



Osnovi programiranja

Vežbe 12

ŠKOLSKA 2025/2026 GODINA
PRIRODNO-MATEMATIČKI FAKULTET
INSTITUT ZA MATEMATIKU I INFORMATIKU

Polja struktura se formiraju na isti način kao i polja podataka elementarnih tipova.

```
struct tacka
{
    int x;
    int y;
};
// deklaracija niza struktura
struct tacka tacke[50];
// pristup članu strukture
tacke[10].x = 5;
// pristup članu strukture
tacke[10].y = 15;
```

Često se u sprezi sa definisanjem strukture koristi i naredba za definisanje novog tipa.

typedef <poznati_tip> <novo_ime>

```
struct tacka
{
    int x;
    int y;
};
// definisanje novog tipa
typedef struct tacka Point;
// deklaracija promenljivih
Point pt1, pt2;
// pristup članu strukture
pt1.x = 15;
```

Napisati program na jeziku C koji koristi stukturu Osoba(ime, prezime, godinaRodj);

a) Ucitati i prikazati navedene podatke za vas kao primer osobe.

b) zatim u programu predvidi opis jos jedne osobe, ucitavanje podataka za tu osobu, a zatim prikaz tih podataka

c) uporedi po starosti osobe i prikazi ime i prezime starije osobe

```
#include <stdio.h>
typedef struct
{
    char ime[21];
    char prezime[21];
    int godRodj;
} Osoba;
main()
{
    Osoba os1, os2;
    printf("\nUnesi podatke za prvu osobu");
    printf("\nIme:");
    scanf("%s", os1.ime);
    printf("\nPrezime:");
    scanf("%s", os1.prezime);
    printf("\nGodina rodjenja:");
    scanf("%d", &os1.godRodj);
    printf("\nPrikaz podataka za prvu osobu:");
    printf("\nIme:");
    printf("%s", os1.ime);
    printf("\nPrezime:");
    printf("%s", os1.prezime);
```

```
printf("\nGodina rodjenja:");
printf("%d", os1.godRodj);
printf("\nUnesi podatke za drugu osobu");
printf("\nIme:");
scanf("%s", os2.ime);
printf("\nPrezime:");
scanf("%s", os2.prezime);
printf("\nGodina rodjenja:");
scanf("%d", &os2.godRodj);
printf("\nPrikaz podataka za drugu osobu:");
printf("\nIme:");
printf("%s", os2.ime);
printf("\nPrezime:");
printf("%s", os2.prezime);
printf("\nGodina rodjenja:");
printf("%d\n", os2.godRodj);
if (os1.godRodj < os2.godRodj)
    printf("%s %s\n", os1.ime, os1.prezime);
else
    printf("%s %s\n", os2.ime, os2.prezime);}
```

Učitati dva kompleksna broja (realne i imaginarne delove) i izračunati i ispisati njihov zbir.

```
#include <stdio.h>
struct kbroj
{
    float realno;
    float imaginarno;
};
main()
{
    struct kbroj a, b, c;
    printf("\n Ucitaj 1. broj - realni i imaginarni dio:");
    scanf("%f %f", &a.realno, &a.imaginarno);
    printf("\n Ucitaj 2, broj - realni i imaginarni dio:");
    scanf("%f %f", &b.realno, &b.imaginarno);
    c.realno = a.realno + b.realno;
    c.imaginarno = a.imaginarno + b.imaginarno;
    printf("\n Zbroj je: %.2f + %.2fi", c.realno, c.imaginarno);
}
```

Napisati program koji izračunava obim i površinu trougla i kvadrata u ravni.

- a) Podatke i koordinatama tačke čuvati u okviru strukture point. Koordinatama mogu biti celi brojevi.
- b) Napisati funkciju koja za dve date tačke računa rastojanje između njih
- c) Napisati funkciju koja za tri date tačke računa površinu trougla koji one obrazuju
- d) Napisati funkciju koja za dati niz tačaka i broj tačaka u nizu računa obim poligona koji te tačke obrazuju.
- e) Napisati funkciju koja za dati niz tačaka i broj tačaka u nizu računa površinu poligona koji te tačke obrazuju.

```
#include <stdio.h>
#include <math.h>
struct point
{
    int x;
    int y;
};
```

