Наибольший общий делитель (НОД)

$a = bq + r, \qquad 0 \le r < |b| \qquad$ НОД(a,b) = НОД(b,r)

$$a = bq_0 + r_0, \qquad 0 \le r_0 < b$$
$$b = r_0 q_1 + r_1, \qquad 0 \le r_1 < r_0$$
$$r_0 = r_1 q_2 + r_2 \qquad 0 \le r_2 < r_1$$
$$r_1 = r_2 q_3 + r_3 \qquad 0 \le r_3 < r_2$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$r_{n-2} = r_{n-1} q_{n-1} + r_n \qquad 0 \le r_n < r_{n-1}$$
$$r_{n-1} = r_n q_n$$

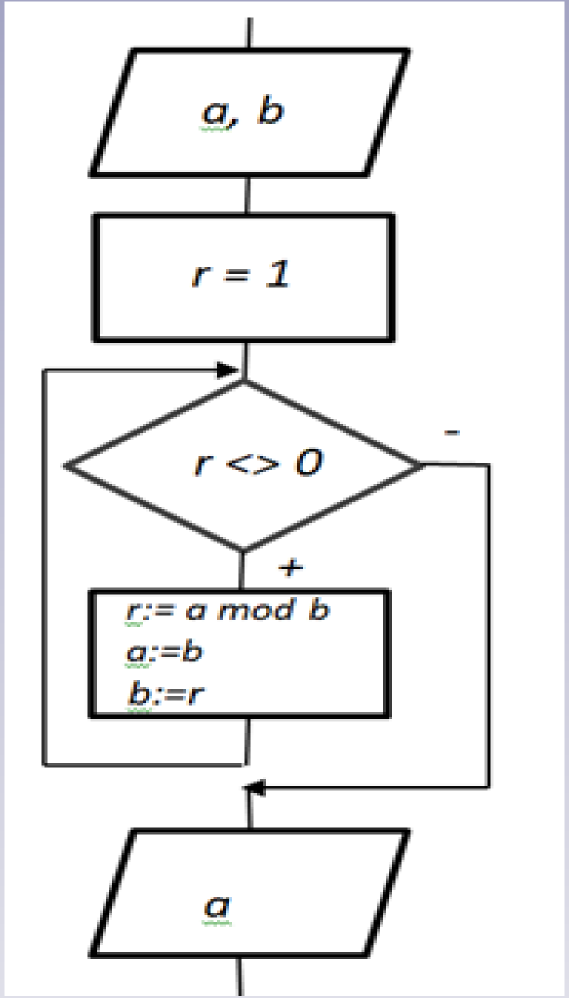
$$(a, b) = (b, r_0) = (r_0, r_1) = (r_1, r_2) = \cdots = (r_{n-1}, r_n) = (r_n, 0) = r_n$$

Вычислить (3009,894).

$$3009 = 894 \cdot 3 + 327$$
$$894 = 327 \cdot 2 + 240$$
$$327 = 240 \cdot 1 + 87$$
$$240 = 87 \cdot 2 + 66$$
$$87 = 66 \cdot 1 + 21$$
$$66 = 21 \cdot 3 + 3$$
$$21 = 3 \cdot 7$$

Дано: натуральные a и b $(b < a)$.

1. Вычислить $r = a \; mod \; b$
2. Если $r = 0$, то (a, b)= $b,$ СТОП
3. Если $r \neq 0,$ то заменить пару (a, b) парой (b, r) и перейти к п.1




```
var  a,b,r: integer;
begin
read(a,b);
r:=1;
while (r<>0) do
  begin
   r:= a mod b;
   a:=b;
   b:=r;
  end;
writeln(a);
end.
```

Можно предусмотреть особые случаи

- если два числа равны, то, их НОД также равен им.
- если какое-то из чисел равно нулю, то НОД будет равняться второму числу (НОД(0,0) не определен)
- В качестве входных данных могут использоваться только неотрицательные значения. Соответственно, для отрицательных можно использовать этот же метод, но взяв числа по модулю.

$$\text{НОК}[a,b] = \frac{a \cdot b}{\text{НОД}(a,b)}$$

НОД и НОК нескольких чисел можно вычислять последовательно

$$(a_1, a_2, a_3) = \big((a_1, a_2), a_3\big)$$
$$(a_1, a_2, a_3, a_4) = \big((a_1, a_2, a_3), a_4\big)$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.$$

$$(a_1, a_2, a_3, \ldots, a_n) = \big((a_1, a_2, \ldots, a_{n-1}), a_n\big)$$

$$[a_1, a_2, a_3] = [[a_1, a_2], a_3]$$
$$[a_1, a_2, a_3, a_4] = [[a_1, a_2, a_3], a_4]$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.$$

$$[a_1, a_2, a_3, \ldots, a_n] = [[a_1, a_2, \ldots, a_{n-1}], a_n]$$

Числа $a_1, a_2, a_3, \ldots, a_n$ называются взаимно простыми, если их наибольший общий делитель равен единице.

