

C++ pętla for (podstawy)

Co zrobić, gdy pewien fragment algorytmu trzeba wykonać większą ilość razy? Wielokrotnie powtórzyć w tekście programu? Dobra odpowiedź, ale niezbyt praktyczna bo zdarzają się algorytmy, które należy powtarzać milion razy...

Zadanie 1

Napisz program, który zsumuje liczby naturalne od 1 do 1000 oraz wprowadzi wynik na ekran:

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

int i, a;
main()
{
for(i=a=0; i<=1000; ++i)
{
a = a + i;
}
cout << "1+2+...+1000 = " << a;

system("PAUSE");
return EXIT_SUCCESS;
}
```

Pętla for() **zawsze** w swym ustroju ma dwa średniki, które tworzą trzy charakterystyczne pola.

Pierwsze pole jest *połem inicjowania*. Umieszczony tam algorytm wykonuje się tylko raz tuż przed obrotem pętli.

Drugie (środkowe) pole zawiera warunek, który ma udzielić odpowiedzi na pytanie: czy *wykonać kolejny obrót?*. Pole warunku jest wykonane wielokrotnie, tuż przed każdym obrotem pętli.

Trzecie (ostatnie) pole zawiera algorytm, który wykonuje się *na koniec każdego obrotu pętli*. Zazwyczaj jest to pole, w którym zwiększamy licznik obrotów pętli, czyli w naszym przykładzie zmienną *i*.

Zadanie 2

Działanie programu: podajemy długości boków *a* i *b*, a następnie program „rysuje” nam prostokąt o tych wymiarach. Jeżeli jednak podamy ujemne bądź też błędne wartości długości boków to program wygeneruje błąd w postaci tekstu.

1. Deklarujemy zmienne:

```
int a, b, i, j;
```

a – długość boku a

b – długość boku b

i, j – zmienne do pętli

.....

2. Wyświetlamy tekst powitalny lub też krótką instrukcję dotyczącą programu.

```
cout << "Wprowadz dlugosci bokow a i b prostokata \n" << endl;
```

\n oraz endl ma takie samo działanie – przeskok do początku nowego wiersza

.....

3. Program wyświetli tekst „a = ” a następnie jeśli wpisujemy obok jakąś cyfrę to pobierze ją (cin >> a;)

```
cout << "a = ";  
cin >> a;
```

.....

4. To samo robimy dla boku prostokąta b

```
cout << "b = ";  
cin >> b;
```

.....

5. Dodajemy odstęp aby wszystko nie zlewało się oraz po to aby program wyglądał w miarę estetycznie.

```
cout << "\n";
```

.....

6. Musimy napisać jeszcze **instrukcję warunkową if** na wszelki wypadek gdyby jakiś dowcipniś chciał podać ujemne, błędne lub też zbyt duże wartości boków prostokąta (np. -10, 100, 😊, mama itp.)

```
if ((a<0) || (b<0) || (a>30) || (b>30) || (1*a!=a) || (1*b!=b))  
{  
cout << "\nWprowadziles ujemne lub zbyt duze wartosci bokow\n";  
} else
```

|| - jest to operator OR (alternatywa)

!= – jest nierówne (np 1!=2)

Tłumaczenie na polski powyższego skryptu: **Jeśli:** bok a będzie mniejszy od zera **lub** bok b będzie mniejszy od zera **lub** bok a będzie większy od 30 **lub** bok b będzie większy od 30 **lub** bok a pomnożony przez 1 nie da wpisanej wartości **lub** bok b pomnożony przez 1 nie da tej samej wartości **to:** napisz „Wprowadziles ujemne lub zbyt duze wartosci bokow”.

else – inaczej...

.....

7. Pętla

```
for (i=0; i<a; i++){  
for (j=0; j<b; j++){  
cout << "#";  
}  
cout << "\n";  
}
```

Widzimy tutaj pętlę for zawartą w pętli for. Niezły mętlik ale tak naprawdę to bardzo proste. Program najpierw przypisuje i wartość 0 a następnie sprawdza kiedy ma się zatrzymać, a zatrzyma się wtedy gdy i będzie mniejsze od podanej wartości boku a. Następnie pętla zawierająca zmienną j powtórzy ją w linii ciągłej tyle razy na ile długi jest bok b.

Będzie tyle „#” w jednej linii na ile szeroki jest ten prostokąt, a szerokość ta powtórzy się tyle razy na ile wysoki on jest.

Aby wszystko nie było w linii ciągłej niezbędne jest dodanie odstępu na końcu pętli (cout << „\n”;

.....

8. Cały skrypt programu powinien wyglądać następująco:

```
#include <cstdlib>  
#include <iostream>  
using namespace std;  
  
int main()  
{  
int a, b, i, j;  
cout << "Wprowadz dlugosci bokow a i b prostokata \n" << endl;  
cout << "a = ";  
cin >> a;  
cout << "b = ";  
cin >> b;  
cout << "\n";  
if ((a<0) || (b<0) || (a>30) || (b>30) || (1*a!=a) || (1*b!=b))  
{  
cout << "\n Wprowadziles ujemne lub zbyt duze wartosci bokow\n";  
}else  
for (i=0; i<a; i++){  
for (j=0; j<b; j++){  
cout << "#";  
}  
cout << "\n";  
}  
cout << "\n";  
  
system("PAUSE");
```

```
return EXIT_SUCCESS;  
}
```