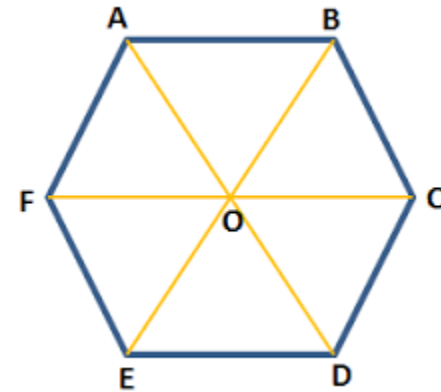
```
float segment_length(struct point A, struct point B)
{
    int dx = A.x - B.x;
    int dy = A.y - B.y;
    return sqrt(dx * dx + dy * dy);
}
```

c) Napisati funkciju koja za tri date tačke računa površinu trougla koji one obrazuju

d) Napisati funkciju koja za dati niz tačaka i broj tačaka u nizu računa obim poligona koji te tačke obrazuju.

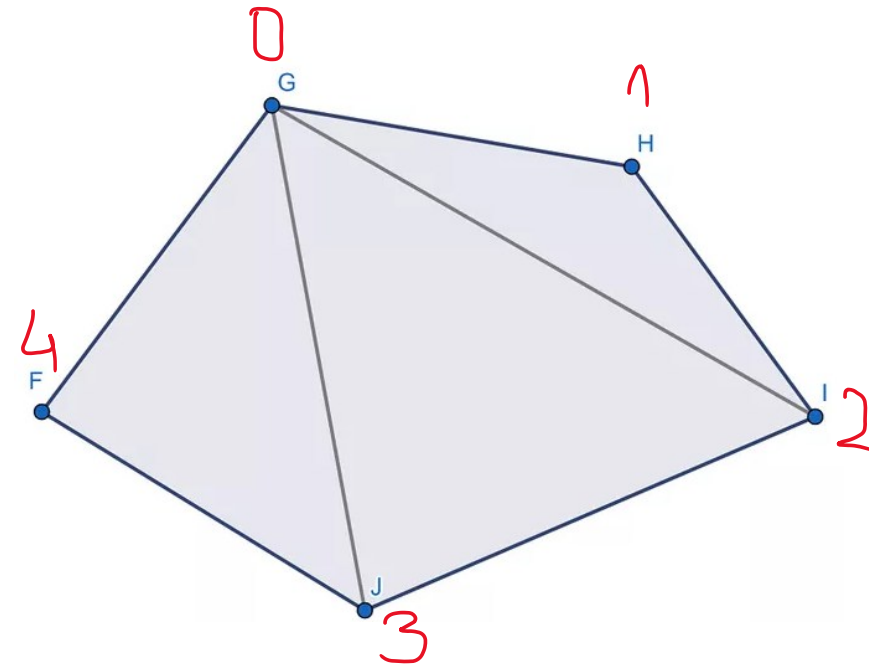
```
float Heron(struct point A, struct point B, struct point C)
{
    float a = segment_length(B, C), b = segment_length(A, C), c = segment_length(A, B);
    float s = (a + b + c) / 2;
    return sqrt(s * (s - a) * (s - b) * (s - c));
}
```

```
float circumference(struct point polygon[], int num)
{
    int i;
    float o = 0.0;
    for (i = 0; i < num - 1; i++)
        o += segment_length(polygon[i], polygon[i + 1]);
    o += segment_length(polygon[num - 1], polygon[0]);
    return o;
}
```



e) Napisati funkciju koja za dati niz tačaka i broj tačaka u nizu računa površinu poligona koji te tačke obrazuju.

```
float area(struct point polygon[], int num)
{
    float a = 0.0;
    int i;
    for (i = 1; i < num - 1; i++)
        a += Heron(polygon[0], polygon[i], polygon[i + 1]);
    return a;
}
```



