Анализ алгоритма, содержащего цикл и ветвление

Галганова Дарья Анатольевна,
учитель информатики ГБОУ «Школа №185»

Место задания в ЕГЭ по информатике

№20 (первая часть)

Время на решение: 5 минут

Максимальное количество баллов: 1 балл

Уровень: Повышенный

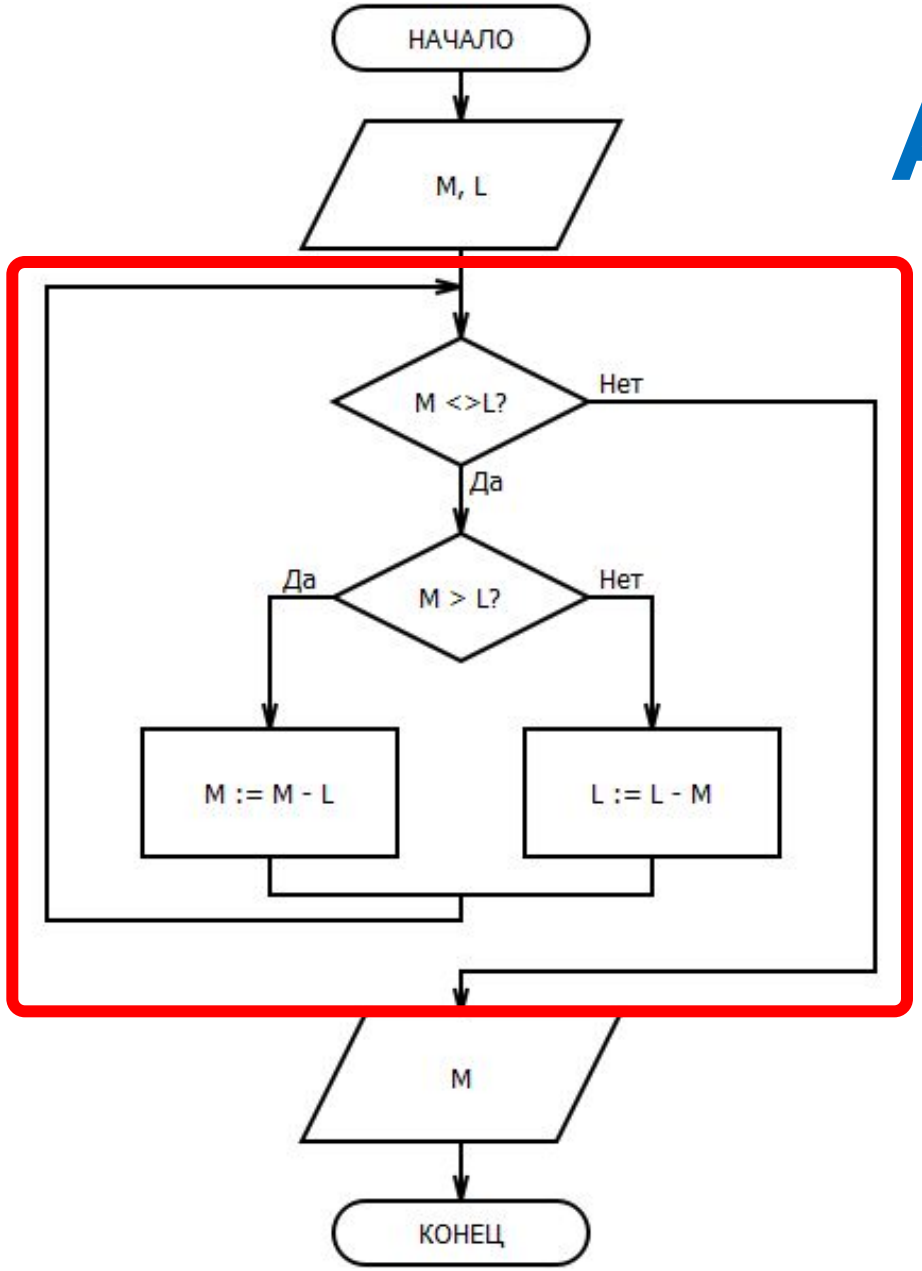
Типичные задания

Задание 20

- Выделение цифр числа
  - Десятичная система счисления
  - Другие системы счисления
- Алгоритм Евклида
- Другие

Общий подход к решению – «действуй с конца»

1. Что выводится? Чему равны M и L?
2. Определить тип алгоритма (цифры числа, алгоритм Евклида или другое)
3. Составить математическую модель.
4. Нахождение искомого числа.

