*Яковлева Дарья Владимировна,*
*Ульянцев Владимир Игоревич*

# ЗАДАЧА «НОД И НОК»

Этой статьей мы продолжаем цикл публикаций олимпиадных задач по информатике для школьников. Решение таких задач и изучение разборов поможет Вам повысить уровень практических навыков программирования и подготовиться к олимпиадам по информатике. В этой статье рассматривается задача «НОД и НОК», которая предлагалась XXII командном чемпионате школьников Санкт-Петербурга по программированию (идея задачи – Сергей Копелиович, подготовка тестов – Дмитрий Филиппов). Материалы этой олимпиады можно найти на сайте http://neerc.ifmo.ru/school/spb/.

## УСЛОВИЕ ЗАДАЧИ

Сережа очень любит математические задачи. Недавно на математическом кружке ему рассказали, что такое НОД и НОК.


Наибольший общий делитель (НОД) двух натуральных чисел $a$ и $b$ – такое максимальное число $z$, что $a$ делится на $z$ и $b$ делится на $z$. Например, НОД(24, 18) = 6. Наименьшее общее кратное (НОК) целых чисел $a$ и $b$ – такое минимальное число $z$, что $z$ делится на $a$ и $z$ делится на $b$. Например, НОК(24, 18) = 72.

Сережа сразу заметил, что может существовать несколько пар чисел с одинаковыми НОД и НОК. Теперь он заинтересовался вопросом: если заданы числа $a$ и $b$, насколько близко друг к другу могут быть два числа, у которых такие же НОД и НОК.

Помогите ему по заданным двум числам $a$ и $b$ найти такие числа $x$ и $y$, что

НОД$(a, b)$ = НОД$(x, y)$,

НОК$(a, b)$ = НОК$(x, y)$,

а их разность $y - x$ минимальна.

Формат входного файла

В первой строке входного файла находятся два натуральных числа $a$ и $b$ $(1 \le a, b \le 10^9)$.

Формат выходного файла

Выведите два натуральных числа $x$ и $y$ $(1 \le x \le y)$, таких, что НОД$(a, b)$ = НОД$(x, y)$, НОК$(a, b)$ = НОК$(x, y)$, а их разность $y - x$ минимальна.

Примеры входных и выходных данных

| gcm.in | gcm.out |
|---|---|
| 3 4 | 3 4 |
| 1 12 | 3 4 |

## РАЗБОР ЗАДАЧИ

*Факторизацией* натурального числа называется его разложение в произведение простых множителей. Факторизуем числа $a$ и $b$: пусть $a = p_1^{n1} \cdot p_2^{n2} \cdot \ldots \cdot p_k^{nk}$, $b = p_1^{m1} \cdot p_2^{m2} \cdot \ldots \cdot p_k^{mk}$, где все $p_i$ – простые.

Реализация факторизации чисел на языке *Pascal* приведена в листинге 1. Будем хранить как `factors[i]` $i$-ый простой делитель чисел, `powers[0][i]` – степень этого делителя в числе $a$, `powers[1][i]` – в числе $b$, `k` – текущее количество делителей. Отметим, что при реализации удобней нумеровать делители с нуля, а не с единицы.

НОД и НОК выражаются через простые множители следующим образом:

$$\text{НОД}(a,b) = p_1^{\min(n1,m1)} \cdot p_2^{\min(n2,m2)} \cdot \ldots \cdot p_k^{\min(nk,mk)} = \text{НОД}(x,y),$$

$$\text{НОК}(a,b) = p_1^{\max(n1,m1)} \cdot p_2^{\max(n2,m2)} \cdot \ldots \cdot p_k^{\max(nk,mk)} = \text{НОК}(x,y),$$

