

Tema 6

Strukture, fajlovi i argumenti komandne linije

```
1  #include <stdio.h>
2
3  struct tacka {
4      float x;
5      float y;
6  };
7
8  void ispis(struct tacka t);
9
10 int main() {
11     struct tacka t1, t2;
12
13     printf("Unesite X i Y koordinatu tacke 1: ");
14     scanf("%f %f", &t1.x, &t1.y);
15
16     printf("Unesite X i Y koordinatu tacke 1: ");
17     scanf("%f %f", &t2.x, &t2.y);
18
19     ispis(t1);
20     ispis(t2);
21
22     return 0;
23 }
24
25 void ispis(struct tacka t) {
26     printf("\n ( %.2f, %.2f)\n", t.x, t.y);
27 }
```

Zadatak 1

- Učitati niz tačaka u ravni (maksimalno 30). Naći tačku koja je najbliža koordinatnom početku. Koristiti funkcije.
- Uraditi isto u prostoru.
- Za domaći: naći tačku koja je najudaljenija.
- Učitati dva kompleksna broja, realizovati sabiranje i oduzimanje kompleksnih brojeva.
- Za domaći realizovati vektorski proizvod.

Primer 2

Napisati program kojim se čitaju broj fudbalskih ekipa, za svaku ekipu: naziv, broj osvojenih bodova i gol razlika, a zatim se ispisuje tabela. Ako dve ekipe imaju isti broj bodova bolja je ona koja ima veću gol razliku. Da bi program bio prostiji čitanje, ispisivanje i sortiranje struktura realizovati pomoću funkcija.

```

#include <stdio.h>
#define MAXIME 30
#define MAXEKIPA 20
struct ekipa
{
    char ime[MAXIME];
    unsigned bodovi;
    int razlika;
}
void ispisi(struct ekipa tabela[], int n);
void ucitaj(struct ekipa tabela[], int *n);
void sortiraj(struct ekipa tabela[], int n);

int main() {
    int n;
    struct ekipa tabela[MAXEKIPA];
    ucitaj(tabela, &n);
    sortiraj(tabela, n);
    ispisi(tabela, n);
    return 0;
}

```

```

void ucitaj(struct ekipa tabela[], int *n){
    int i;
    do{
        printf("Unesite broj ekipa:\n");
        scanf("%d",&n);
    }while(*n<=0 || *n>MAXEKIPA);

    for(i=0;i<*n;i++)
    {
        printf("Ekipa [%d]: ",i);
        scanf("%s %u %d",tabela[i].ime, &tabela[i].bodovi, &tabela[i].razlika);
    }
}

void ispisi(struct ekipa tabela[], int n){
    int i;

    for(i=0;i<n;i++)
    {
        printf("Ekipa [%d]: ",i);
        printf("%s %u %d\n",tabela[i].ime, tabela[i].bodovi, tabela[i].razlika);
    }
}

```

```

void razmeni(struct ekipa *a, struct ekipa *b){
    struct ekipa pom;
    pom = *a;
    *a = *b;
    *b = pom;
}

void sort(struct ekipa t, int n){
    int i,j;
    for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
            if((t[i].bodovi<t[j].bodovi)||((t[i].bodovi==t[j].bodovi) && (t[i].razlika<t[j].razlika)))
                razmeni(&t[i],&t[j]);
}

```

Zadatak 2

Napiši program koji učitava niz od n elemenata niza strukture ličnost, koja sadrži sledeća polja: ime, adresa, dan rođenja, mesec rođenja i godina rođenja. Prikazati osobe koje su u horoskopu bik.

Zadatak 3

Napisati program koji učitava niz od n elemenata niza strukture auto, koja sadrži sledeća polja:

- Marka automobila (jedna reč, do 20 karaktera)

- Kubikaža (prirodan broj)

- Godište (prirodan broj)

Na osnovu zadate kubikaže pronaći najnoviji auto sa kubikažom ne većom od zadate i ispisati njegove podatke na standardni izlaz.