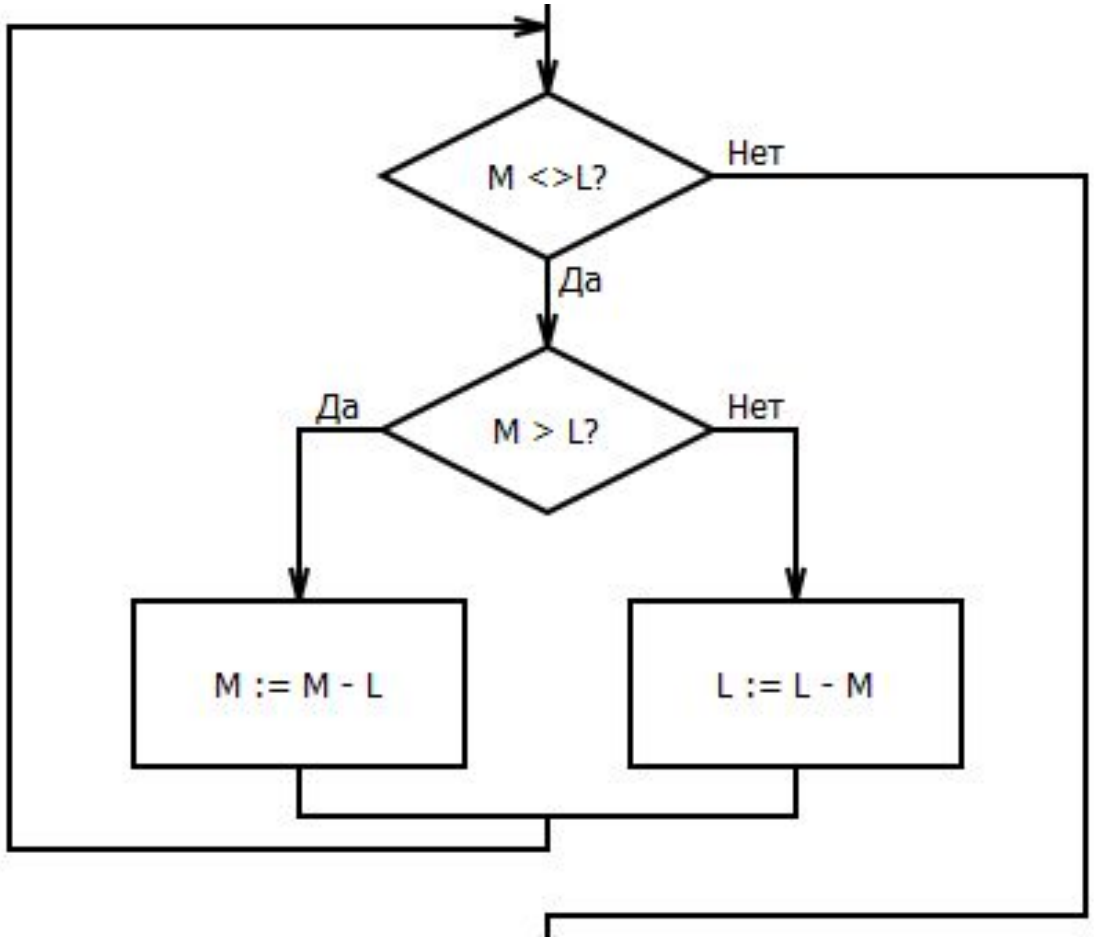


Алгоритм Евклида

Нахождение НОД двух чисел.

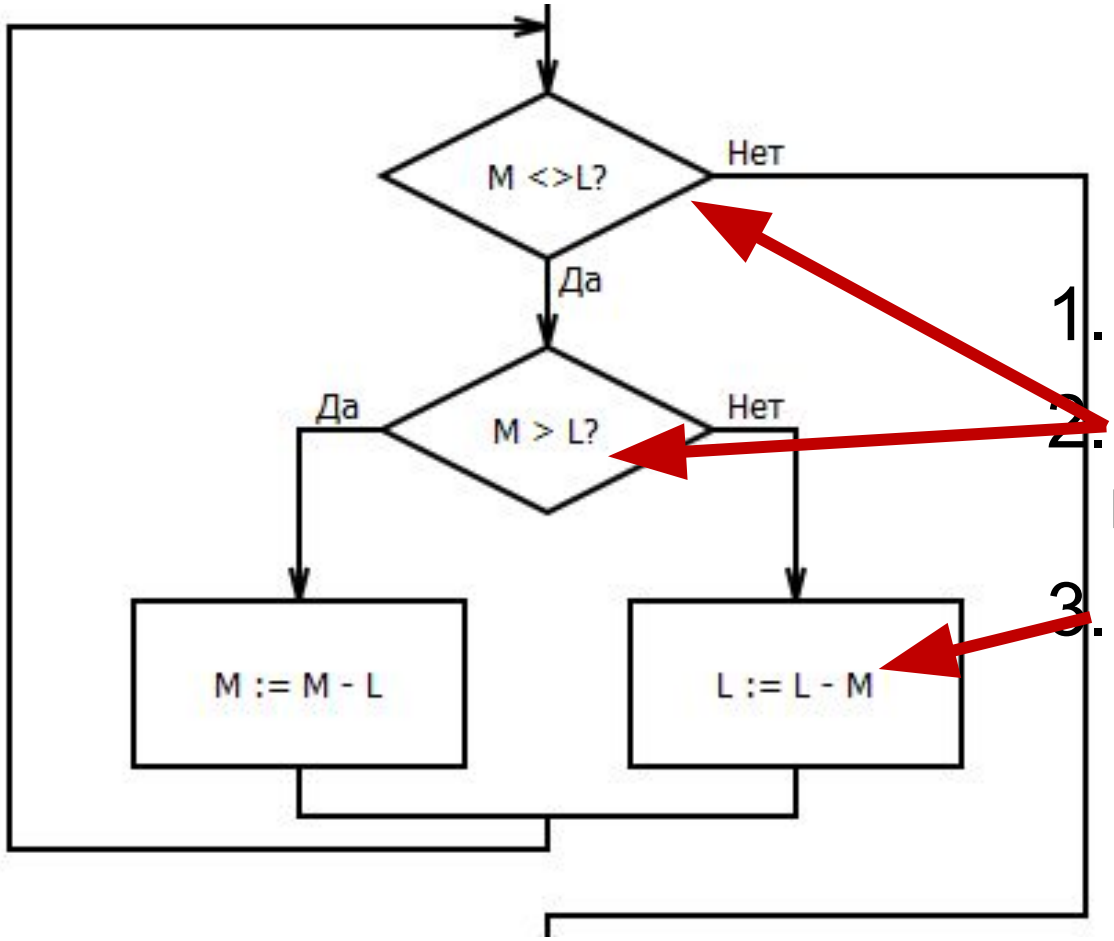
```
# пока числа M и L не равны
# сравнить числа
# большее заменить разницей
  большего и меньшего
```

