НОД и НОК

НОД

Наибольшим общим делителем (НОД) для двух целых чисел **m** и **n** называется наибольшее число, на которое делятся числа **m** и **n**. Наибольший общий делитель существует и однозначно определён, если хотя бы одно из чисел **m** или **n** не равно нулю.

Правила

Алгоритм был придуман Евклидом в Древней Греции более 2000 лет назад и основан на следующем правиле.

Для любых целых чисел **x, y > 0** выполняется равенство

НОД (x, y) = НОД (x % y, y)

Любое число, которое делит оба числа **x** и **y**, делит также и **x — y**, поэтому

НОД (x, y) ≤ НОД (x — y, y).

Аналогично, любое число, которое делит оба числа **x − y** и **y**, делит также и их сумму **x**, поэтому

НОД (x, y) ≥ НОД (x + y, y).

Алгоритм

Идея алгоритма отыскания наибольшего общего делителя заключается в том, чтобы отнимать от большего меньшее, пока числа не станут равны. Полученное число и является наибольшим общим делителем.

Пример

Например, необходимо определить наибольший общий делитель чисел 50 и 20.

1. Находим **50-20=30**. Из трех чисел 50, 20, 30 отбрасываем наибольшее.
2. Находим **30-20=10**. Из трех чисел 30, 20, 10 отбрасываем наибольшее.
3. Находим **20-10 = 10**. Из трех чисел 20, 10, 10 отбрасываем наибольшее.
4. Получаем **10=10**, значит это число является наибольшим общим делителем исходных.

Реализация НОД на языке С++

```
// Функция нахождения НОД
int NOD(int n1, int n2)
{
  int div;
  if (n1 == n2)   // если числа равны, НОД найден
    return n1;
  int d = n1 - n2; // Находим разность чисел
  if (d < 0)       // если разность отрицательная,
  {
    d = -d;     // меняем знак
    div = NOD(n1, d); // вызываем функцию NOD() для двух наименьших чисел
  }
  else      // если разность n1-n2 положительная
  {
    div = NOD(n2, d); // вызываем функцию NOD() для двух наименьших чисел
  }
  return div;
}
```

НОК

Наименьшее общее кратное (НОК) двух целых чисел **m** и **n** есть наименьшее натуральное число, которое делится на **m** и **n** без остатка.

Зная наибольший общий делитель (НОД) двух целых чисел **m** и **n**, их наименьшее общее кратное можно вычислить по такой формуле:

НОК = m * n / НОД (m, n)

Реализация НОК на языке С++

```
// Наименьшее общее кратное
int NOK(int n1, int n2)
{
  return n1*n2 / NOD(n1, n2);
}
```

Задача

Два натуральных числа *a* и *b* называются *взаимно простыми*, если их *наибольший общий делитель равен 1*.

Несколько натуральных чисел называются *попарно взаимно простыми*, если каждое из этих чисел является взаимно простым с каждым другим из них.

Например, 10, 11, 21 – попарно взаимно простые числа, а 10, 11, 25 таковыми не являются.

Сколько троек попарно взаимно простых чисел можно составить из двузначных натуральных чисел?

Решение

Для решения задачи понадобится вычислять НОД двух чисел.

При этом придется перебирать все возможные тройки двузначных натуральных чисел и для каждой тройки вычислять НОД для пар чисел, составляющих тройку.

Таких НОД для каждой тройки будет три, и *если все три НОД равны единице, то составляющие тройку натуральные числа будут взаимно и попарно простыми*.