Операторы управления циклом

break – прерывает выполнение цикла, управление передается операторам, следующим за оператором цикла.

continue – прерывает выполнение очередного шага цикла и возвращает управление в начало цикла, начиная следующий шаг.

Что будет напечатано???

```
for n:=1 to 10 do
  begin
    if n mod 2 = 0 then continue;
    if n = 7 then break;
    writeln(n);
  end;
```

1, 3, 5.

Проверка числа на простоту

```
var n,i:uint64;
    f:boolean;
begin
  readln(n);
  i:=2;
  f:=true;
  while(i*i<=n) do begin
    if ((n mod i)=0) then begin
      f:=false;
      break;
    end;
    i:=i+1;
  end;
  if(f) then writeln('Простое')
else writeln('Составное');
end.
```

Вложенные циклы

Напечатать числа в виде следующей таблицы

```
3  3  3  3 3
3  3  3  3 3
3  3  3  3 3
3  3  3  3 3
```

```
for i:=1 to 5 do
    write(3, '   ');
```

```
for k:=1 to 4 do
{4 раза делаем то, что написано между begin’ом и end’ом}
  begin
    for i:=1 to 5 do
      write(3, '   ');  {Выводим одну строку}
    writeln;    {Переводим курсор на следующую строку}
end;
```

Вывести на экран числа в следующем виде

1

$3 \quad 5$

$5 \quad 7 \quad 9$

$7 \quad 9 \quad 11 \quad 13$

…………….

(количество строк n вводится с клавиатуры)

```
var i, j, n, k:  integer;
begin
  n:=5;
  for i:=1 to n do
  begin
    k:=i;
    for j:=1 to i do
    begin
      write(2*k-1, '  ');
      k:=k+1;
    end;
    writeln();
  end;
 end.
```

Основная теорема арифметики

Всякое натуральное число можно представить в виде произведения простых чисел, причем единственным образом (с точностью до порядка сомножителей)

$$n = p_1^{\alpha_1} p_2^{\alpha_2} p_3^{\alpha_3} \dots p_k^{\alpha_k}$$

$p_1, p_2, \dots, p_k$ - различные простые делители числа n

$\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_k$ - число их повторений в разложении числа

Такое разложение называется **ФАКТОРИЗАЦИЕЙ** числа.

Воспользуемся той же идеей, что и при проверке числа на простоту: будем перебирать числа вплоть до $\sqrt{n}$ и пытаться делить n на каждое из них; при этом в ответ добавляется этот делитель в степени, равной числу раз, на которое удалось поделить без остатка.

Единственная тонкость: если после окончания цикла делимое осталось больше единицы, то это оставшееся число – простой множитель, который надо также добавить в ответ, со степенью 1.

```
var i,n,k:integer;
f: boolean;
begin
  readln(n);
  i:=2;
while(i*i<=n) do begin
k:=0; f:=false;
while((n mod i)=0) do begin
  k:=k+1;
  if k=1 then f:=true;
  n:=n div i;
end;
if(f) then writeln(i,' ',k);
i:=i+1;
end;
if(n>1) then writeln(n,' ',1);
end.
```

Output Window

```
138000
2 4
3 1
5 3
23 1
```

Разложение факториала на простые множители

- разложение факториала N на простые множители содержит все простые числа от 2 до N
- никакое простое число большее N не входит в разложение $N!$

Простое число p входит в разложение $N!$ со степенью

$$\left\lfloor \frac{N}{p} \right\rfloor + \left\lfloor \frac{N}{p^2} \right\rfloor + \left\lfloor \frac{N}{p^3} \right\rfloor + \ldots (\text{пока } p^i \leq N)$$

Пример. $17! = 2^{\alpha_1} \cdot 3^{\alpha_2} \cdot 5^{\alpha_3} \cdot 7^{\alpha_4} \cdot 11^{\alpha_5} \cdot 13^{\alpha_6} \cdot 17^{\alpha_7}$

$$\alpha_1 = \left\lfloor \frac{17}{2} \right\rfloor + \left\lfloor \frac{17}{2^2} \right\rfloor + \left\lfloor \frac{17}{2^3} \right\rfloor + \left\lfloor \frac{17}{2^4} \right\rfloor = 8 + 4 + 2 + 1 = 15$$