Алгоритм Евклида



КАК УЗНАТЬ?

Алгоритм Евклида



Признаки:

1. В цикле всего две переменные!
2. В цикле два условия: условие цикла и поиск большего!
3. Наличие разностей!

КАК УЗНАТЬ?

Пример 1 (Открытый банк заданий ФИПИ):

Ниже на пяти языках программирования записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *M*. Известно, что *x* > 50.
Укажите **наименьшее** такое число *x*, при вводе которого алгоритм печатает 14.

```
var x, L, M: integer;
begin
  readln(x);
  L := x;
  M := 35;
  if L mod 2 = 0 then
    M := 28;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

Пример 1 (Открытый банк заданий ФИПИ):

Ниже на пяти языках программирования записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *M*. Известно, что *x* > 50.
Укажите **наименьшее** такое число *x*, при вводе которого алгоритм печатает 14.

```
var x, L, M: integer;
begin
  readln(x);
  L := x;
  M := 35;
  if L mod 2 = 0 then
    M := 28;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 14**.
2. **Тип** алгоритма? *Алгоритм Евклида*. Значит **14 = НОД (x, M).**

Пример 1 (Открытый банк заданий ФИПИ):

Ниже на пяти языках программирования записан алгоритм. Получив на вход число *x*, этот алгоритм **печатает число *M***. Известно, что *x* > 50.
Укажите **наименьшее** такое число *x*, при вводе которого алгоритм **печатает 14**.

```
var x, L, M: integer;
begin
  readln(x);
  L := x;
  M := 35;
  if L mod 2 = 0 then
    M := 28;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 14**.
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

14 ≠НОД (x, 35), значит M = 28.

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 1 *(Открытый банк заданий ФИПИ):*

Ниже на пяти языках программирования записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *M*. Известно, что *x* > 50. Укажите **наименьшее** такое число *x*, при вводе которого алгоритм печатает 14.

```
var x, L, M: integer;
begin
  readln(x);
  L := x;
  M := 35;
  if L mod 2 = 0 then
    M := 28;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 14**.
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

$$14 \neq \text{НОД}(x, 35), \text{ значит } M = 28.$$

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 1 *(Открытый банк заданий ФИПИ):*

Ниже на пяти языках программирования записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *M*. Известно, что *x* > 50.
Укажите **наименьшее** такое число *x*, при вводе которого алгоритм печатает 14.

```
var x, L, M: integer;
begin
  readln(x);
  L := x;
  M := 35;
  if L mod 2 = 0 then
    M := 28;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 14.**
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

$$14 \neq \text{НОД}(x, 35), \text{ значит } M = 28.$$

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 2 (К.Ю. Поляков):

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 35.

```
var x, L, M: integer;
begin
  readln(x);
  L := x-15;
  M := x+20;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

Пример 2 (К.Ю. Поляков):

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 35**.

```
var x, L, M: integer;
begin
  readln(x);
  L := x-15;
  M := x+20;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 35**.

Пример 2 (К.Ю. Поляков):

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 35.

```
var x, L, M: integer;
begin
  readln(x);
  L := x-15;
  M := x+20;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 35**.

2. **Тип** алгоритма? Алгоритм Евклида. Значит **35 = НОД (x, M).**

Пример 2 (К.Ю. Поляков):

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 35.

```
var x, L, M: integer;
begin
  readln(x);
  L := x-15;
  M := x+20;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 14**.
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

$$14 \neq \text{НОД}(x, 35), \text{ значит } M = 28.$$

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 2 (К.Ю. Поляков):

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 35.