Napisati program koji izračunava obim i površinu trougla i kvadrata u ravni.

```
main()
{
    struct point a, b = {1, 2}, triangle[3];
    struct point square[4] = {{0, 0}, {0, 1}, {1, 1}, {1, 0}};
    a.x = 0;
    a.y = 0;
    triangle[0].x = 0;
    triangle[0].y = 0;
    triangle[1].x = 0;
    triangle[1].y = 1;
    triangle[2].x = 1;
    triangle[2].y = 0;
    printf("sizeof(struct point) = %ld\n", sizeof(struct point));
    printf("x koordinata tacke a je %d\n", a.x);
    printf("y koordinata tacke a je %d\n", a.y);
    printf("x koordinata tacke b je %d\n", b.x);
    printf("y koordinata tacke b je %d\n", b.y);
    printf("Obim trougla je %f\n", circumference(triangle, 3));
    printf("Obim kvadrata je %f\n", circumference(square, 4));
    printf("Pov. trougla: %f\n", Heron(triangle[0], triangle[1], triangle[2]));
}
```

Napisati program u kome se najpre unosi ime tekstualne datoteke sa podacima o **aranžmanima**.
Definisati strukturu aranzman koja sadrži sledeće podatke:

- ime agencije (može imati više reči),
- ime hotela (jedna reč),
- mesto (može imati više reči),
- broj noćenja (ceo broj),
- cena aradžmana (realan broj).

```
#include <stdio.h>
#include <string.h>
typedef struct aranzman{
    char ime_agencije[30];
    char ime_hotela[30];
    char mesto[30];
    int broj_nocenja;
    float cena;
}Aranzman;
```

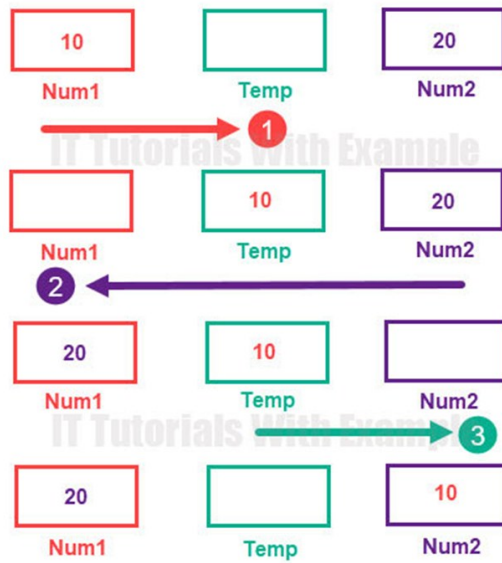
U navedenoj datoteci se u prvoj liniji nalazi broj aradžmana, a zatim se u svakom redu nalazi po jedan podatak o aranzmanu.

Program sadrži funkciju Sort kojom se sortiraju ponude po ceni aradžmana.

U glavnom programu uneti broj k i ispisati podatke o k najjeftinijih aradžmana.

aranzmani.txt:
16
Rapsody Travel
Mega***
Bec
4
1200
Sunny Holidays
Seaside
Herceg Novi
6
68000.75
.....

```
int Unos(Aranzman niz[])
{
    int n, i;
    char linija[255];
    FILE *fajl=fopen("aranzmani.txt","r");
    fscanf(fajl,"%d",&n);
    for(i=0;i<n;i++)
    {
        fgetc(fajl);
        fgets(linija,255,fajl);
        if(linija[strlen(linija)-1]=='\n') linija[strlen(linija)-1]='\0';
        strcpy(niz[i].ime_agencije,linija);
        fscanf(fajl,"%s",niz[i].ime_hotela);
        fgetc(fajl);
        fgets(linija,255,fajl);
        if(linija[strlen(linija)-1]=='\n') linija[strlen(linija)-1]='\0';
        strcpy(niz[i].mesto,linija);
        fscanf(fajl,"%d",&niz[i].broj_nocenja);
        fscanf(fajl,"%f",&niz[i].cena);
    }
    fclose(fajl);
    return n;
}
```



[3, 5, 8, 1, 4]
 [1, 5, 8, 3, 4]
 [1, 3, 8, 5, 4]
 [1, 3, 5, 8, 4]
 [1, 3, 4, 8, 5]
 [1, 3, 4, 5, 8]

```
void Sort(Aranzman niz[], int n)
{
    int i,j;
    Aranzman pom;
    for(i=0;i<n-1;i++)
    {
        for(j=i+1; j<n; j++)
        {
            if(niz[i].cena>niz[j].cena )
            {
                pom=niz[i];
                niz[i]=niz[j];
                niz[j]=pom;
            }
        }
    }
}
```

```

void Ispis(Aranzman niz[], int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%s\n",niz[i].ime_agencije);
        printf("%s\n", niz[i].ime_hotela);
        printf("%s\n",niz[i].mesto);
        printf("%d\n",niz[i].broj_nocenja);
        printf("%.2f\n",niz[i].cena);
        printf("-----\n");
    }
}

