АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
НАМЕРЕТЕ ПРАВИЛНАТА РАЗМЯНА

Решение. Ако купувачът получава x единици от стоката А и y единици от стоката В, заплащайки за това сума c, то се получава уравнението $ax + by = c$. Уравнението има решение тогава и само тогава, когато НОД(a, b) дели c.

Ако съществува решение, съкращаваме a, b, c на техния общ делител и с помощта на разширения алгоритъм на Евклид намираме (x', y'), за които $ax' + by' = 1$. Умножавайки последното равенство на c и полагайки $x_0 = cx'$ и $y_0 = cy'$, получаваме двойката (x_0, y_0), която се явява решение на $ax + by = c$.

Знаем, че всички решения на уравнението имат вида ($x_0 + kb$, $y_0 – ka$), където k е цяло число. Понеже решенията трябва да са неотрицателни, то са в сила неравенствата: $x_0 + kb \geq 0$, $y_0 – ka \geq 0$. От тук получаваме следните ограничения:

$k \geq \left\lceil -\frac{x_0}{b} \right\rceil$ и $k \leq \left\lfloor \frac{y_0}{a} \right\rfloor$.

С $\lfloor x \rfloor$ е означена цялата част на числото x, с $\lceil x \rceil$ - най-малкото цяло, не по-малко от x.

Броят на решенията на уравнението $ax + by = c$, за които $x \geq 0$, $y \geq 0$, е

$\left\lfloor \frac{y_0}{a} \right\rfloor - \left\lceil -\frac{x_0}{b} \right\rceil + 1.$

Пример. Да разгледаме първия пример от условието, където трябва да намерим броя на решенията на уравнението $2x + 3y = 17$ в цели неотрицателни числа. Числата 2 и 3 са взаимно прости, намираме x'и y', за които $2x' + 3y' = 1$. От разширения алгоритъм на Евклид имаме: $x' = -1$, $y' = 1$. Умножаваме ги по 17, получаваме

$x_0 = 17x' = -17$,

$y_0 = 17y' = 17$.

С частичното решение (x_0, y_0) = (-17, 17), може да се опишат всички решения на изходното уравнение: $x = -17 + 3k$,

$y = 17 – 2k$.

Те трябва да са неотрицателни, следователно $-17 + 3k \geq 0$,

$y = 17 – 2k \geq 0$.

Получаваме $3k \geq 17$, $17 \geq 2k$.

Откъдето следва $k \geq \left\lceil \frac{17}{3} \right\rceil = 6$ и $k \leq \left\lfloor \frac{17}{2} \right\rfloor = 8$.

Обединявайки ограниченията за k, получаваме: $6 \leq k \leq 8$, т.е съществуват три варианта на размяна, които може да получим, поставяйки в общото решение стойностите на k = 6, 7, 8. Например, при $k = 6$ получаваме решение (1, 5), при $k = 7$ решение (4, 3), а при $k = 8$ решение ще бъде двойката (7, 1).

За втория пример не съществува решение, защото НОД(10, 36) = 2 и 2 не дели 7.

Реализация. При изчисленията използваме **`long long`.**

```
#include <iostream>
#include <math.h>
using namespace std;

long long gcd(long long a, long long b)
{long long r;
```

```
   while (b!=0)
        {r=a%b;
         a=b;
         b=r;
        }
   return a;
  }

void extended_euclid(long long a, long long b, long long &x1, long long &y1)
{  int q, r, x2,  y2,  t;
   x1 = 1; y1 = 0;
   x2 = 0; y2 = 1;
   while (b != 0)
       {q = a/ b; //частно
        r = a % b;//остатък
        a=b;b=r;//заменяме двойката (a,b) с (b,r)
        //изчисляваме новите стойности  на x1 и x2
        t=x2;//запомняме старата стойност на x2
        x2=x1-q*x2;//изчисляваме новата стойност на x2
        x1=t;//новата стойност на x1 е = на старата стойност на x2
        //аналогично намираме новите стойности на y1 и y2
        t=y2;
        y2=y1-q*y2;
        y1=t;
      }
    }

    int main()
    {long long a,b,d,x,y,c,x0,y0,kmin, kmax, res;
     cin>>a>>b>>c;
```

Изчисляваме най-големия общ делител *d* на числата *a* и *b*.

```
     d=gcd(a,b);
```

Ако *d* дели *c*, съкращаваме входните стойности *a*, *b*, *c* на техния най – голямо общ делител *d*. По разширения алгоритъм на Евклид намираме двойката (*x*, *y*) – решение на уравнението *ax* + *by* = *c*.

```
     a=a/d;b=b/d;c=c/d;
     extended_euclid(a,b, x, y);
     x0=x*c; y0=y*c;
```

Търсим долната *kmin* и горната *kmax* граници за променливата *k*, откъдето и получаваме броя решения на уравнението *ax* + *by* = *c*. Понеже *a*, *b*, *x*, *y* са цели числа, за да не се загуби точност при изчисленията, умножаваме дробите $\left(-\frac{x}{b}\right)$ и $\left(\frac{y}{a}\right)$ по реалното число 1.0.

```
    kmin = (long long)ceil(-1.0 * x0 / b);
    kmax = (long long)floor(1.0 * y0 / a);
    res = kmax - kmin + 1;
```

Ако стойността на *res* не е 0, то я отпечатваме. В противен случай се извежда съобщение, че е невъзможно да се осъществи размяна.

```
if (res) cout<<res<<endl;
else cout<<"Impossible\n";
```

И накрая, ако *d* не дели *c*, то трябва отново да се изведе съобщението, че е невъзможно да се осъществи размяната.

Няма значение дали АЕ с изваждане, или с деление.

Автор: Кинка Кирилова-Лупанова