```
var x, L, M: integer;
begin
  readln(x);
  L := x-15;
  M := x+20;
  while L <> M do
    if L > M then
      L := L - M
    else
      M := M - L;
  writeln(M);
end.
```

1. Определяем конечные значения переменных. **M = 35**.

2. **Тип** алгоритма? Алгоритм Евклида.

3. **Математическая модель**. 35 = НОД (L, M)=НОД (x-15, x+20).

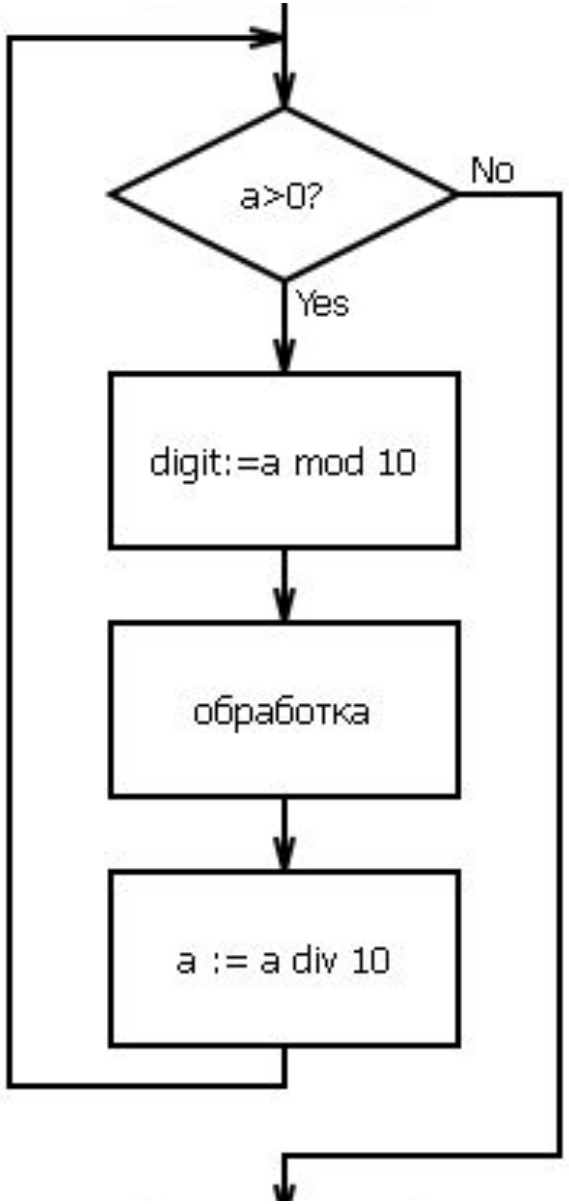
$$\begin{cases} x > 100, \\ L = (x-15) \vdots 35, \\ x - наименьшее \end{cases} \Rightarrow \begin{cases} L - наименьшее, \\ L \vdots 35, \\ L + 15 > 100. \end{cases}$$

4. Среди чисел, кратных 35, модели удовлетворяет число 105, т.о. L = 105, значит x = 120.

Ответ: 120.

Выделение цифр из числа

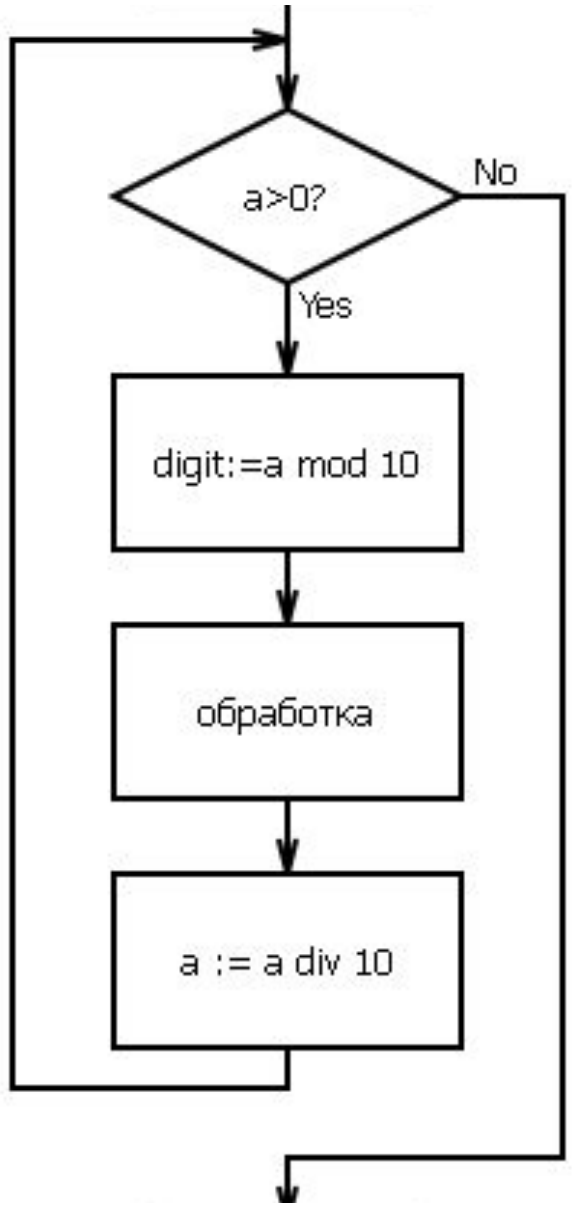
Выделение цифр в числе




