

# Wstęp do Informatyki i Programowania

## Laboratorium: Lista 1

### Instrukcje warunkowe i pętle

Przemysław Kobyłański

## Wprowadzenie

### Instrukcja warunkowa

Instrukcja warunkowa w języku C ma następującą postać:

```
if(Warunek)
{
    // instrukcje wykonywane gdy warunek jest prawdziwy
}
else
{
    // instrukcje wykonywane gdy warunek jest fałszywy
}
```

Jeśli w którymś z dwóch bloków ujętych w klamrowe nawiasy występuje tylko jedna instrukcja, to nawiasy otaczające tę jedną instrukcję można pominąć:

```
if(Warunek)
    // jedna instrukcja wykonywana gdy warunek jest prawdziwy
else
{
    // instrukcje wykonywane gdy warunek jest fałszywy
}
```

Jeśli blok po słowie kluczowym **else** nie zawiera instrukcji, to można go pominąć:

```
if(Warunek)
{
    // instrukcje wykonywane gdy warunek jest prawdziwy
}
```

### Przykłady

```
if(x > 0.0)
    y = sqrt(x);
else
    y = 0.0;
```

```

if(n > 0)
    printf("liczba jest dodatnia");

if(wiek >= 18)
    printf("osoba jest pełnoletnia");
else
    printf("osoba nie jest pełnoletnia");

```

## Instrukcja pętli

Jeśli pewien fragment programu ma wykonać się  $n$  razy, to można w tym celu użyć pętli **for**:

```

for(int i = 0; i < n; i++)
{
    // instrukcje powtarzane n razy dla i = 0, 1, ..., n-1
}

```

W powyższej pętli zostaje zadeklarowana zmienna całkowita  $i$ , która przyjmuje początkową wartość równą 0. Następnie dopóki spełniony jest warunek  $i < n$  powtarzane jest wykonywanie instrukcji wewnątrz bloku, przy czym po każdym ich wykonaniu obliczane jest  $i++$  (zwiększenie wartości zmiennej  $i$  o jeden).

Podobnie jak w przypadku instrukcji warunkowej, nawiasy klamrowe otaczające jedną instrukcję można pomijać.

Instrukcje pętli mogą być zagnieżdżone jedna w drugiej.

## Przykład

W wyniku wykonania poniższego fragmentu programu:

```

int n = 3;
for(int i = 0; i < n; i++)
{
    for(int j = 0; j <= i; j++)
    {
        printf("i = %d, j = %d\n", i, j);
    }
}

```

zostaną wydrukowane następujące wiersze:

```

i = 0, j = 0
i = 1, j = 0
i = 1, j = 1
i = 2, j = 0
i = 2, j = 1
i = 2, j = 2

```

Zwróć uwagę, że wywołanie funkcji `printf` jest jedyną instrukcją w bloku wewnętrznej pętli **for**, zatem nawiasy klamrowe wokół niej można pominąć.

Podobnie wewnętrzna pętla jest jedyną instrukcją w bloku zewnętrznej pętli **for**, zatem nawiasy klamrowe wokół niej można również pominąć.

```

int n = 3;
for(int i = 0; i < n; i++)
    for(int j = 0; j <= i; j++)
        printf("i = %d, j = %d\n", i, j);

```

## Zadania

### Zadanie 1 (5 pkt)

Napisz w C program, który korzystając z funkcji `printf`, wydrukuje następujący „magiczny” trójkąt:

```

A B R A K A D A B R A
  A B R A K A D A B R
    A B R A K A D A B
      A B R A K A D A
        A B R A K A D
          A B R A K A
            A B R A K
              A B R A
                A B R
                  A B
                    A

```

### Zadanie 2 (5 pkt)

Napisz w C program, który czyta trzy liczby rzeczywiste  $a$ ,  $b$  i  $c$  a następnie rozwiązuje równanie kwadratowe  $a \cdot x^2 + b \cdot x + c = 0$ .

Program powinien wydrukować albo informację, że nie ma w zbiorze liczb rzeczywistych rozwiązania, albo przedstawić jedno rozwiązanie gdy delta jest równa 0 albo wydrukować dwa rozwiązania gdy delta jest dodatnia.

### Zadanie 3 (5 pkt)

Napisz program w C, który czyta liczbę całkowitą  $n$  (można założyć, że będzie ona z zakresu od 1 do 20) a następnie drukuje gwiazdkami prostokąt złożony z  $n$  wierszy i  $2 \cdot n$  kolumn.

#### Przykład

Dla  $n = 5$ :

```

*****
*****
*****
*****
*****

```

#### Zadanie 4 (10 pkt)

Napisz program w C, który czyta liczbę całkowitą i drukuje gwiazdkami równoramienny trójkąt złożony z  $n$  wierszy (w pierwszym wierszu jedna gwiazdka a kolejne wiersze coraz szersze).

#### Przykład

Dla  $n = 6$ :

```
  *
 ***
*****
*****
*****
*****
```