Zadatak 3

Ako korisnik unese vrednost kubikaže 2000 i niz podataka o automobilima:

AlfaRomeo 1200 2005

BMW 2200 2011

AUDI 1900 2010

Zastava 1600 1998

Ferrari 2500 2013

Na standardnom izlazu treba ispisati:

Model: AUDI

Kubikaza: 1900

Godiste: 2010

Fajlovi i argumenti komandne linije

Argumenti komandne linije

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

```
int main( int brArg, char *arg[ ]) {
```

```
    int i;
```

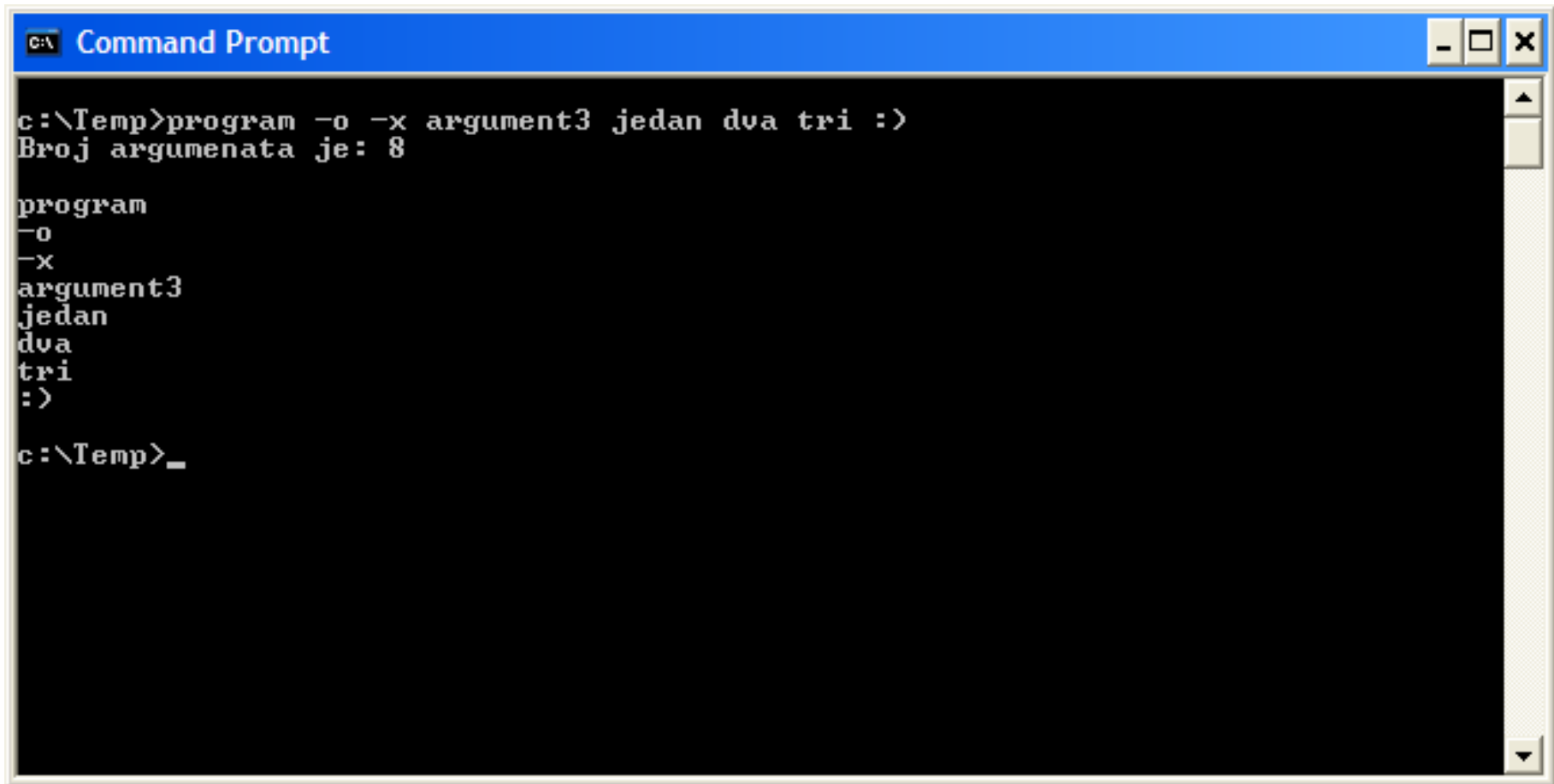
```
    printf("Broj argumenata je: %d\n\n", brArg);
```

```
    for(i=0; i<brArg; i++) {  
        puts(arg[i]);  
    }
```

```
    return 0;
```

```
}
```