```
# пока цифры в числе есть
# выделить последнюю цифру
# обработать её
# удалить цифру из числа
```

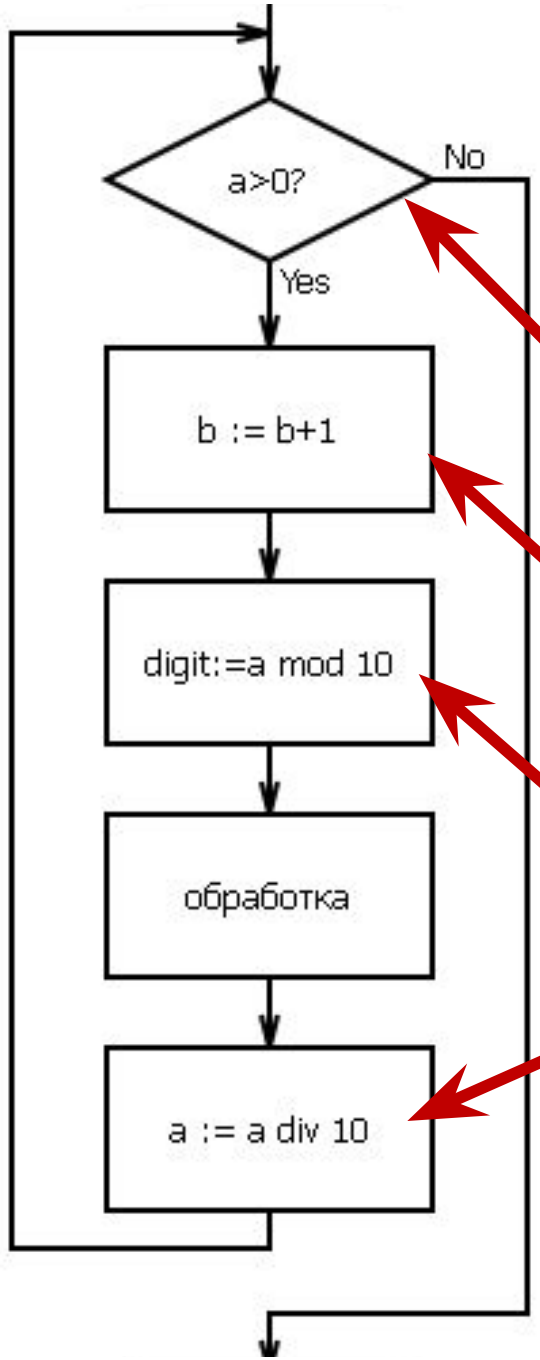
a mod b – остаток от деления числа *a* на число *b*
a div b – целая часть от деления числа *a* на число *b*

Выделение цифр в числе



КАК УЗНАТЬ?

Выделение цифр в числе



КАК УЗНАТЬ?

Признаки:

1. Цикл идет, пока введенная переменная больше нуля.
2. Считается количество выполнений цикла.
3. Наличие целочисленного деления.

Пример 3 (Сайт Информатик.БУ http://infbu.ru/):

Ниже на языке Pascal записан алгоритм. При каком наибольшем *n* программа выведет на экран сначала 3, а потом 2?

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

Общий подход к решению – «действуй с конца»

1. Что выводится? Чему равны M и L?
2. Определить тип алгоритма (цифры числа или алгоритм Евклида)
3. Составить математическую модель.
4. Нахождение искомого числа.

Пример 3 (Сайт Информатик.БУ http://infbu.ru/):

Ниже на языке Pascal записан алгоритм. При каком наибольшем *n* программа выведет на экран сначала 3, а потом 2?

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

Пример 3 (Сайт Информатик.БУ http://infbu.ru/):

Ниже на языке Pascal записан алгоритм. При каком наибольшем *n* программа выведет на экран **сначала 3, а потом 2**?

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

1. Определяем конечные **значения переменных:** ***a*** = *3*, ***b***=*2*.

Пример 3 (Сайт Информатик.БУ http://infbu.ru/):

Ниже на языке Pascal записан алгоритм. При каком наибольшем *n* программа выведет на экран сначала 3, а потом 2?

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

1. Определяем конечные значения переменных**:** ***a*** = *3*, ***b***=*2*.
2. Определяем **тип** алгоритма: *выделение цифр из числа.*

Пример 3 (Сайт Информатик.БУ http://infbu.ru/):

Ниже на языке Pascal записан алгоритм. При каком наибольшем *n* программа выведет на экран сначала 3, а потом 2?

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

1. Определяем конечные значения переменных: ***a*** = *3*, ***b***=*2*.
2. Определяем **тип** алгоритма: *выделение цифр из числа.*

   a – считает итерации □*считает разряды исходного числа,*

   b – результат обработки цифр числа,

   n div 10 – десятичная система счисления.

Пример 3 *(Сайт Информатик.БУ http://infbu.ru/):*

Ниже на языке Pascal записан алгоритм. При каком наибольшем *n* программа выведет на экран сначала 3, а потом 2?

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

1. Определяем конечные значения переменных: ***a*** = *3*, ***b***=*2*.
2. Определяем тип алгоритма: *выделение цифр из числа.*

   a – считает итерации □*считает разряды исходного числа,*

   b – результат обработки цифр числа,

   n div 10 – десятичная система счисления.
3. **Математическая модель:**

   a=3 →*n – трёхзначное* число.

   b=*b*+*c* =2 →*сумма цифр* числа *n* равна 2.

Пример 3 (Сайт Информатик.БУ http://infbu.ru/):

Ниже на языке Pascal записан алгоритм. При каком **наибольшем** *n* программа выведет на экран сначала 3, а потом 2?

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

1. Определяем конечные значения переменных: ***a*** = *3*, ***b***=*2*.
2. Определяем тип алгоритма: *выделение цифр из числа.*

   a – считает итерации □*считает разряды исходного числа,*

   b – результат обработки цифр числа,

   n div 10 – десятичная система счисления.
3. Математическая модель:

   $a=3 \rightarrow n$ – *трёхзначное* число.

   $b=b+c=2 \rightarrow$ *сумма цифр* числа *n* равна 2.
4. **Вывод**: возможные цифры искомого числа - (2,0,0) или (1,1,0).

   Т.к. нам надо *наибольшее*, то получаем число **200**.

Ответ: 200.

Пример 4 (Сайт К.Ю. Полякова):

Ниже записана программа. Получив на вход число x, эта программа печатает два числа. Укажите наибольшее из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 8.