```

```

int main()
{
    int n;
    Aranzman niz[30];
    n=Unos(niz);
    //Ispis(niz,n);
    int k;
    scanf("%d" , &k );
    Sort(niz , n);
    Ispis(niz , k);

    return 0;
}

```

? DOMAĆI

1. Napisati program u kome se najpre unose ime tekstualne datoteke sa podacima o super brojevima i ime izlazne datoteke.

Definisati strukturu `super_broj` koja sadrži sledeće podatke:

- ime broja,
- vrednost

Napisati sledeće funkcije:

- a) Funkciju `unosBrojeva` koja iz datoteke datog naziva čita podatke o super brojevima sve dok ne dodje do kraja datoteke i podatke smešta u niz super brojeva i vraća ceo broj koji predstavlja broj učitanih super brojeva.
- b) Funkciju `ispisBrojeva` koji ispisuje podatke o super brojevima koji se nalaze u nizu date dužine.
- c) Funkciju `formMatrica` koja na osnovi niza date dužine formira kvadratnu matricu dimenzije kao i niz i čiji su elementi prve vrste jednaki elementima niza, a i-ta vrsta se dobija cikličkim pomeranjem unetog niza za i-1 mesto u levo.
- d) Funkciju `ispisMatrice` koja kvadratnu matricu date dimenzije ispisuje u matričnom obliku u datoteku datog naziva.

U glavnom delu programa učitati nazive ulazne i izlazne datoteke, zatim formirati niz od super brojeva koji se nalaze u datoteci, ispisati ga na standardni izlaz i od njega formirati matricu koristeći funkciju `formMatrica` i ispsati je u datoteku čiji je naziv `unet`.

```
#include <stdio.h>
struct super_broj
{
    char ime_broja[20];
    int vrednost;
};
int unosBrojeva(char ime_dat[20], struct super_broj niz[])
{
    FILE *f = fopen(ime_dat, "r");
    int i = 0;
    while (!feof(f))
    {
        fscanf(f, "%s", niz[i].ime_broja);
        fscanf(f, "%d", &niz[i].vrednost);
        i++;
    }
    fclose(f);
    return i;
}
```

```

void ispisBrojeva(struct super_broj niz[], int n)
{
    int i;
    for (i = 0; i < n; i++)
    {
        printf("%s\n", niz[i].ime_broja);
        printf("%d\n", niz[i].vrednost);
    }
}

void formMatrica(struct super_broj niz[], int n, struct super_broj matrica[20][20])
{
    int i, j;
    struct super_broj k;
    for (j = 0; j < n; j++)
        matrica[0][j] = niz[j];
    for (i = 1; i < n; i++)
    {
        k = niz[0];
        for (j = 0; j < n - 1; j++)
            niz[j] = niz[j + 1];
        niz[n - 1] = k;
        for (j = 0; j < n; j++)
            matrica[i][j] = niz[j];
    }
}

```

```

void ispisMatrice(char ime_dat[20], struct super_broj matrica[20][20], int n)
{
    int i, j;
    FILE *f = fopen(ime_dat, "w");
    for (i = 0; i < n; i++)
    {
        for (j = 0; j < n; j++)
            fprintf(f, " %s %d", matrica[i][j].ime_broja, matrica[i][j].vrednost);
        fprintf(f, "\n");
    }
    fclose(f);}

main(){
    struct super_broj niz[20], matrica[20][20], pom;
    int n, i, j, k;
    char ime_dat[20], ime_izlaz[20];
    printf("Ime ulazne datoteke\n");
    scanf("%s", ime_dat);
    printf("Ime izlazne datoteke\n");
    scanf("%s", ime_izlaz);
    n = unosBrojeva(ime_dat, niz);
    ispisBrojeva(niz, n);
    formMatrica(niz, n, matrica);
    ispisMatrice(ime_izlaz, matrica, n);
}

```

Hvala na pažnji!