

Strukture podataka i algoritmi 1

(zadatak - max 30 poena)

Jul, 2017

Distributer filmova ima ugovore sa više bioskopa. Svaki od filmova ovog distributera u svim bioskopima u kojima se prikazuje ima isti broj projekcija i u isto vreme. Napisati program koji iz datoteke učitava broj filmova koje distributer nudi, zatim za svaki film se učitava

njegov ID (ceo broj), ime (niz karaktera), žanr (niz karaktera), broj projekcija (ceo broj) i za svaku projekciju sat i minut prikazivanja (dva cela broja) i

formira niz svih filmova. Potom se iz datoteke učitava broj bioskopa sa kojima ovaj distributer ima ugovor i za svaki bioskop se učitava

ID bioskopa (ceo broj), ime (niz karaktera), grad (niz karaktera), broj filmova tog distributera koji se trenutno prikazuju (ceo broj) i ID svakog filma koji se prikazuje i

formira niz bioskopa. Nakon učitavanja, u datoteku *Repertoar.txt* ispisati repertor svih bioskopa. Sa standardnog ulaza učitati grad, žanr filma i željeni vremenski interval za početak prikazivanja filma i na izlazu ispisati sve filmove koji zadovoljavaju uneti kriterijum i koliko takvih filmova ima. Distributer je u nekom trenutku povećao ponudu filmova, učitati iz datoteke broj novih filmova i podatke o njima i dodati ih u niz svih filmova. Omogućiti promenu repertoara bioskopa, tako što se sa standardnog ulaza učitava ID bioskopa koji menja repertoar, zatim novi broj filmova i njihovi ID-jevi.

(4 poena)

Za rešavanje problema napisati sledeće funkcije:

a) Napisati funkciju **UcitajFilmove** koja iz u datoteke učitava i formira niz filmova.

(3+1 poena)

b) Napisati funkciju **NadjiFilm** koja za zadati ID filma vraća pokazivač na odgovarajući element u nizu filmova.

(2 poena)

c) Napisati funkciju **UcitajBioskope** koja iz datoteke učitava podatke o bioskopima i ID-jeve filmova koji se prikazuju formirajući niz svih bioskopa, pri čemu svaki bioskop "pokazuje" na filmove koji se u njemu prikazuju.

(4+2 poena)

d) Napisati funkciju **Stampa** koja u datoteku štampa repertoar svih bioskopa.

(2 poena)

e) Napisati funkciju **OdlazakUBioskop** koja za dati grad, žanr filma i željeni vremenski interval za početak prikazivanja filma, na izlazu ispisuje sve filmove koji zadovoljavaju uneti kriterijum i vraća koliko takvih filmova ima.

(4 poena)

f) Napisati funkciju **NoviFilmovi** koja će sa standardnog ulaza učitati broj novih filmova, podatke o filmovima i dodati ih u niz filmova.

(3 poena)

g) Napisati funkciju **PromenaRepertoara** koja za bioskop sa unetim ID-jem, stari repertoar zameniti novim.

(5 poena)

Zadatak rešiti bez korišćenja globalnih promenljivih i bez unapred definisanih dužina korišćenih nizova.

