикла	k := x mod 4	21	3	<u>1</u>
	n := 10*n + k	21	$3*10+1 = \underline{31}$	1
	x := x div 4	<u>2</u>	31	1
3-й проход цикла	k := x mod 4	2	31	<u>2</u>
	n := 10*n + k	2	$31*10+2 = \underline{312}$	2
	x := x div 4	<u>0</u>	312	2
Цикл завершен				

Но теперь нужно еще проверить: не могло ли быть так, что еще меньшее исходное значение x по завершении цикла давало четырехзначное значение n , и оно становилось пятизначным уже в результате прибавления к нему самого исходного числа.

Поэтому повторяем наше решение, выбрав меньшее значение n . А поскольку брать $n = 10000$ мы не можем (ранее было уже указано, почему), то мы берем предыдущее ему значение $n = 3333$.

Для такой четверичной записи исходное значение $x = 3 \cdot 4^3 + 3 \cdot 4^2 + 3 \cdot 4 + 3 = 255$. Проверим, станет ли сумма предполагаемого числа n и исходного значения x пятизначной? $3333 + 255 = 3588$. Нет, число осталось только четырехзначным. Значит, верным является наше первое предположение, и минимально возможное значение x , дающее в итоге пятизначное значение n , равно 257.

Ответ: 257.

В данной задаче завершающий оператор $n := n + m$; не повлиял на решение. Но выполнять описанную выше дополнительную проверку все-таки нужно. Скажем, если бы «финальным» оператором было $n := n + 27 \cdot m$; то у нас действительно бы четырехзначное число 3333 после такого суммирования становилось бы пятизначным, и правильным ответом было бы уже $x = 255$, а не 257.

Задача 23. Укажите наименьшее число x , для которого программа выводит число, большее 10200.

```
var x, k, m, n: integer;
begin
  readln(x);
  m := x; n := 0;
  while x > 0 do begin
    k := x mod 5;
    n := 10*n + k;
    x := x div 5
  end;
  n := n + m;
  writeln(N);
end.
```

¹ Идею решения подсказал Арсений Парфенов – выпускник школы № 1360 г. Москвы, в настоящее время – студент МГУ имени М.В. Ломоносова.

Решение¹

1. Анализ алгоритма:

- введенное число x запоминается в переменной m ,
- значение n изначально равно нулю,
- в цикле сначала от числа справа отделяется очередная цифра (в пятеричной системе счисления), эта цифра прибавляется к n , а затем обработанная цифра пятеричной записи числа отбрасывается,
- после завершения цикла к n еще прибавляется ранее запомненное в переменной m исходное число x .

При этом очередная выделенная цифра пятеричной записи прибавляется к предыдущему значению n , умноженному на 10, то есть происходит переписывание пятеричных цифр уже как десятичных и в обратном порядке.

2. Основная идея решения – в том, что хотя к полученному числу n и прибавляется исходное число x , но из-за того, что пятеричная запись более «громоздка» (требует больше цифр), а число x ищется минимальное, мы предполагаем, что этот «довесок» будет не очень большим.

3. Число, с которым в условии нужно сравнивать результат (10200), – пятизначное, а x , согласно нашему предположению, много к этому числу не добавляет. Значит, получаемое число n тоже должно быть пятизначным. (Можно это проверить. Наибольшее возможное четырехзначное число, состоящее из пятеричных цифр, равно 4444. При этом $4444_5 = 624_{10}$. В этом случае получаемое значение $n = 4444 + 624 = 5068 < 10200$. И все меньшие значения x тоже можно не проверять – они еще меньше.)

4. Пробуем взять первое (самое маленькое) допустимое пятизначное число – 10001 (число 10000 не подходит, так как при его «переворачивании» – записи в обратном порядке – получилось бы «00001», то есть не пятизначное число 1).

Число 10001_5 в десятичном представлении равно $5^4 + 1 = 626_{10}$.

В «перевернутой» записи тоже получается число 10001, на этот раз понимаемое как десятичное.

