



## Janusz kupuje telewizor



### Wskazówki do rozwiązania zadania

W zadaniu należy obliczyć, ile ułamków postaci  $w/h$  spełnia warunki  $1 \leq w \leq a$  oraz  $1 \leq h \leq b$ . Dodatkowo spełniony musi być warunek  $w/h = x/y$ , gdzie  $x, y$  są zadanymi liczbami. Obliczamy zatem dwa wyrażenia:

$$\left\lfloor \frac{a}{x} \right\rfloor \text{ oraz } \left\lfloor \frac{b}{y} \right\rfloor$$

i wybieramy mniejsze z nich.

Przed tym obliczeniem należy jednak skrócić ułamek  $x/y$ , aby nie „zgubić” części rozwiązań. Największy wspólny dzielnik (NWD) liczb  $x$  i  $y$  obliczamy przy pomocy algorytmu Euklidesa. Ważne, by *najpierw* obliczyć NWD i podstawić go za jakąś pomocniczą zmienną, a potem podzielić przez nią  $x$  oraz  $y$  – tak, aby obydwie liczby podzielone były przez ten sam dzielnik.

W zadaniu należy używać typu danych *long long* (ośmiobajtowe liczby całkowite), który umożliwi działania na liczbach występujących w tym problemie. Dla uproszczenia zapisu warto wykorzystać instrukcję `typedef`.

### Kod programu w języku C++

```
#include <iostream>
using namespace std;

typedef long long ll;

ll NWD(ll a, ll b)
{
    if(b == 0) return a;
    return NWD(b, a % b);
}

int main()
{
    ll a, b, x, y;
    cin >> a >> b >> x >> y;
    ll d = NWD(x, y);
    x /= d; y /= d;
    cout << min(a / x, b / y) << endl;
    return 0;
}
```

## Uwagi

W programie należy użyć wersji algorytmu Euklidesa z resztą z dzielenia (ze względu na duży zakres danych i konieczność szybkiego obliczenia wyniku). Jest to metoda bez porównania szybsza od powszechnie znanej wersji algorytmu z odejmowaniem.

Typ danych `long long` (ośmiobajtowy) pozwala na używanie rzeczywiście dużych liczb (mniej więcej do tryliona, czyli  $10^{18}$ ). Należy jednak pamiętać, aby wybrać ten typ nie tylko dla końcowego wyniku, ale także dla liczb występujących w wyrażeniach w programie. Wtedy bowiem wszelkie operacje arytmetyczne (tutaj: dzielenie i znajdowanie reszty) będą wykonywane zgodnie z regułami tego typu, a nie zwykłego typu `int`.