Primer poziva programa sa argumentima komandne linije



```
C:\> Command Prompt

c:\Temp>program -o -x argument3 jedan dva tri :>
Broj argumenata je: 8

program
-o
-x
argument3
jedan
dva
tri
:>

c:\Temp>_
```

Primer rada sa tekstualnom datotekom - upis

```
#include <stdio.h>
#include <stdlib.h>
typedef char Tizraz30[31];           // 30 karaktera i /0
int main()
{
    FILE *IZLdat;
    Tizraz30 ime, prezime, NazivIZLdat;
    printf("\nUnesite Vase ime: ");
    scanf("%s",ime);
    __fpurge(stdin);
    printf("\nUnesite vase prezime: ");
    scanf("%s",prezime);
    __fpurge(stdin);
    printf("\nUnesite naziv datoteke u koju zelite da zapisete podatke: ");
    scanf("%s",NazivIZLdat);

    if((IZLdat=fopen(NazivIZLdat,"w"))==NULL)           // otvaranje datoteke sa proverom
                                                         //prava na pisanje(w)
    {
        printf("\nGreska prilikom otvaranja datoteke %s za pisanje",NazivIZLdat);
        exit(EXIT_FAILURE);           // prevremeni izlaz iz programa u slucaju zabrane
                                     //pisanja u datoteku
    }
    fprintf(IZLdat,"%s\n%s",ime, prezime);           // upis u izlaznu datoteku
    fclose(IZLdat);                                 // zatvaranje datoteke
    printf("\n Podaci studenta su upisani u datoteci: %s", NazivIZLdat);
    printf("\n\n Za izlazak iz programa pritisnite bilo koji taster!");//;

    return 0;
}
```

Primer rada sa tekstualnom datotekom - čitanje

```
#include<stdio.h>
#define MAXL 80

int main() {
    FILE *pf;
    char str[MAXL];

    pf=fopen("test.txt","r");

    if(pf!=NULL){
        while(fgets(str, MAXL, pf)!=NULL)
            puts(str);

        fclose(pf);
    } else {
        printf("Nije moguće otvoriti datoteku ili datoteka ne postoji.");
    }

    return 0;
}
```

Zadatak 4

- Prebrojati reči iz ulazne datoteke. U izlaznoj datoteci ispisati broj reči i najdužu pronađenu reč. Koristiti funkcije, imena datoteka prihvatiti kroz argumente komandne linije.

Zadatak 5

Iz ulazne datoteke učitati niz studenata koji su položili ispit (indeks, ime, ocena). Koristiti funkcije, ime ulazne datoteke prihvatiti kroz argumente komandne linije.

- U izlaznu datoteku "najvisi.txt" ispisati podatke o studentu sa najvišim prosekom
- U izlaznu datoteku "najnizi.txt" ispisati podatke o studentu sa najnižim prosekom
- Na standardnom izlazu ispisati prosečnu ocenu svih studenata

Zadatak 6

- Napisati funkciju `void kodiraj(char *ul_dat, char *izl_dat)` koja kodira ulaznu datoteku tako što svako slovo iz ulazne datoteke pomera za jednu poziciju unapred, u alfabetu. Parametri `ul_dat` i `izl_dat` su nazivi ulazne, odn. izlazne datoteke.
- Napisati funkciju `void dekodiraj(char *ul_dat, char *izl_dat)` koja dekodira ulaznu datoteku tako što svako slovo iz ulazne datoteke pomera za jednu poziciju unazad, u alfabetu. Parametri `ul_dat` i `izl_dat` su nazivi ulazne, odn. izlazne datoteke.
- Napisati program koji testira prethodno definisane funkcije