

Na **Desktop**-u u direktorijumu **Rad** kreirati direktorijum **ImePrezime_BrIndeksi** unutar njega sačuvati programe koji sadrže rešenja datih zadataka. Rešenje 1. zadatka mora da se nalazi u fajlu **Zadatak1.c**, rešenje 2. zadatka mora da se nalazi u fajlu **Zadatak2.c**, rešenje 3. zadatka mora da se nalazi u fajlu **Zadatak3.c**. Od tri ponuđena zadatka birate dva koja ćete raditi.

Napisati C program koji:

1. kao parametar pri startovanju program dobija imena ulazne i izlazne datoteke. U ulaznoj datoteci se nalaze samo pozitivni celi brojevi koji predstavljaju grupe signala po sledećem pravilu brojIndikator (koji predstavlja koliko brojeva se nalazi u toj signal grupi) pa za njim brojevi te signal grupe i tako redom do kraja datoteke. U izlaznu datoteku prepisati samo signalne brojeve bez indikatora.

Primer: Ulaz.txt: 4 5 8 7 2 3 4 5 6 1 2 3 5 4 3 2 7 8

Izlaz.txt: 5 8 7 2 4 5 6 2 5 4 3 7 8

```
#include<stdio.h>
#include<string.h>
main(int argc,char* argv[]){
    int c,k,p,i;

    FILE *in=fopen(argv[1],"r");
    if(in==NULL){
        fprintf(stderr,"Datoteka %s se ne moze otvoriti!!\n",argv[1]);
        return 1;
    }
    FILE *out=fopen(argv[2],"w");
    if(out==NULL){
        fprintf(stderr,"Datoteka %s se ne moze otvoriti!!\n",argv[2]);
        return 1;
    }

    while((fscanf(in,"%d",&c))!=EOF){
        for(i=0;i<c;i++){
            fscanf(in,"%d",&k);
            fprintf(out,"%d",k);
        }

        fprintf(out,"\n");
        fclose(in);
        fclose(out);
    }
}
```

2. kao parametar pri startovanju programa dobija ime tekstualne datoteke sa podacima o igraču (ime igrača, naziv proizvod, i ponuđena cena(realan broj)) i to po jedan podatak u jednom redu. Definirati strukturu **igrac** koja sadrži navedene podatke. Organizovana je javna prodaja proizvoda, tako da će neki proizvod biti prodat ako je minimalna cena koja je ponuđena jedinstvena, a ukoliko je više igrača ponudilo istu minimalnu cenu, proizvod se ne prodaje. Program treba da sadrži funkciju **Proizvod** koja za zadati naziv proizvoda određuje cenu po kojoj je prodat taj proizvod, ukoliko proizvod nije prodat vraća -1. Organizatori javne prodaje žele da naprave niz prodatih proizvoda, gde će za svaki proizvod u okviru strukture **proizvod** čuvati naziv proizvoda i cenu po kojoj je prodat. U glavnom delu programa učitati podatke o igračima iz datoteke, i formirati i ispisati traženi niz proizvoda.

Bonus (4 poena): Umesto navedenog načina javna prodaja je organizovana tako da će proizvod biti prodat po minimalnoj jedinstvenoj ceni. Program treba da sadrži funkciju **Proizvod** koja za zadati naziv proizvoda određuje cenu po kojoj je prodat taj proizvod ukoliko se zna da je prodat po minimalnoj jedinstvenoj ceni.

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

struct igrac
{
    char ime[20];
    char nazivpr[20];
}
```

```

        float cena;
    };

    struct proizvod
    {
        char naziv[20];
        float cena;
    };

    float Proizvod(struct igrac a[],int n, char ime[]);

    main(int argc,char* argv[])
    {
        FILE* f=fopen(argv[1],"r");
        int i,n=0,j,m=0,ind=0;
        struct igrac a[100];
        struct proizvod b[100];
        char s[20];
        while(1)
        {
            fscanf(f,"%s",s);
            if (feof(f)) break;
            strcpy(a[n].ime,s);
            fscanf(f,"%s",a[n].nazivpr);
            fscanf(f,"%f",&a[n++].cena);
        }
        fclose(f);

