Prosty algorytm: wyznaczanie Największego Wspólnego Dzielnika

Ze znajdowaniem największego wspólnego dzielnika dwóch liczb naturalnych mamy do czynienia przy skracaniu ułamków:

$$\frac{a}{b}=\frac{x\cdot c}{y\cdot c}$$
$$a, b, x, y, c \in \mathbb{N}$$

, przy czym chcemy, aby *c* było możliwie największe. Największe *c*, spełniające powyższe równanie, nosi nazwę *Największego Wspólnego Dzielnika* (*NWD*).

Najprostszą koncepcyjnie (choć kłopotliwą technicznie) metodą na znalezienie *NWD* jest rozkład obu liczb na czynniki pierwsze i znalezienie w rozkładach części wspólnej – jest to *NWD*.

Przykład:

Dla *a=234* i *b=186* mamy:

$$a=2\cdot 3\cdot 3\cdot 13$$
$$b=2\cdot 3\cdot 31$$
$$c=2\cdot 3$$

NWD wynosi zatem 6.

Istnieje prosty algorytm na wyznaczenie *NWD*, podany przez Euklidesa. Polega on na sekwencyjnym wyznaczaniu odpowiednich reszt z dzielenia, *NWD* jest ostatnią niezerową resztą. Algorytm Euklidesa prześledzimy na wybranym wcześniej przykładzie:

$$234=1\cdot 186+48$$
$$186=3\cdot 48+42$$
$$48=1\cdot 42+6$$
$$42=7\cdot 6+0$$

Najpierw dokonuje się dzielenia większej z liczb (234 - *dzielna*) przez mniejszą (186 – *dzielnik*). Iloraz wynosi 1, zaś reszta z dzielenia 48. Jeżeli reszta z dzielenia jest różna od zera, to należy kontynuować algorytm (wyznaczyć kolejną resztę), dokonując uprzednio następującego przyporządkowania: *dzielnej* ma być przyporządkowana poprzednia wartość *dzielnika*, a następnie należy przyporządkować *dzielnikowi* poprzednią wartość *reszty* (istotna jest kolejność przyporządkowań!). W naszym przykładzie *dzielna* wyniesie obecnie 186 (poprzedni *dzielnik*), zaś *dzielnik* 48 (poprzednia reszta z dzielenia). Po dokonaniu przypisania badamy po raz kolejny, ile wynosi reszta z dzielenia dla nowych wartości zmiennych. Gdy w danym sprawdzeniu otrzymamy resztę z dzielenia równą zero, algorytm się kończy, zaś wynik – NWD – zawarty jest w zmiennej *dzielnik* (gdyż tam znajduje się ostatnia niezerowa reszta z dzielenia). Zapiszmy ten algorytm w następujący sposób:

Dane wejściowe: *a*, *b* – liczby naturalne
Zmienne: *dzielna*, *dzielnik*, *reszta* – liczby naturalne

1. *dzielna* = większa z liczb (a, b);
2. *dzielnik* = mniejsza z liczb (a, b);
3. dopóki (reszta z dzielenia dzielna/dzielnik jest różna od zera)
   - *dzielna = dzielnik;*
   - *dzielnik = reszta;*
4. Wynikiem jest obecna wartość zmiennej *dzielnik*.

Zadanie: Napisać program realizujący algorytm Euklidesa.

Ponieważ warunkiem zakończenia pętli, zawartej w punkcie trzecim, jest to, aby reszta z dzielenia była równa zeru, możliwe jest skorzystanie z dwóch własności języka *C++*:

1. wyrażenie przypisania ma wartość, jest nią wartość wyrażenia, stojącego po prawej stronie operatora przypisania.
2. przy zamianie liczby na wartość logiczną, wartość zero traktowana jest jako logiczny fałsz, a wartości niezerowe jako logiczna prawda.

Możliwe jest zatem wstawienie wyrażenia obliczającego resztę jako warunku pętli, np. *while*.

Rozwiązanie:

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

int cgd(int a, int b);
int min(int a, int b);
int max(int a, int b);

int main()
{
int x, y, wynik;

   cout<<"Program wyznacza dla podanych liczb najwiekszy wspolny dzielnik\n";
   cout<<"Podaj pierwsza liczbe:";
   cin>>x;
   cout<<"Podaj druga liczbe:";
   cin>>y;
   if(wynik=cgd(x,y))
         cout<<"Najwiekszy wspolny dzielnik tych liczb to "<<wynik<<endl;
   else
         cout<<"Podales nieprawidlowe dane"<<endl;
}

int cgd(int a, int b)
{
int dzielna, dzielnik, reszta;

   if( a<1 || b<1 ) return 0;

   dzielna = max(a, b);
   dzielnik = min(a, b);
   while(reszta = dzielna % dzielnik)
   {
         dzielna = dzielnik;
         dzielnik = reszta;
   }
   return dzielnik;
}

int min(int a, int b)
{
   return a<b ?a:b;
}

int max(int a, int b)
{
   return  a>b ?a:b;
}
```