

## Ćwiczenie nr 2

Temat: Zastosowania pętli iteracyjnych, instrukcji wyboru switch

Zagadnienia:

- Pętla for.
- Pętla while.
- Pętla do-while.
- Instrukcja wyboru switch.

### 1. Pętla for

Składnia pętli for:

```
for ( <wyrażenie1> ; <wyrażenie2> ; <wyrażenie3> )  
    instrukcja;
```

gdzie:

*wyrażenie1* - wyrażenie wykonywane jeden raz przed rozpoczęciem pętli;

*wyrażenie2* - wyrażenie wykonywane przed każdym rozpoczęciem powtarzania pętli, wyrażenie to musi być wyrażeniem logicznym (zwraca 0 lub 1)- jeżeli 1 - prawda - instrukcja lub grupa instrukcji w pętli jest powtarzana;

*wyrażenie3* - wyrażenie wykonywane przed powtórzeniem pętli;

Przykład:

```
for (int i=1 ; i<=n ; i++ ) printf("Wartosc i!=%d \n\r",i);
```

Przykładowa pętla na początku inicjuje zmienną 'i' typu int o wartości początkowej równej 1. Następnie sprawdzane jest wyrażenie logiczne  $i \leq n$ . Jeżeli jest prawdziwe wykonywane jest wyrażenie 3 a następnie instrukcja.

*Przykład 1. Program obliczający silnię liczby n - pętla for*

---

```
#include <stdio.h>  
#include <conio.h>  
  
void main()  
{  
    unsigned long n = 0, S = 1;  
    int i;  
    clrscr();  
    printf("Program oblicza silnie liczby n\n\r");  
    printf("Podaj wartosc n=");  
    scanf("%li",&n);  
  
    for (i=1;i<=n;i++)  
        S = S * i;  
  
    printf("Wartosc n!=%li\n\r",S);  
    getch();  
}
```

---

---

## 2. Pętla while

Składnia pętli while:

```
while (<wyrażenie logiczne>) instrukcja;
```

Możemy "przetłumaczyć" na:

*Dopóki wyrażenie logiczne jest prawdziwe powtarzaj instrukcję.*

Przykład:

```
float x = 1.2;
while (x <= 2.3)
{
    x+=0.1; // zwiększenie o 0.1
    printf("Wartosc x = %f \n\r",x);
}
```

Pętla będzie powtarzana dopóki x mniejsze lub równe 2,3.

## 3. Pętla do-while

Składnia pętli do-while:

```
do
    instrukcja;
while (<wyrażenie logiczne>);
```

Możemy "przetłumaczyć" na:

*Wykonuj instrukcję dopóki wyrażenie logiczne jest prawdziwe.*

Różnica pomiędzy while i do-while: pętla while sprawdza warunek przed wykonaniem instrukcji (jeżeli "nie prawda" instrukcja nigdy nie będzie wykonana). Pętla do-while wykonuje instrukcję - dopiero potem sprawdza wyrażenie logiczne (instrukcja wykonana będzie przynajmniej raz).

Przykład:

```
float z = 10.2;
do {
    printf("Wartosc x = %f \n\r",x);
    x-=0.2; // zmniejszenie o 0.2
} while (z >= -2.3);
```

Pętla będzie powtarzana dopóki x większe lub równe -2,3.

## 4. Instrukcja wyboru switch

Składnia instrukcji switch:

```
switch (<wyrażenie>) {  
  
    case <wartosc_1> : instrukcja_1;  
    case <wartosc_2> : instrukcja_2;  
    case <wartosc_3> : instrukcja_3;  
    ...  
    case <wartosc_n> : instrukcja_n;  
    default: instrukcja_default;  
}
```

Wyrażenie po słowie kluczowym switch jest wyrażeniem porządkowym (wartość całkowita). Wyrażenie to jest porównywane kolejno do wartości po słowie kluczowym case - jeżeli jest równe wykonana zostanie odpowiednia instrukcja.

*Przykład 2. Program wyznaczający nazwę słowną oceny w zależności od punktów z kolokwium*

---

```
#include <stdio.h> // podłączenie plików nagłówkowych  
#include <conio.h>  
  
void main() // program główny  
{  
    int pkt = 0, c = 0;  
    clrscr(); // czyszczenie ekranu  
    printf("\n\r Program wystawia ocene na podstawie punktow");  
    printf("\n\r Podaj ilosc punktow:");  
    scanf("%d", &pkt);  
    printf("\n\r Otrzymałeś ocene :");  
  
    switch (pkt) // instrukcja wyboru - początek  
    { case 0 :  
      case 1 :  
      case 2 : printf(" Niedostateczna - nie dobrze !!!"); break;  
      case 3 :  
      case 4 : printf(" Dostateczna"); break;  
      case 5 :  
      case 6 : printf(" Dostateczna plus"); break;  
      case 7 :  
      case 8 :  
      case 9 : printf(" Dobra"); break;  
      case 10 :  
      case 11 : printf(" Dobra plus"); break;  
      case 12 :  
      case 13 : printf(" Bardzo dobra - gratulacje !!!"); break;  
      default : printf("\n\r Jakis blad !!!"); break;  
    } // instrukcja wyboru - koniec  
}
```

## **5. Zadania do wykonania na zajęciach lub w domu:**

1. Napisać program drukujący na ekranie monitora tabliczkę mnożenia (od 0 do 9).
2. Napisać program drukujący na ekranie monitora tablicę funkcji sinus (od  $0^0$  do  $180^0$ ).
3. Napisać program wyznaczający sumę dzielników całkowitych danej z klawiatury liczby naturalnej.
4. Napisać program sprawdzający czy dana liczba naturalna jest liczbą pierwszą.