**Листинг 1.** Факторизация чисел $a$ и $b$

```
procedure add_divider(x: int64, pos: integer);
var
  have: boolean;
  i: integer;
begin
  have := false;
  for i := 0 to k - 1 do
    if factors[i] = x then begin
      powers[pos, i] := powers[pos, i] + 1;
      have := true;
    end;
  if not have then begin
    factors[k] := x;
    powers[pos, k] := 1;
    powers[1 - pos, k] := 0;
    inc(k);
  end;
end;

procedure factorization(a: int64, pos: integer);
var
  x: int64;
  i: integer;
begin
  x := a;
  i := 2;
  while (i * i <= a) do begin
    while (x mod i = 0) do begin
      add_divider(i, pos);
      x := x div i;
    end;
    inc(i);
  end;
  if x <> 1 then
    add_divider(x, pos);
end;
```

где $p_i$ – $i$-ый простой множитель в разложении чисел $a$ и $b$, $n_i$ – степень $p_i$ в разложении числа $a$, $m_i$ – в разложении числа $b$. Следовательно, для того, чтобы НОД($a$, $b$) был равен НОД($x$, $y$), степень вхождения $p_i$ в разложении на множители числа $x$ должна быть равна либо $n_i$ (а для $y - m_i$), либо $m_i$ (а для $y - n_i$).

Заметим, что из ограничений на $a$ и $b$ в условии задачи следует, что число делителей $k$ заведомо не превосходит 20. Значит, ограничение по времени позволяет перебрать все возможные варианты $x$ и $y$ и найти оптимальный ответ. Для этого рассмотрим и будем перебирать множество бинарных последовательностей. Условимся, что если $i$-ый символ в последовательности равен 0, то в $x$ входит $i$-ый делитель в степени $\max(n_i, m_i)$, а в $y$ – в степени $\min(n_i, m_i)$. Если $i$-ый символ равен 1, то наоборот.

Таким образом, каждой бинарной последовательности однозначно соответствуют $x$ и $y$, удовлетворяющие условию задачи. Ответом на поставленную задачу будет пара чисел $x$ и $y$ такая, что разность $y - x$ минимальна.

В листинге 2 приведена реализация перебора множества бинарных последовательностей и выбор наилучшего ответа. Ответ находится в переменных `x` и `y`. Напомним, что `1 << k` обозначает побитовый сдвиг единицы на $k$ разрядов влево; цикл `for i = 0 to (1 << k) - 1` перебирает множество бинарных последовательностей длины $k$. Значение выражения `i & (1 << j)` больше нуля, если в битовом представлении числа $i$ на $j$-ом месте стоит единица, и равно нулю в другом случае.

Время работы факторизации числа составляет $O(\sqrt{\max(x, y)})$, перебора последо-

**Листинг 2.** Перебор множества бинарных последовательностей и получение ответа

```
factorization(a, 0);
factorization(b, 1);
x := a;
y := b;
if x > y then begin
  t := x;
  x := y;
  y := t;
end;
for i := 0 to (1 << k) - 1 do begin
  cur_x := 1;
  cur_y := 1;
  for j := 0 to k - 1 do begin
    min_power := min(powers[0, j], powers[1, j]);
    max_power := max(powers[0, j], powers[1, j]);
    if i & (1 << j) then begin
      cur_x := cur_x * power(factors[j], min_power);
      cur_y := cur_y * power(factors[j], max_power);
    end else begin
      cur_x := cur_x * power(factors[j], max_power);
      cur_y := cur_y * power(factors[j], min_power);
    end;
  end;
  if (cur_y > cur_x) and (cur_y - cur_x < y - x) then begin
    y := cur_y;
    x := cur_x;
  end;
end;
```

вательностей – $O(2^k)$. Итоговое время работы составляет $O(\sqrt{\max(x,y)} + 2^k)$, что удовлетворяет ограничениям на время работы программы.

***Яковлева Дарья Владимировна,***
***студентка второго курса кафедры «Компьютерные технологии» НИУ ИТМО, член жюри Интернет-олимпиад по информатике,***

***Ульянцев Владимир Игоревич,***
***аспирант кафедры «Компьютерные технологии» НИУ ИТМО, член жюри Интернет-олимпиад по информатике.***