

Zajęcia 3 (22.03.2017)

Cele zajęć:

- pisanie funkcji w programie
- posługiwanie się tablicami

Funkcje.

Po co pisać własne funkcje w programie? Fragmenty kodu, które mogą stanowić pewną całość, a które używane będą wiele razy w programie można i należy zapisywać w postaci osobnych funkcji. Często nawet fragment programu, który nie będzie wiele razy powtarzany, dobrze jest zapisać w postaci odrębnej funkcji dla czytelności kodu programu głównego. Używanie funkcji wymusza pewną formalizację zapisu, funkcja bowiem musi mieć zdefiniowaną liczbę i typ parametrów, oraz zdefiniowaną wartość zwracaną. Funkcja może korzystać z własnych lokalnych zmiennych, które są niezależne od zmiennych zdefiniowanych w programie głównym.

Deklaracja i zarazem definicja funkcji może odbywać się następująco:

Przed funkcją „main” należy zapisać:

```
typ_zwracanej_wielkości nazwa_funkcji (typ_parametru
nazwa_parametru)
{
//tutaj ma być zawarte „ciało” funkcji
}
```

//parametrów może być w zasadzie dowolna ilość, a oddzielone muszą być przecinkami, na przykład:

```
float kwadrat (float podstawa)
{
//tutaj można deklarować zmienne wewnętrzne
//i wyliczyć to co trzeba
float wynik; // to jest zmienna wewnętrzna
wynik=podstawa*podstawa; //obliczenia
return(wynik) // instrukcja zwrócenia wartości
}
```

Następnie, już wewnątrz funkcji 'main' można użyć własnej funkcji np. tak:

```
delta=kwadrat (b) -4*a*c;
```

W przypadku szczególnym funkcja nie musi pobierać żadnego parametru i żadnej wartości zwracać, a za to może np. coś rysować na ekranie na przykład funkcja rysująca poziomą kreskę na ekranie (w konsoli):

```
void rysuj_linie (void)
{
int i;
for (i=0; i<12; i++)
{
    printf ("—");
}
printf ("\n");

return;
}
```

Tablice.

Tablica w języku C to zmienna która może przechowywać wiele wartości danego typu. Tablicę można wyobrazić sobie jako tabelkę w której można przechować określoną liczbę wartości, a zapisywanie i odczytywanie tych wartości dokonuje się poprzez podanie indeksu (numeru kolejnego) „komórki”. Pierwszy element ma indeks równy 0, następne elementy mają kolejne indeksy. Przykład:

```
float liczby[10]; //deklaracja tablicy liczb typu float
                  //mieszczącej 10 elementów

liczby[0]=12.1;   //wpisanie wartości pierwszej komórki tablicy
liczby[1]=1.1;    //wpisanie wartości kolejnej komórki tablicy
```

Oczywiście wpisywanie i odczytywanie wartości może odbywać się w pętli, np:

```
for (i=0; i<10; i++)
{
    liczby[i]=5; // wpisanie wartości 5 do wszystkich 10
                // komórek tablicy
}
```

UWAGA! Kompilator nie sprawdza czy indeksy tablicy nie wychodzą poza jej obszar (tzn czy indeksy nie są wyższe od zadanej wielkości tablicy). Czytanie, a szczególnie wpisanie wartości poza tablicą kończy się poważnym błędem podczas wykonania programu.

Do wykonania:

0. Napisać dowolną funkcję która pobiera jeden parametr i zwraca jedną wartość

1. Napisać funkcję, która pobiera dwa parametry: nazwę tablicy i ilość elementów do wyświetlenia, a następnie wypisuje te elementy na ekranie. Funkcję przetestować. Funkcję można zadeklarować następująco:

```
void wyswietl_tablice(float *tablica, int ilosc);
```

2. Zmodyfikować program który wczytuje liczby i podaje ich średnią, tak aby na końcu wyświetlał wszystkie podane liczby (używając wcześniej utworzonej funkcji).

3. Napisać funkcję, która liczy sumę liczb w tablicy. Deklaracja może być bardzo podobna do funkcji wyświetlającej liczby.

4. Zmodyfikować program tak, aby podawał także odchylenie standardowe, wzór poniżej:

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - x_s)^2}{N}}$$

gdzie x_i to kolejne elementy, a x_s to wartość średnia. Do obliczenia

pierwiastka użyć funkcji `sqrt(..)` z biblioteki `<math.h>`