

KORIŠĆENJE TYPEDEF DEKLARACIJE KOD STRUKTURA

Korišćenjem typedef deklaracije možemo definisati novi tip podatka i kada je u pitanju rad sa strukturama. Tako pri deklaraciji novih promenljivih izostavljamo pisanje ključne reči struct. U nastavku je dat primer gde smo korišćenjem typedef deklaracije definisali novi tip podatka koji smo zatim koristili u programu:

```
typedef struct Books
{
    char    title[50];
    char    author[50];
    char    subject[100];
    int     book_id;
} Book;
```

Stoga je moguće deklarirati strukturu promenljivu korišćenjem sledećeg iskaza:

```
Book book;
```

i zatim je kao takvu koristiti u programu:

```
int main( )
{
    Book book;

    strcpy( book.title, "C Programming");
    strcpy( book.author, "Marko Markovic");
    strcpy( book.subject, "C Tutorial");
    book.book_id = 6495407;

    printf( "Book title : %s\n", book.title);
    printf( "Book author : %s\n", book.author);
    printf( "Book subject : %s\n", book.subject);
    printf( "Book book_id : %d\n", book.book_id);

    return 0;
}
```

Strukture i funkcije

Strukture se mogu prosleđivati kao argumenti funkcije i mogu biti povratne vrednosti funkcija. Na sledećem listingu prikazan je primer programa u kome imamo definisan tip strukture pod imenom Student i dve funkcije. Funkcija ispisiStudent kao parametar prima tip Student i ispisuje vrednosti svih elemenata strukture. Funkcija kreiraj vraća povratnu vrednost tipa Student, a kao argumente uzima pojedinačne vrednosti elemenata strukture.

```
#include
#include
#include

typedef struct{
    char ime[50];
    char prezime[50];
    int brojPoena;
}Student;

void ispisiStudent(Student s);
Student kreiraj(char ime[], char prezime[], int brojPoena);

int main()
{
    Student s1, s2;
    strcpy(s1.ime, "Petar");
    strcpy(s1.prezime, "Peric");
    s1.brojPoena = 30;
    printf("Student 1:\n");
    ispisiStudent(s1);
    s2 = kreiraj("Bojan","Milosevic",45);
    printf("Student 2:\n");
    ispisiStudent(s2);
    return 0;
}

void ispisiStudent(Student s){
    printf("Ime studenta: %s\n",s.ime);
    printf("Prezime studenta: %s\n", s.prezime);
    printf("Broj poena %d\n", s.brojPoena);
}

Student kreiraj(char ime[50], char prezime[50], int brojPoena){
    Student s;
    strcpy(s.ime, ime);
    strcpy(s.prezime, prezime);
    s.brojPoena = brojPoena;
    return s;
}
```

NIZOVI STRUKTURA

Umesto da se skup složenih podataka čuva u nezavisnim promenljivama, nekada je bolje čuvati taj skup podataka u nizovima struktura

Veoma često u programerskim problemima postoji povezan skup složenih podataka. Umesto da se oni čuvaju u nezavisnim promenljivama (što bi vodilo programima teškim za održavanje) bolje je koristiti nizove struktura. Na primer, ako je potrebno imati podatke o imenima i broju dana i meseci u godini, moguće je te podatke čuvati u nizu sa brojevima dana i u (nezavisnom) nizu imena meseci. Bolje je, međutim, opisati strukturu mesec koja sadrži brojDana i ime:

```
struct mesec {
    char ime[10];
    int brojDana;
};
```

i koristiti niz ovakvih struktura:

```
struct opis_meseca meseci[13];
```

U prethodnom iskazu deklarisan je niz dužine 13 da bi se meseci mogli referisati po svojim rednim brojevima, pri čemu se početni element niza ne koristi. Moguća je i deklaracija sa inicijalizacijom (u kojoj nije neophodno navođenje broja elemenata niza):

```
struct opis_meseca meseci[] = {
    { "", 0 },
    { "januar", 31 },
    { "februar", 28 },
    { "mart", 31 },
    ...
    { "decembar", 31 }
}
```

ili

```
struct opis_meseca meseci[] = {
    "", 0,
    "januar", 31,
    "februar", 28,
    "mart", 31,
    ...
    "decembar", 31
}
```

Nakon navedene deklaracije, ime prvog meseca u godini se može dobiti sa `meseci[1].ime`, njegov broj dana sa `meseci[1].brojDana` itd. Kao i obično, broj elemenata ovako inicijalizovanog niza može se izračunati na sledeći način

```
sizeof(meseci)/sizeof(struct opis_meseca)
```

Zadaća

1. Opisati strukturu **apartman** koja sadrži sledeća polja: naziv, mesto, broj kreveta, cena. Napisati program u kome se unose podaci za n apartmana turističke agencije i određuje koliko je različitih mesta u ponudi i prikazati ih.
2. Opisati strukturu **ucenik** koja sadrži sledeća polja: ime, pol, razred i prosek. Napisati program u kome se unose podaci za n učenika i ispisuje odlične učenike sortirane po razredima