



Kocur do kocura, a będzie fura

```
#wczytywanie_danych #cin
#zmienna #deklaracja
#liczba_całkowita #int #suma
```

Napišemy program, który czyta dwie liczby całkowite i wypisuje ich sumę.

Wszystkie wczytane dane musimy przechować w wybranych przez siebie zmiennych. Ustalmy ich nazwy na x oraz y – tak jak w matematyce, pod tymi symbolami będą ukryte liczby wpisane przez użytkownika. Wolno nam użyć dowolną (rozsądną) ilość zmiennych, ale musimy „powiedzieć” systemowi, jakiego rodzaju dane będą w nich przechowywane.

W naszym przypadku będą to liczby całkowite, a po angielsku „liczba całkowita” to **integer** (wymawiaj: *intidżer*). W języku C++ dla oznaczenia liczb całkowitych używamy skrótu `int`, zatem poniższa instrukcja oznacza *zadeklarowanie* dwóch zmiennych całkowitych:

```
int x, y;
```

W poprzednim przykładzie użyliśmy instrukcji „`cout <<`” do wypisania tekstu. Jaki może być odpowiednik służący do *wczytywania* danych? Ano taki:

```
cin >> x >> y;
```

Przypominamy o prawidłowej wymowie: `cout` czytamy *si aut*, a `cin` wymawiamy *si in*. (Oczywiście literki „s” nie wymawiamy jak „ś”.)

Pozostaje tylko obliczenie sumy i wypisanie jej na ekranie. Cały program prezentuje się następująco:

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int x, y;
    cin >> x >> y;
    cout << x + y << '\n';
    return 0;
}
```

Jeżeli uruchomimy teraz ten program (w środowisku Codeblocks najlepiej przyciskiem F9), to na ekranie pojawi się okienko, w którym należy wpisać z klawiatury dwie liczby całkowite oddzielone odstępem (można także zapisać je w oddzielnych liniijkach).

Na przykład, jeżeli wpiszemy (i naciśniemy Enter):

```
24 37
```

program wypisze wynik dodawania:

```
61
```

Pojawia się pytanie, jak wielkie liczby możemy tu wpisać: typ danych `int` jest typem czterobajtowym, co w praktyce oznacza, że liczby x oraz y powinny się mieścić w zakresie od około -2 mld do 2 mld. Dla dociekliwych: spróbujcie wpisać wspomniane dwa miliardy jako wartość zarówno x , jak i y , po czym wytłumaczcie uzyskany wynik.

Jeśli zamiast liczby wpiszemy jakiś ciąg znaków (na przykład „a kuku!”), wtedy nasz program ani się nie zająknie, coś tam wyliczy i coś tam wypisze, ale wynik będzie bezsensowny. Zatem przy wprowadzaniu danych musimy sami zadbać o ich poprawność.

Dlaczego przed nazwą funkcji `main()` stoi słówko `int`? Otóż funkcja ta zwraca pewien kod w formie liczby całkowitej będący informacją dla systemu operacyjnego, czy w trakcie jej wykonywania nie doszło do błędu. Wartość `0` zwracana przy pomocy instrukcji `return` oznacza, że wykonanie programu przebiegło bez zakłóceń. Do tego zagadnienia jeszcze niebawem wrócimy.