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

```
typedef struct {
    int h,min;
}vreme;
```

```
typedef struct {
    int ID;
    char *ime;
    char *zanr;
    int br_projekcija;
    vreme *projekcija;
}film;
```

```
typedef struct {
    int ID;
    char *ime;
    char *grad;
    int br_filmova;
    film **f;
}bioskop;
```

```
void UcitajFilmove(FILE *f, film * fl, int n);
void UcitajBioskope(FILE *f, bioskop *b, int m, film * fl, int n);
film* NadjiFilm(int id,film *fl,int n);
void Stampa(bioskop *b,int m);
void OdlazakUBioskop(bioskop *b, int m);
void NoviFilmovi(film **fl,int *n);
void PromenaRepetoara(bioskop *b, int m, film *fl,int n);
```

```

int main(){
    FILE *f;
    f=fopen("ulaz.txt","r");
    if (f==NULL){
        printf("Ne moze se otvoriti fajl");
        return 1;
    }
    int n;
    fscanf(f,"%d",&n);
    film *filmovi;
    filmovi=(film *) malloc(n*sizeof(film));
    if (filmovi==NULL){
        printf("Ne moze se alocirati niz");
        return 1;
    }
    UcitajFilmove(f,filmovi,n);
    int m;
    fscanf(f,"%d",&m);
    bioskop *bioskopi;
    bioskopi=(bioskop *) malloc(n*sizeof(bioskop));
    if (f==NULL){
        printf("Ne moze se alocirati niz ");
        return 1;
    }
    UcitajBioskope(f,bioskopi,m,filmovi,n);
    Stampa(bioskopi,m);
    fclose(f);
    OdlazakUBioskop(bioskopi,m);
    NoviFilmovi(&filmovi,&n);
    PromenaRepetoara(bioskopi,m,filmovi,n);
}

```

```

void UcitajFilmove(FILE *f, film * fl, int n){
    int i,j,k;
    char rec[20];
    for(i=0;i<n;i++){
        fscanf(f,"%d",&(fl[i].ID));
        fgetc(f);
        fgets(rec,20,f);
        k=strlen(rec)+1;
        fl[i].ime=(char *) malloc(k*sizeof(char));
        strcpy(fl[i].ime,rec);
        fl[i].ime[k-2]='\0';
        fgets(rec,20,f);
        k=strlen(rec);
        fl[i].zanr=(char *) malloc(k*sizeof(char));
        strcpy(fl[i].zanr,rec);
        fl[i].zanr[k-1]='\0';
        fscanf(f,"%d",&(fl[i].br_projekcija));
        k=fl[i].br_projekcija;
        fl[i].projekcija=(vreme *) malloc(k*sizeof(vreme));
        for(j=0;j<k;j++){
            fscanf(f,"%d%d",&(fl[i].projekcija[j].h),
                    &(fl[i].projekcija[j].min));
        }
    }
}

```

```

film* NadjFilm(int id,film *fl,int n){
    int i;
    for(i=0;i<n;i++)
        if (fl[i].ID==id) break;
    if (i==n) return NULL;
        else return &(amp;fl[i]);
}

```

```

void UcitajBioskope(FILE *f, bioskop *b, int m, film * fl, int n){
    int i,j,k,id;
    char rec[20];
    for(i=0;i<m;i++){
        fscanf(f,"%d",&(b[i].ID));
        fgetc(f);
        fgets(rec,20,f);
        k=strlen(rec);
        b[i].ime=(char *) malloc(k*sizeof(char));
        strcpy(b[i].ime,rec);
        b[i].ime[k-1]='\0';
        fgets(rec,20,f);
        k=strlen(rec);
        b[i].grad=(char *) malloc(k*sizeof(char));
        strcpy(b[i].grad,rec);
        b[i].grad[k-1]='\0';
        fscanf(f,"%d",&(b[i].br_filмова));
        k=b[i].br_filмова;
        b[i].f=(film **) malloc(k*sizeof(film *));
        for(j=0;j<k;j++){
            fscanf(f,"%d",&id);
            b[i].f[j]=NadjFilm(id,fl,n);
            if (b[i].f[j]==NULL){
                j--;
                k--;
            }
        }
        b[i].br_filмова=k;
    }
}