$10001 + 626 = 10627 > 10200$, что и требовалось.

Ответ: 626

Задача 24. Укажите наименьшее число x , для которого программа выводит число, большее 10300.

```
var x, k, m, n: integer;
begin
  readln(x);
  m := x; n := 0;
  while x > 0 do begin
    k := x mod 4;
    n := 10*n + k;
    x := x div 4
  end;
  n := n + m;
  writeln(N);
end.
```

Решение

1. Анализ алгоритма показывает, что отличие от предыдущей задачи только в том, что рассматривается представление исходного числа в четверичной системе счисления.

2. Так же, как и в прошлый раз, предполагаем, что добавление в конце программы числа x значение числа n меняет не сильно.

3. Число, с которым в условии нужно сравнивать результат (10300), – пятизначное, тогда получаемое число n тоже пятизначное. (Проверяем: $4444 + 255 = 4699 < 10300$.)

4. Если не учитывать прибавления x , то можно взять наименьшее значение n , боль-

шее 10300, равным 10301. Тогда «перевернутая» запись числа равна 10301, и если считать ее за пятеричное число, то получим $x = 4^4 + 3 \cdot 4^2 + 1 = 305$.

5. Конечно, это еще не ответ. «Правильное» число x будет меньше, поскольку получаемое число должно быть больше заданного 10300 только уже после прибавления x . Поэтому теперь перебором ищем наименьшее значение, которое удовлетворяет условиям задачи. Для этого строим таблицу (см. табл. 6), в которой сначала записываем итоговое (уже «перевернутое») пятизначное число, затем его исходную запись до «переворачивания», потом десятичный эквивалент этой записи, считая ее четверичной, а в заключение – вычисленное итоговое значение n .

Уже сейчас можно заметить:

- число в левой колонке не должно заканчиваться нулем,
- наименьшие значения получаются, если число в левой колонке заканчивается единицей, а не цифрой 2 или 3.

Поэтому далее заполняем таблицу только для таких чисел.

10211	11201	353	10564	да
10201	10201	289	10490	да
10131	13101	465	10596	да
10121	12101	401	10522	да
10111	11101	337	10448	да
10101	10101	273	10374	да
10031	13001	449	10480	да
10021	12001	385	10406	да
10011	11001	321	10332	да
10001	10001	257	10258	нет

Табл. 6

Предполагаемое «перевернутое» десятичное число n (до прибавления x)	«Исходное» четверичное число n	x	Сумма «перевернутого» десятичного n и исходного x	Сумма больше 10300?
10300	Недопустимо (при «переворачивании» не получается пятизначное число)			
10233	33201	993	11226	да
10232	23201	737	10969	да
10231	13201	481	10712	да
10230	Недопустимо (при «переворачивании» не получается пятизначное число)			
10223	32201	929	11152	да
10222	22201	673	10895	да
10221	12201	417	10638	да

Итак, мы нашли значение x , для которого в последней графе получается «нет».

Очевидно, брать еще меньшее число бесполезно: значение 10000 недопустимо, а далее уже получаются четырехзначные числа, которые, как мы уже показали в начале решения, нам не пригодны.

Попробуем взять в качестве x число, на 1 больше найденного «критического» значения 257. Проверяем: $258_{10} = 10002_4$; «переворачиваем» – получаем 20001; прибавляем исходное число 258 – получаем $20259 > 10300$.

Ответ: 258.

Особняком стоят задачи, в которых программы построены на базе алгоритма Евклида по поиску наименьшего общего делителя (НОД).

