

Ćwiczenie nr 7

Temat: Struktury, operacje na strukturach.

Zagadnienia:

- Operacje na strukturach.
 - deklaracje typów strukturalnych (słowo kluczowe *struct*),
 - wprowadzenie danych do struktury,
 - deklaracje unii.

1. Definicja typu i deklaracja zmiennej strukturalnej

Składnia deklaracji struktury:

```
struct [<nazwa_typu_strukturalnego >] {  
    [<typ> <nazwa_zmiennej [,nazwa_zmiennej, ...]>] ;  
} [<nazwa_zmiennej_strukturalnej>];
```

Struktury są typem danych umożliwiającym przechowywanie w jednej zmiennej wielu wartości. Od tablic różnią się tym że mogą to być wartości różnych typów, np. deklaracja:

```
struct osoba {  
    char imie[20];  
    char nazwisko [20];  
    char nr_telefonu[20];  
    int  wiek, wzrost;  
} student1;
```

gdzie *osoba* to nowy typ strukturalny złożony z kilku składowych, *student1* to zmienna strukturalna typu *osoba*.

Możliwe jest także podanie danych początkowych zmiennej strukturalnej:

```
struct osoba {  
    char imie[20];  
    char nazwisko [20];  
    char nr_telefonu[20];  
    int  wiek, wzrost;  
} student1 = {"Zenon", "Zawierucha", "1234567890", 23, 182};
```

Operacje wejścia wyjścia mogą być wykonane podobnie jak dla zmiennych typów prostych lub łańcuchów tekstowych – przykład 1.

Przykład 1. Operacje wejścia/wyjścia dla zmiennej strukturalnej.

```
#include <stdio.h>  
  
int main()  
{  
    struct osoba {
```

```
    char imie[20];
    char nazwisko [20];
    char nr_telefonu[20];
    int  wiek, wzrost;
} student1;

printf("\n Podaj imie: ");
scanf("%s", student1.imie);
printf("\n Podaj nazwisko: ");
scanf("%s", student1.nazwisko);
printf("\n Podaj wiek: ");
scanf("%i", &student1.wiek);
printf("\n Podaj wzrost: ");
scanf("%i", &student1.wzrost);

printf("%s %s ma %i lat i %i cm wzrostu.\n",
       student1.imie, student1.nazwisko, student1.wiek, student1.wzrost);

system("pause");
return 0;
}
```

2. Słowo kluczowe „typedef”

Składnia definicji typu strukturalnego:

```
typedef struct {
    char imie[35];
    char nazwisko[35];
    float wiek;
    int nr_indeksu;
} osoba;
```

jest to **definicja typu strukturalnego** (słowo kluczowe **typedef** - *type definition*) zawierającego łańcuchy znaków, wartość zmiennoprzecinkową i wartość całkowitą. Po powyższej deklaracji nie istnieje jeszcze zmienna strukturalna (nie jest zarezerwowana pamięć) – jest to tylko **definicja typu**. Dopiero:

```
osoba student;
```

jest deklaracją zmiennej – zmienna „*student*” typu strukturalnego „*osoba*”.
Możliwe jest wtedy deklarowanie także tablic zmiennych strukturalnych:

```
osoba studenci[10];
```

Odwołanie do elementu tablicy, np.:

```
studenci[0].wiek = 23.5;
studenci[1].wiek = 22;
studenci[1].nr_indeksu = 123456;
```

3. Dynamiczny przydział pamięci na zmienną strukturalną

W aplikacjach, w których potrzebna będzie większa ilość zmiennych lub nie można określić ile rekordów będzie przetwarzanych przez program można zastosować dynamiczny przydział pamięci. Funkcja malloc przydziela pamięć wielkości podanej jako argument, np.:

```
punkt *p; /* wskaznik na strukture typu punkt */
p = (punkt *)malloc(sizeof(punkt));
```

gdzie sizeof(punkt) to wielkość rekordu w bajtach (liczba bajtów do przechowania danych struktury punkt). Operując na wskaźniku do zmiennej strukturalnej należy odwołać się do składowej struktury operatorem "->" zamiast "." (kropki), np.:

```
p->x = 10;
p->y = 10;
```

operacja pobrania danych ze standardowego wejścia może wyglądać tak:

```
scanf("%i", &(p->x));
```

Przykład 2. Dynamiczny przydział pamięci dla struktury.

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

typedef struct {
    int x, y;
} punkt;

int main(void)
{
    punkt *p; /* wskaznik na strukture typu punkt */

    p = (punkt *)malloc(sizeof(punkt)); /* przydzial pamieci */

    /* sprawdzenie czy wskaznik nie jest pusty (null) */
    if( !p ) {
        printf("Blad alokacji pamieci!");
        exit(1);
    }

    printf("Podaj wspolrzedna x punktu: ");
    scanf("%i", &(p->x));

    printf("Podaj wspolrzedna y punktu: ");
    scanf("%i", &(p->y));

    printf("\n x = %i, y = %i \n", p->x, p->y);
    free(p) ;

    system("pause");
    return 0;
}
```

4. Zadania do wykonania na zajęciach lub w domu:

1. Napisać program, który wykona operację na tablicy struktur:
 - utworzyć typ strukturalny „student” składający się ze składowych: numer indeksu, imię, nazwisko, miejscowość, ulica, nr domu/mieszkania, nr telefonu,
 - utworzyć tablicę statyczną 10-elementową o nazwie „studenci” składającą się ze zmiennych typu „student”,
 - program zapyta się o liczbę wprowadzanych studentów (maks. 10),
 - dane zostaną wprowadzone do tablicy z klawiatury (w pętli),
 - dane zostaną wydrukowane na konsoli tekstowej w „ładnej” i opisanej tabeli:

nr indeks	imie	nazwisko	miejscowosc
1235	Franek	Dolas	Wierzchowice
2745	Zenon	Jalocha	Koszalin
8235	Janek	Ziomal	Mielno
1533	Wiesiek	Starosta	Warszawa

2. Napisać program, który wykona działania identyczne jak w zadaniu 1 lecz wykona je na tablicy wskaźników 10-elementowej do zmiennych strukturalnych z dynamicznym przydziałem pamięci. Deklaracja tablicy wskaźników:

```
student *studenci[10]; /* 10 wskaźników na zmienne typu student */
```