```

```

void Stampa(bioskop *b,int m){
    FILE *f;
    f=fopen("Repetoar.txt","w");
    int i,j,k;
    for(i=0;i<m;i++){
        fprintf(f,"%s\t%s\n",b[i].ime,b[i].grad);
        for(j=0;j<b[i].br_filmova;j++){
            fprintf(f,"%s\t%s\n",b[i].f[j]->ime,b[i].f[j]->zanr);
            for(k=0;k<b[i].f[j]->br_projekcija;k++)
                fprintf(f,"%2.0d : %2.0d\t",b[i].f[j]->projekcija[k].h,
                    b[i].f[j]->projekcija[k].min);

            fputc('\n',f);
        }
        fputc('\n',f);
    }
    fclose(f);
}

```

```

void OdlazakUBioskop(bioskop *b, int m){
    char grad[20], zanr[20];
    printf("Unesite grad: ");
    fgets(grad,20,stdin);
    grad[strlen(grad)-1]='\0';
    printf("Unesite zanr: ");
    fgets(zanr,20,stdin);
    zanr[strlen(zanr)-1]='\0';
    int h1,min1,h2,min2;
    printf("Unesite vremenski period: ");
    scanf("%d%d%d%d",&h1,&min1,&h2,&min2);
    int t1,t2;
    t1=60*h1+min1;
    t2=60*h2+min2;
    int i,j,k=0;
    film **f;
    f=(film **) malloc(sizeof(film *));
    for(i=0;i<m;i++){
        if (strcmp(b[i].grad,grad)==0) {
            f=(film **) realloc(f,b[i].br_filmova*sizeof(film *));
            for(j=k;j<(k+b[i].br_filmova);j++){
                f[j]=b[i].f[j-k];
            }
            k=k+b[i].br_filmova;
        }
    }
    if (k==0){
        printf("Ne postoji bioskop\n");
        return;
    }
}

```

```

for(i=0;i<k-1;i++)
    for(j=i+1;j<k;j++)
        if(f[i]->ID==f[j]->ID){
            f[j]=f[k-1];
            k--;
            j--;
        }

for(i=0;i<k;i++)
    if(strcmp(f[i]->zanr,zanr)!=0){
        f[i]=f[k-1];
        k--;
        i--;
    }
if (k==0){
    printf("Ne postoji trazeni zanr\n");
    return;
}
int t;
for(i=0;i<k;i++){
    for(j=0;j<f[i]->br_projekcija;j++){
        t=60*f[i]->projekcija[j].h+f[i]->projekcija[j].min;
        if((t>=t1)&&(t<=t2))
            break;
    }
    if (j==f[i]->br_projekcija){
        f[i]=f[k-1];
        k--;
        i--;
    }
    if (k==0){
        printf("Ne postoje filmovi u trazenom periodu\n");
        return;
    }
}
for(i=0;i<k;i++){
    printf("%s\n",f[i]->ime);
}
}

```

```

void NoviFilmovi(film **fl,int *n){
    int i,j,k,d;
    char rec[20];
    FILE *f=fopen("Novi filmovi.txt","r");
    fscanf(f,"%d",&d);
    (*fl)=(film *) realloc((*fl),d*sizeof(film));
    for(i=*n;i<*n+d;i++){
        fscanf(f,"%d",&((*fl)[i].ID));
        fgetc(f);
        fgets(rec,20,f);
        k=strlen(rec);
        (*fl)[i].ime=(char *) malloc(k*sizeof(char));
        strcpy((*fl)[i].ime,rec);
        (*fl)[i].ime[k-1]='\0';
        fgets(rec,20,f);
        k=strlen(rec);
        (*fl)[i].zanr=(char *) malloc(k*sizeof(char));
        strcpy((*fl)[i].zanr,rec);
        (*fl)[i].zanr[k-1]='\0';
        fscanf(f,"%d",&((*fl)[i].br_projekcija));
        k=(*fl)[i].br_projekcija;
        (*fl)[i].projekcija=(vreme *) malloc(k*sizeof(vreme));
        for(j=0;j<k;j++){
            fscanf(f,"%d%d",&((*fl)[i].projekcija[j].h),
                &((*fl)[i].projekcija[j].min));
        }
    }
    *n=*n+d;
    fclose(f);
}