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

Пример 4 (Сайт К.Ю. Полякова):

Ниже записана программа. Получив на вход число x, эта программа печатает два числа. Укажите наибольшее из таких чисел x, при вводе которых алгоритм печатает **сначала 3, а потом 8.**

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

1. Определяем **конечные значения переменных**: ***L*** *= 3,* ***M****=8.*

Пример 4 (Сайт К.Ю. Полякова):

Ниже записана программа. Получив на вход число x, эта программа печатает два числа. Укажите наибольшее из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 8.

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

1. Определяем конечные значения переменных: ***L*** *= 3,* ***M****=8.*
2. Определяем **тип** алгоритма: *выделение цифр из числа.*

Пример 4 (Сайт К.Ю. Полякова):

Ниже записана программа. Получив на вход число x, эта программа печатает два числа. Укажите наибольшее из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 8.

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

1. Определяем конечные значения переменных: ***L*** *= 3,* ***M****=8.*
2. Определяем **тип** алгоритма: *выделение цифр из числа.*

   L – считает итерации □*считает разряды исходного числа,*

   M – результат обработки цифр числа,

   x div 10 – десятичная система счисления.

Пример 4 (Сайт К.Ю. Полякова):

Ниже записана программа. Получив на вход число x, эта программа печатает два числа. Укажите наибольшее из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 8.

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

1. Определяем конечные значения переменных: ***L*** *= 3,* ***M****=8.*
2. Определяем тип алгоритма: *выделение цифр из числа.*

   L – считает итерации □*считает разряды исходного числа,*

   M – результат обработки цифр числа,

   x div 10 – десятичная система счисления.
3. **Математическая модель**:

   L=3 → *x – трёхзначное* число.

   M =8 - сумма четных цифр числа *x.*

Пример 4 (Сайт К.Ю. Полякова):

Ниже записана программа. Получив на вход число x, эта программа печатает два числа. Укажите **наибольшее** из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 8.

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

1. Определяем конечные значения переменных: ***L*** *= 3,* ***M****=8.*
2. Определяем тип алгоритма: *выделение цифр из числа.*

   L – считает итерации □*считает разряды исходного числа,*

   M – результат обработки цифр числа,

   x div 10 – десятичная система счисления.
3. Математическая модель:

   L=3 → *x – трёхзначное* число.

   M =8 - сумма четных цифр числа *x.*
4. **Вывод**: т.к. нам надо *наибольшее*, то это число, состоящее из цифр (8,9,9). Наибольшее такое число 998.

Ответ: 998.

Пример 5 (Сайт Решу ЕГЭ):

Ниже записана программа. Получив на вход число x, эта программа печатает два числа. Укажите **наибольшее** из таких чисел x, при вводе которых алгоритм печатает сначала 3, а потом 8.

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

1. Определяем конечные значения переменных: ***L*** *= 3,* ***M****=8.*
2. Определяем тип алгоритма: *выделение цифр из числа.*

   L – считает итерации □*считает разряды исходного числа,*

   M – результат обработки цифр числа,

   x div 10 – десятичная система счисления.
3. Математическая модель:

   L=3 → *x – трёхзначное* число.

   M =8 - сумма четных цифр числа *x.*
4. **Вывод**: т.к. нам надо *наибольшее*, то это число, состоящее из цифр (8,9,9). Наибольшее такое число 998.

Ответ: 998.

Пример 6 (Открытый банк заданий ФИПИ):

Ниже записана программа. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите наименьшее число *x*, при вводе которого алгоритм печатает сначала 3, а потом 6.

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x > 0 do
  begin
    M := M + 1;
    if x mod 2 <> 0 then
      L := L + 1;
    x := x div 2;
  end;
  writeln(L);
  writeln(M);
end.
```