Способы суммирования рядов

Дано: $n \in \mathbb{N}$, $x \in \mathbb{R}$. Вычислить

$$\sum_{k=1}^{n} \frac{x^k}{k!} = \frac{x}{1} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^{n-1}}{(n-1)!} + \frac{x^n}{n!}.$$

$$C = \frac{x_{k+1}}{x_k} = \frac{x^{k+1}}{(k+1)!} \cdot \frac{k!}{x^k} = \frac{x}{k+1}.$$

$x_1 = x \quad x_{k+1} = C\, x_k \quad$ k=1…n-1.

$$x_1 = x$$

$$k = 1 \quad x_2 = \frac{x}{2} \cdot x_1 = \frac{x^2}{2}$$

$$k = 2 \quad x_3 = \frac{x}{3} \cdot x_2 = \frac{x}{3} \cdot \frac{x^2}{2} = \frac{x^3}{6}$$

$$k = 3 \quad x_4 = \frac{x}{4} \cdot x_3 = \frac{x}{3} \cdot \frac{x^3}{6} = \frac{x^4}{24}$$

………………………………………………………………

```
var   k, n: integer;
      c, s, x, t :   real;
begin
  read(n, x);
  s:=x;   t:=x;
  for k:=1 to n-1 do
  begin
    c:=(x/(k+1));
    t:=c*t;
    s:=s+t;
  end;
 writeln(s);
end.
```

Вычислить значение функции $\sin(x)$ с точностью $eps = 10^{-6}$ с помощью бесконечного ряда

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots (-1)^n \frac{x^{2n-1}}{(2n-1)!} + \cdots$$

C_n – очередной член ряда, прибавляемый к сумме

$|\mathrm{C}_n| \leq eps$ – условие окончания вычислений

Получим рекуррентную формулу

$$C = \frac{x_{n+1}}{x_n} = \frac{(-1)^{(n+1)}\, x^{2(n+1)-1}}{(2(n+1)-1)!} \cdot \frac{(2n-1)!}{(-1)^n\, x^{2n-1}} = -\frac{x^2}{2n(2n+1)}.$$

```
var   x0,c,s,x, eps:real;
          MaxIter,n:integer;
          p: boolean;
begin
  MaxIter:=500;//ограничитель
количества итераций
  eps:=1e-10;
  read(x0);
  s:=x0;  c:=x0;  x:=x0;  n:=1;
  p:=true;//признак достижения точности
  while(abs(x)>=eps) do
  begin
    c:=-x0*x0/(2*n*(2*n+1));
    x:=c*x;//очередной член ряда
    s:=s+x;//накапливаем сумму
```

```
if(n>MaxIter) then
      begin
    p:=false;
    break;
    end;
    n:=n+1;
  end;
if (p) then begin
               writeln('s=',s);
               writeln('Контрольное значение=',sin(x0));
               end
    else writeln('Ряд расходится');
end.
```

Способы вычисления значения члена ряда

$$1+\frac{1}{2\cdot 3}+\frac{1}{3\cdot 5}+\frac{1}{4\cdot 7}+\cdots\frac{1}{N(2N-1)}+\cdots$$

Проще вычислять значения члена ряда по его общей формуле

$$R_N=\frac{1}{N(2N-1)}$$

Рекуррентная формула для сравнения $R_N=\frac{(N-1)(2N-3)}{N(2N-1)}R_{N-1}$

$$1+\frac{2}{1\cdot 2}+\frac{2^3}{1\cdot 2\cdot 3}+\cdots+\frac{2^N}{N!}+\cdots$$

Лучше воспользоваться рекуррентной формулой

$$R_N = \frac{2}{N} R_{N-1}$$

$$\frac{X^3\ln(X^3)}{1}+\frac{X^6\ln(X^6)}{1\cdot 2}+\frac{X^9\ln(X^9)}{1\cdot 2\cdot 3}+\cdots\frac{X^{3N}\ln(X^{3N})}{N!}$$

следует применить смешанный способ

$$A_N = X^{3N} \qquad A_N = X^3\cdot A_{N-1} \qquad A_1 = 1$$

$$B_N = N! \qquad B_N = N\cdot B_{N-1} \qquad B_1 = 1$$

$$R_N = \frac{A_N\cdot\ln(A_N)}{B_N}.$$