

Programowanie 1

Zadanie 6

Piotr Błaszyński

14 listopada 2022

Napisać program przechowujący dane liczbowe (liczby całkowite) w tablicy i wyświetlający je w postaci histogramu. Na początku programu należy wczytać z klawiatury 10 liczb całkowitych z przedziału 1-9. Następnie wyświetlić dla tych wartości słupki o odpowiedniej wysokości. Można użyć pętli while lub for (opis for na końcu dokumentu).

Dla przykładowych danych podanych w tablicy, program powinien dawać następujące wyniki:

```
tab={1, 2, 3, 1, 2, 3, 5, 5, 4, 5};
#####
## #####
# #####
      #####
      #####
      ## #
```

Wersja na ocenę 5: program powinien losować nowe wartości w tablicy co 100ms, czyścić ekran, wyświetlać histogram, to wszystko należy wykonywać w pętli przez 15 sekund.

Losowanie:

```
#include <stdlib.h>
#include <time.h>

srand(time(NULL)); //inicjacja zmiennej pseudolosowej

int i = rand(); //losowanie liczby pseudolosowej
```

Czekanie przez określony czas (w ms) – Windows, VS:

```
#include <windows.h>

Sleep(100);
```

Czyszczenie ekranu - najbardziej prymitywna wersja:

```
system("cls");
```

Definicja zmiennej tablicowej wygląda następująco:

```
typ nazwa[ilosc_elementow]; //lub
typ nazwa[] = {wartosc1, wartosc2, ... wartoscN}; //lub
typ nazwa[n] = {wartosc1, wartosc2, ... wartoscN};
//Przykłady:
int t1[10];
int t2[] = {1, 2, 3, 4, 5};
int t3[5] = {1, 2, 3, 4, 5};
//Do tablicy odnosimy sie za pomoca indeksu:
printf("%d\n", t1[0]);
t2[3] = 4;
```

Należy pamiętać, że tablice w C/C++/C#/Javie mają indeksy z zakresu 0 - rozmiar-1. W pracy z tablicą ważne jest aby nie odwoływać się poza jej rozmiar. Należy pamiętać, że tablica nie zawiera informacji o sobie samej w tym swojego rozmiaru.

Pętla for ma następującą postać:

```
for ( init ; condition ; expression ) code_block
```

gdzie *init* jest albo jedną instrukcją wyrażeniową, albo listą oddzielonych przecinkami instrukcji wyrażeniowych, albo jest pominięte. Ten fragment służy najczęściej do ustawienia wartości początkowej zmiennej sterującej pętlą. Wyrażenie *condition* ma sprawdzaną wartość logiczną; jeśli zostanie ono pominięte, to warunek jest spełniony, czyli pętla działa do jej przerwania (*break*). Trzecia część (*expression*), jest jedną instrukcją wyrażeniową albo listą oddzielonych przecinkami instrukcji wyrażeniowych, albo jest pominięta. Służy najczęściej do zmiany wartości zmiennej sterującej pętlą. Jeżeli poszczególne elementy są pominięte, nie wolno pominąć średników ani nawiasów okrągłych. *code_block* jest to pojedyncze wyrażenie lub kilka wyrażeń w nawiasach klamrowych.

Przykładowe użycie pętli do wypełnienia tablicy kwadratami liczb i wypisania pierwszych pięciu par liczb (indeks *i* jego kwadrat) w kolejnych wierszach:

```
#include <stdio.h>
int main(void)
{
    int j;
    int t[10];
    for(j=0 ; j<10 ; j++)
    {
        t[j] = j * j;
    }
    for(int i=0 ; i<5 ; i++)
    {
        printf("%d->%d\n", i, t[i]);
    }
}
```

Zwracam uwagę, że w C++ i nowoczesnych wersjach C (C99 i C11 - ale nie C89) możliwe jest deklarowanie zmiennej indeksującej bezpośrednio w bloku *init*, tak jak zmienna *i* w powyższym przykładzie.