

Tema 6

Strukture i argumenti komandne
linije

Strukture

- Pravljenje složenih tipova
 - Od više primitivnih tipova (`int`, `double` itd.)
 - Od više složenih tipova (ranije formiranih struktura)
 - Kombinacije prethodna dva
- Ključna reč `struct`
- Pun naziv novog tipa: `struct novi_tip`
- Primereno za objedinjavanje zajedničkih osobina određene pojave ili objekta
 - Tačka (koordinate)
 - Geometrijska figura (broj, dužina stranica)
 - Student (broj indeksa, ime i prezime, prosek...)
 - Automobil (registarski broj, marka, tip, kubikaža...)

Primer 1

Predstavljanje tačke u dvodimenzionalnom prostoru, unos i ispis.

```
#include <stdio.h>

struct tacka_st
{
    double x;
    double y;
};

void unos_tacke(struct tacka_st *);
void ispis_tacke(struct tacka_st);

int main()
{
    struct tacka_st t1, t2;

    unos_tacke(&t1);
    unos_tacke(&t2);

    ispis_tacke(t1);
    ispis_tacke(t2);
}
```

```
    return 0;
}

void unos_tacke(struct tacka_st *pt)
{
    printf("Unesite x i y koordinatu tacke: ");
    scanf("%lf %lf", &(*pt).x, &(*pt).y);
}

void ispis_tacke(struct tacka_st t)
{
    printf("(%.2lf, %.2lf)\n", t.x, t.y);
}
```

- Operator dereferenciranja (*) ima niži prioritet u odnosu na operator člana strukture (.)
- (*pt).x i (*pt).y su primeri čestih konstrukcija u C programima
 - Kraći zapis (uvek korišćen): pt->x i pt->y

Primer 2

Rad sa nizovima struktura.

```
#include <stdio.h>

#define MAX_SIZE 30

struct tacka_st
{
    double x;
    double y;
};

void unos_tacaka(struct tacka_st *, int *);
void ispis_tacaka(struct tacka_st *, int);

int main()
{
    struct tacka_st tacke[MAX_SIZE];
    int n;

    unos_tacaka(tacke, &n);
    printf("\n");
}
```

```
    ispis_tacaka(tacke, n);  
  
    return 0;  
}  
  
void unos_tacaka(struct tacka_st *t, int *pn)  
{  
    int i;  
  
    do  
    {  
        printf("Unesite broj tacaka: ");  
        scanf("%d", pn);  
    } while(*pn <= 0 || *pn > MAX_SIZE);  
  
    for(i = 0; i < *pn; i++)  
    {  
        printf("t[%d] = ", i);  
        scanf("%lf %lf", &t[i].x, &t[i].y);  
    }  
}
```

```
void ispis_tacaka(struct tacka_st *t, int n)
{
    int i;

    printf("[");
    for(i = 0; i < n; i++)
    {
        if(i > 0)
        {
            printf(", ");
        }
        printf("%.21f, %.21f", t[i].x, t[i].y);
    }
    printf("]\n");
}
```

Preimenovanje tipa

- Puno ime tipa za strukture glasi `struct ime_strukture`
- Moguće je preimenovati tip i tako ga skratiti pomoću `typedef` naredbe

```
typedef struct tacka_st
{
    double x;
    double y;
} TACKA;
```

- Nadalje je moguće koristiti ime tipa `TACKA` umesto `struct tacka_st`
- Korišćenje `typedef` naredbe za preimenovanje tipa je opciono
 - Ako se ne koristi, potrebno je pisati puno ime tipa

Zadatak 1

Na osnovu koda iz prethodnog primera, učitati niz tačaka u ravni, maksimalno 30. Naći tačku koja je najbliža koordinatnom početku. Realizovati zadatak pomoću funkcija.

Primer zaglavlja funkcije koja nalazi tačku najbližu koordinatnom početku:

```
struct tacka_st najbliza_pocetku(struct tacka_st*, int);
```

- Za domaći uraditi isto za tačke u prostoru.

Zadatak 2

Onlajn striming platforma sadržaj koji nudi deli u posebne kategorije. Napraviti strukturu koja sadrži ime kategorije i trenutni broj gledalaca sadržaja. Dozvoliti korisniku da unese niz kategorija od najviše 30 elemenata. Sortirati kategorije po trenutnom broju gledalaca u opadajućem redosledu i ispisati ih na ekran terminala.

Primer vrednosti kategorija:

Games	403829
IRL	297405
Music	303699
Esports	498305
Creative	170493

Argumenti komandne linije

```
#include <stdio.h>

int main(int argc, char **argv)
{
    int i;

    printf("\nBroj argumenata je: %d\n", argc);

    printf("\nKonkretni argumenti su:\n");
    for(i = 0; i < argc; i++)
    {
        puts(argv[i]);
    }

    return 0;
}
```

Primer ispisa programa:

```
./a.out -o -x argument3 jedan dva tri
```

Broj argumenata je: 7

Konkretni argumenti su:

```
./a.out
```

```
-o
```

```
-x
```

```
argument3
```

```
jedan
```

```
dva
```

```
tri
```

- Prosleđivanje vrednosti programu prilikom pokretanja
- Omogućuje upravljanje programom spolja (primer gcc, gedit itd.)

Zadatak 1

Napisati program koji prima argument komandne linije koji se sastoje od crtice i jednog slova i jednu reč. U programu zahtevati od korisnika da, takođe, unese jednu reč i na osnovu zadate opcije uraditi sledeće:

- -s, spojiti uneti string i string argumenta komandne linije sa razmakom između njih i ispisati rezultujući string
- -p, uporediti uneti string i string argumenta komandne linije i ispisati da li su isti po sadržaju
- -n, ispisati da li se string argumenta komandne linije nalazi u unetom stringu

Primer poziva programa:

```
./a.out -p test
```