Задача 25. Укажите наименьшее такое число x , большее 100, при вводе которого программа выводит число 15.

```
var x, L, M: integer;
begin
  readln(x);
  L := x-30;
  M := x+30;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

Решение

Нетрудно увидеть, что это – алгоритм Евклида для вычисления НОД двух чисел. Но решить задачу можно, и просто исходя из общего анализа программы.

Из условия цикла **while** понятно, что значение M , по условию равное 15, выводится, когда оба числа (и L , и M) равны друг другу.

Тогда очевидно, что на предыдущем шаге цикла какое-то из чисел было равно $15 + 15 = 30$. Именно поэтому после вычитания из большего значения (30) меньшего (15) в результате мы получили тоже 15.

Еще на один шаг раньше большее число могло быть еще на 15 больше, то есть $30 + 15 = 45$.

Повторяя эти рассуждения и «двигаясь» от конца вычислений к началу, прибавляем к большему числу по 15: $45 + 15 = 60$; $60 + 15 = 75$; $75 + 15 = 90$; $90 + 15 = 105$.

Однако 105 – это и есть наименьшее число, большее 100. Значит, возможно, это и есть ответ.

Проверяем этот ответ, учитывая, что для $x = 105$ значение $L = 105 - 30 = 75$, а $M = 105 + 30 = 135$: вычисляем НОД (75, 135). Он равен 15. Следовательно, наше решение верно.

Ответ: 105.

А что если при проверке предполагаемого ответа выяснится, что получается не то значение НОД, которое нужно по условию? Тогда мы просто берем следующее по порядку значение в нашей цепочке. В данном случае – прибавляем к предполагаемому значению 105 еще 15, получая $x = 120$, и для него снова проверяем НОД вычисленных значений L и M .

Задача 26. Получив на вход число x , этот алгоритм печатает число M . Известно, что $x > 100$. Укажите наименьшее такое (то есть большее 100) число x , при вводе которого алгоритм печатает 11.

```
var x, L, M: integer;
begin
  readln(x);
  L := x - 21;
  M := x + 12;
  while L M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

Решение

Главное отличие здесь – в том, что перед началом цикла к исходному числу прибавляются и из него вычитаются *разные* значения. В этом случае ход решения несколько иной.

1. Известно, что в результате оба числа – и L , и M – равны 11, и это число есть НОД исходных чисел L и M . Следовательно, изначально оба они должны делиться на 11.

2. Как и в предыдущем случае, выпишем все возможные значения L , начиная с заданного числа 11 и каждый раз прибавляя это число 11, пока не получим первое же значение, превышающее 100:

11; 22; 33; 44; 55; 66; 77; 88; 99; 110.

3. Если, согласно программе, $L = x - 21$, то $x = L + 21$. Выбираем из только что выписанного ряда такое значение L , что при прибавлении к нему числа 21 получается наименьшее из возможных чисел, больших 100. Если мы возьмем 77, то $77+21 = 98$ – мало, а вот $88+21 = 109$ – это то, что нам надо. Тогда число 109 – это предполагаемое зна-

чение x .

4. Опять же, согласно программе, $M = x + 12$. Тогда предполагаемое значение $M = 109 + 12 = 121$.

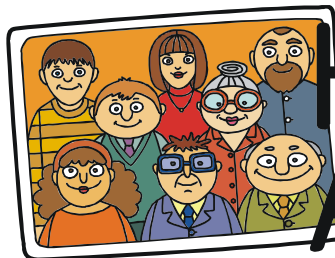
5. Для полученной пары чисел $L = 88$ и $M = 121$ вычисляем значение НОД: оно равно 11. Следовательно, наше решение верно, и ответом является $x = 109$.

Ответ: 109.

Если бы на пятом шаге решения мы получили бы другое значение НОД, то нужно было бы взять из ранее записанного ряда следующее, большее, значение L (99), повторить для него вычисление предполагаемых значений x и M , определить для полученных L и M значение НОД и сверить с заданным, и так – до нахождения нужного x .

Богомолова Ольга Борисовна,
доктор педагогических наук,
почетный работник сферы
образования Российской Федерации,
учитель информатики и
математики ГБОУ СОШ № 1360,
г. Москва,

Усенков Дмитрий Юрьевич,
Московский государственный
институт индустрии туризма
имени Ю.А. Сенкевича, г. Москва.



**Наши
Авторы**