```

```

void PromenaRepetoara(bioskop *b, int m, film *fl,int n){
    int i,j,k,id;
    printf("Unesite ID bioskopa: ");
    scanf("%d",&id);
    for(i=0;i<m;i++){
        if (b[i].ID==id) break;
    }
    if (i<m){
        printf("Unestie broj filmova: ");
        scanf("%d",&(b[i].br_filmova));
        free(b[i].f);
        k=b[i].br_filmova;
        b[i].f=(film **) malloc(k*sizeof(film *));
        for(j=0;j<k;j++){
            printf("Unesite ID filma: ");
            scanf("%d",&id);
            b[i].f[j]=NadjiFilm(id,fl,n);
            if (b[i].f[j]==NULL){
                j--;
                k--;
                printf("Ne postoji film");

            }
        }
        b[i].br_filmova=k;
    }
    else printf("Ne postoji bioskop");
}

```

Strukture podataka i algoritmi 1
Test – max 20 poena

Jul, 2017

1. Definirati strukturu za čuvanje podataka o bioskopima iz zadatka o bioskopima.

2. Zašto rezultat sledećeg koda nije NONE?

```
#include<stdio.h>
int main() {
    int a=10;
    switch(a) {
        case '1':
            printf("ONE\n");
            break;
        case '2':
            printf("TWO\n");
            break;
        default:
            printf("NONE\n");
    }
    return 0;
}
```

3. Ispraviti sledeći kod tako da štampa 20 puta znak '-'

```
#include <stdio.h>
int main()
{
    int i;
    int n = 20;
    for( i = 0; i < n; i-- )
        printf("-");
    return 0;
}
```

Bonus. Može li se kod u prethodnom zadatku ispraviti promenom samo jednog karaktera?

4. Ispraviti grešku u sledećem kodu

```
#include <stdio.h>
int main()
{
    int* ptr1,ptr2;
    ptr1 = malloc(sizeof(int));
    ptr2 = ptr1;
    *ptr2 = 10;
    return 0;
}
```

5. Rezultat koda

```
float f = 1.0;
int i1 = (int) f;
int i2 = * (int *) &f;
printf("%d\n, i1);
printf("%d\n, i2);
```

je

1

1065353216

Objasniti zašto.

6. Objasniti razliku između strukture i unije.
7. Neka je struktura definisana na sledeći način

```
struct {  
    char *s;  
    ...  
} nesto;
```

Napisati funkciju **NadjiElement** koja pronalazi element u nizu ovako definisanih struktura gde je član s jednak nekom zadatom stringu i vraća pokazivač na taj element.

8. Funkcija **obradi** treba da prihvati i vrati niz karaktera nepoznate dužine, koje od navedenih deklaracija su ispravne?

- a. `char* obradi(char*);`
- b. `char obradi(char);`
- c. `*char obradi(*char);`
- d. `char* obradi(char[]);`
- e. `char[] obradi(char*);`
- f. `*char obradi(char[]);`
- g. `char[30] obradi(char[]);`

9. Struktura je definisana na sledeći način

```
struct spa{  
    char *ime;  
    float broj_poenas;  
    int ocena;};
```

Definisati niz elemenata tipa date strukture i funkciju koja će sa ulaza učitati broj elemenata nizu (unapred nije poznato ograničenje za broj elemenata u nizu) i podatke za sve članove niza. Funkcija treba da bude tipa **void**.

10. Šta će biti rezultat rada sledećeg koda

```
#include <stdio.h>  
int main()  
{  
    int i = 6;  
    if( (++i < 7) && ( i++/6) || (++i <= 9))  
        ;  
    printf("%d\n",i);  
    return 0;  
}
```

11. Šta je opisano sledećim deklaracijama:

- a. `char (*f)[5];`
- b. `char* (*f)(char*);`
- c. `void (*f[10]) (int, int)`