        for (i=0;i<n;i++)
        {
            for (j=0;j<m;j++)
                if (strcmp(a[i].nazivpr,b[j].naziv)==0) ind=1;
            if (ind==0) strcpy(b[m++].naziv,a[i].nazivpr);
            ind=0;
        }
        for (i=0;i<m;i++) b[i].cena=Proizvod(a,n,b[i].naziv);
        for (i=0;i<m;i++) if (b[i].cena!=-1) printf("%s %.2f\n",b[i].naziv,b[i].cena);
    }

    float Proizvod(struct igrac a[],int n,char ime[])
    {
        float min;
        int i,br=0;

        for (i=0;i<n;i++) if (strcmp(ime,a[i].nazivpr)==0)
        {
            min=a[i].cena;
            break;
        }
        for (i=0;i<n;i++) if ((min>a[i].cena)&&strcmp(ime,a[i].nazivpr)==0) min=a[i].cena;
        for (i=0;i<n;i++) if ((min==a[i].cena)&&strcmp(ime,a[i].nazivpr)==0) br++;
        if (br>1) return (-1);
        else return(min);
    }

```

bonus:

```

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

struct igrac
{
    char ime[20];
    char nazivpr[20];
    float cena;

```

```

    };

    struct proizvod
    {
        char naziv[20];
        float cena;
    };

    float Proizvod(struct igrac a[],int n, char ime[]);

    main(int argc,char* argv[])
    {
        FILE* f=fopen(argv[1],"r");
        int i,n=0,j,m=0,ind=0;
        struct igrac a[100];
        struct proizvod b[100];
        char s[20];
        while(1)
        {
            fscanf(f,"%s",s);
            if (feof(f)) break;
            strcpy(a[n].ime,s);
            fscanf(f,"%s",a[n].nazivpr);
            fscanf(f,"%f",&a[n++].cena);
        }
        fclose(f);

        for (i=0;i<n;i++)
        {
            for (j=0;j<m;j++)
                if (strcmp(a[i].nazivpr,b[j].naziv)==0) ind=1;
            if (ind==0) strcpy(b[m++].naziv,a[i].nazivpr);
            ind=0;
        }
        for (i=0;i<m;i++) b[i].cena=Proizvod(a,n,b[i].naziv);
        for (i=0;i<m;i++) if (b[i].cena!=-1) printf("%s %.2f\n",b[i].naziv,b[i].cena);
    }

    float Proizvod(struct igrac a[],int n,char ime[])
    {
        float b[100],t;
        int i,j,m=0;
        for (i=0;i<n;i++) if (strcmp(ime,a[i].nazivpr)==0) b[m++]=a[i].cena;
        for (i=0;i<m-1;i++)
            for (j=i+1;j<m;j++)
                if (b[i]>b[j])
                {
                    t=b[i];
                    b[i]=b[j];
                    b[j]=t;
                }
        if (m==1) return b[0];
        else if ((m==2)&&(b[0]!=b[1])) return b[0];
        else if (m>2)
        {
            if (b[0]!=b[1]) return b[0];
            else for (i=1;i<m-1;i++) if ((b[i]!=b[i-1])&&(b[i]!=b[i+1])) return b[i];
            if (b[m-1]!=b[m-2]) return b[m-1];
        }
        return -1;
    }

```

3. učitava niz od n karaktera. Napisati rekurzivnu funkciju **asciiSimetricanZbir** koja u svakom redu štampa zbir ascii vrednosti prvog i poslednjeg karaktera, zatim drugog i pretposlednjeg itd, ukoliko je niz neparne dužine štampa se ascii vrednost karaktera u sredini niza. U glavnom delu programa korišćenjem funkcije ispisati ascii simetrične zbirove.

Primer: Ulaz: N = 7 niz: Ana uci

Izlaz: 170

209

214

32

```
#include<stdio.h>
#include<stdlib.h>
int asciiSimetricanZbir(char *a,int n,int j)
{
    if (j>n) return 1;
    if (j==n){ printf("%d",a[j]); return 2;}
    printf("%d ",a[j]+a[n]);
    asciiSimetricanZbir(a,n-1,j+1);
}
main()
{
    int n,i;
    printf("unesite n\n");
    scanf("%d",&n);
    char *a=(char*)malloc(n*sizeof(char));
    printf("unesite clanove niza:\n");
    for (i=0;i<n;i++)
        a[i]=getchar();
    asciiSimetricanZbir(a,n-1,0);
}
```

Broj poena: 1. zadatak – 7 poena, 2. zadatak – 14 poena, 3. zadatak – 11 poena

Vreme izrade: 60 minuta