Пример 6 (Открытый банк заданий ФИПИ):

Ниже записана программа. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите наименьшее число *x*, при вводе которого алгоритм печатает **сначала 3, а потом 6**.

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x > 0 do
  begin
    M := M + 1;
    if x mod 2 <> 0 then
      L := L + 1;
    x := x div 2;
  end;
  writeln(L);
  writeln(M);
end.
```

1. Определяем конечные **значения переменных:** ***L*** *= 3,* ***M****=6.*

Пример 6 (Открытый банк заданий ФИПИ):

Ниже записана программа. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите наименьшее число *x*, при вводе которого алгоритм печатает сначала 3, а потом 6.

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x > 0 do
  begin
    M := M + 1;
    if x mod 2 <> 0 then
      L := L + 1;
    x := x div 2;
  end;
  writeln(L);
  writeln(M);
end.
```

1. Определяем конечные значения переменных: ***L*** = *3*, ***M***=*6*.
2. Определяем **тип** алгоритма: *выделение цифр из числа.*

Пример 6 (Открытый банк заданий ФИПИ):

Ниже записана программа. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите наименьшее число *x*, при вводе которого алгоритм печатает сначала 3, а потом 6.

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x > 0 do
  begin
    M := M + 1;
    if x mod 2 <> 0 then
      L := L + 1;
    x := x div 2;
  end;
  writeln(L);
  writeln(M);
end.
```

1. Определяем конечные **значения переменных:** ***L*** *= 3,* ***M****=6.*
2. Определяем **тип** алгоритма: *выделение цифр из числа.*

   M – считает итерации □*считает разряды исходного числа,*

   L– результат обработки цифр числа,

   x div 2 – двоичная система счисления.

Пример 6 (Открытый банк заданий ФИПИ):

Ниже записана программа. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите наименьшее число *x*, при вводе которого алгоритм печатает сначала 3, а потом 6.

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x > 0 do
  begin
    M := M + 1;
    if x mod 2 <> 0 then
      L := L + 1;
    x := x div 2;
  end;
  writeln(L);
  writeln(M);
end.
```

1. Определяем конечные **значения переменных:** ***L*** *= 3,* ***M****=6.*
2. Определяем тип алгоритма: *выделение цифр из числа.*

   M – считает итерации □*считает разряды исходного числа,*

   L– результат обработки цифр числа,

   x div 2 – двоичная система счисления.
3. **Математическая модель:**

   M=6 →*двоичная запись* числа *x* содержит *шесть* разрядов.

   L=L+1, если *x* mod 2<> 0 →*количество нечетных цифр* числа *x*.

Пример 6 (Открытый банк заданий ФИПИ):

Ниже записана программа. Получив на вход число *x*, этот алгоритм печатает два числа: *L* и *M*. Укажите **наименьшее** число *x*, при вводе которого алгоритм печатает сначала 3, а потом 6.

```
var x, L, M: integer;
begin
  readln(x);
  L := 0;
  M := 0;
  while x > 0 do
  begin
    M := M + 1;
    if x mod 2 <> 0 then
      L := L + 1;
    x := x div 2;
  end;
  writeln(L);
  writeln(M);
end.
```

1. Определяем конечные **значения переменных:** ***L*** *= 3,* ***M****=6.*
2. Определяем тип алгоритма: *выделение цифр из числа.*

   M – считает итерации □*считает разряды исходного числа,*

   L– результат обработки цифр числа,

   x div 2 – двоичная система счисления.
3. Математическая модель:

   M=6 →*двоичная запись* числа *x* содержит *шесть* разрядов.

   L=L+1, если *x* mod 2<> 0 →*количество нечетных цифр* числа *x*.
4. **Вывод:** *x – наименьшее число*, в *двоичной* записи которого всего *шесть разрядов*, из них - *три нечетные цифры*.

(1,1,1,0,0,0) □ *x* = $100011_2 = 35_{10}$.

Ответ: 35.

Другие виды задач

(Сборник ЕГЭ. Информатика и ИКТ: типовые экзаменационные варианты.
С.С. Крылов, Т.Е. Чуркина, ФИПИ – школе, 2017 год)

Общий подход к решению – «действуй с конца»

1. Что выводится? Чему равны M и L?
2. Определить тип алгоритма (цифры числа или алгоритм Евклида)
3. Составить математическую модель.
4. Нахождение искомого числа.

Пример 7:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *L*. Укажите наибольшее нечётное число *x*, при вводе которого алгоритм печатает 125.

```
var x, L, M: integer;
begin
  readln(x);
  D := x;
  L := 17;
  M := 70;
  while L <= M do
  begin
    L := L + 2*D;
    M := M + D;
  end;
  writeln(L);
end.
```

1. Определяем конечные **значения переменных:** $L = 125$.

Пример 7:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *L*. Укажите наибольшее нечётное число *x*, при вводе которого алгоритм печатает 125.

```
var x, L, M: integer;
begin
  readln(x);
  D := x;
  L := 17;
  M := 70;
  while L <= M do
  begin
    L := L + 2*D;
    M := M + D;
  end;
  writeln(L);
end.
```

1. Определяем конечные значения переменных: ***L*** = *125*.
2. Определяем **тип** алгоритма: *прогрессия*.

Пример 7:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *L*. Укажите наибольшее нечётное число *x*, при вводе которого алгоритм печатает 125.

```
var x, L, M: integer;
begin
  readln(x);
  D := x;
  L := 17;
  M := 70;
  while L <= M do
  begin
    L := L + 2*D;
    M := M + D;
  end;
  writeln(L);
end.
```

1. Определяем конечные значения переменных**:** $\boldsymbol{L} = 125$.
2. Определяем тип алгоритма: *прогрессия.*
3. **Математическая модель**: Пусть n – количество итераций.

   $125 = 17 + 2 \cdot n \cdot D$, т.о. $n \cdot D = 54$.

Пример 7:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *L*. Укажите **наибольшее нечётное число *x*,** при вводе которого алгоритм печатает 125.

```
var x, L, M: integer;
begin
  readln(x);
  D := x;
  L := 17;
  M := 70;
  while L <= M do
  begin
    L := L + 2*D;
    M := M + D;
  end;
  writeln(L);
end.
```

1. Определяем конечные значения переменных. **M = 14**.
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

14 ≠НОД (x, 35), значит M = 28.

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 7:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *L*. Укажите **наибольшее нечётное число *x*,** при вводе которого алгоритм печатает 125.

```
var x, L, M: integer;
begin
  readln(x);
  D := x;
  L := 17;
  M := 70;
  while L <= M do
  begin
    L := L + 2*D;
    M := M + D;
  end;
  writeln(L);
end.
```

1. Определяем конечные значения переменных. **M = 14**.
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

14 ≠НОД (x, 35), значит M = 28.

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Ответ: 27.

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите наибольшее допустимое число *x*, при вводе которого алгоритм печатает 30.

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while ( D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите наибольшее допустимое число *x*, при вводе которого алгоритм **печатает 30.**

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while (D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

1. Определяем **конечные значения переменных:** $S = 30$.

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите наибольшее допустимое число *x*, при вводе которого алгоритм печатает 30.

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while (D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

1. Определяем конечные значения переменных: ***S*** = *30*.
2. Определяем **тип** алгоритма: *другой*.

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите наибольшее допустимое число *x*, при вводе которого алгоритм печатает 30.

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while (D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

1. Определяем конечные значения переменных: ***S*** = *30*.
2. Определяем **тип** алгоритма: *другой*.

S – зависит от чётности текущего *D*,

D div 2 – двоичная система счисления.

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите наибольшее допустимое число *x*, при вводе которого алгоритм печатает 30.

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while (D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

1. Определяем конечные значения переменных. **M = 14**.
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

14 ≠НОД (x, 35), значит M = 28.

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите наибольшее допустимое число *x*, при вводе которого алгоритм печатает 30.

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while (D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

1. Определяем конечные значения переменных. **M = 14.**
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

$$14 \neq \text{НОД}(x, 35), \text{ значит } M = 28.$$

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите **наибольшее** допустимое число *x*, при вводе которого алгоритм печатает 30.

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while (D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

1. Определяем конечные значения переменных. **M = 14.**
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

$$14 \neq \text{НОД}(x, 35), \text{ значит } M = 28.$$

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Пример 8:

Ниже записан алгоритм. Получив на вход число *x*, этот алгоритм печатает число *S*. Известно, что 100 ≤ x ≤ 200. Укажите **наибольшее** допустимое число *x*, при вводе которого алгоритм печатает 30.

```
var x, A, D, S: integer;
begin
  readln(x);
   A := 9;
  D := x;
  S := 0;
  while (D div 2)>0 do
  begin
    if (D mod 2) = 1 then
   S := S + 1
   else
   S := S + A;
   D := D div 2;
  end;
  writeln(S);
end.
```

1. Определяем конечные значения переменных. **M = 14.**
2. Тип алгоритма? Алгоритм Евклида. Значит **14 = НОД (x, M).**
3. **Математическая модель.**

$$14 \neq \text{НОД}(x, 35), \text{ значит } M = 28.$$

$$\begin{cases} x > 50 \\ x \text{ делится на } 14 \\ x \text{ НЕ делится на } 4 \end{cases}$$

Ответ: 120.

Общий подход к решению – «действуй с конца»

1. Что выводится? Чему равны конечные значения?
2. Определить тип алгоритма (цифры числа, алгоритм Евклида или другое)
3. Составить математическую модель.
4. Нахождение искомого числа.

Спасибо за внимание!