

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadataka

Zadatak 1 (16 bodova) Napišite program koji kao prvi argument komandne linije prima nenegativan cijeli broj, a kao drugi argument nenegativan realni broj zapisan u notaciji pomične točke (bez eksponenta, kao kod formata %f). Program treba spremi učitane brojeve u odgovarajuće varijable tipa `int` i `double`, te ispisati njihove vrijednosti. Pretvorba za oba broja se vrši pozivom funkcije `pretvori` koja kao argumente prima string reprezentaciju učitano broj, a preko varijabilnog argumenta vraća `int` ili `double` reprezentaciju tog broja (ovisno o ulaznom broju). Napišite sve dodatne potrebne funkcije i definirajte formalne argumente funkcije `pretvori` (može ih biti i više od nabrojanih) da bi mogli ostvariti navedenu funkcionalnost. Obavezno napišite implementaciju funkcije `main`. Pripazite da izbjegnute upozorenja prevodioca uzrokovana konverzijama tipova argumenata funkcija.

Napomene: zabranjeno je korištenje funkcija `atof`, `atoi`, `atol`, `strtod`, `strtoul`, `strtoul`. Dopušteno je korištenje ostalih funkcija definiranih u datotekama zaglavlja `stdlib.h` i `ctype.h`.

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

void pretvori(char* broj, void *(*f)(void* ulaz), void **izlaz){

    *izlaz = f(broj);

    return;
}

int *pretvoriUInt(char *ulaz){

    int *rez = (int*) malloc (sizeof(int));

    *rez = 0;

    for(int i=0;ulaz[i]!='\0';i++){

        if(isdigit(ulaz[i])){
            *rez*=10;
            *rez+=(ulaz[i]-'0');
        }
        else return NULL;
    }
    return rez;
}

double *pretvoriUDouble(char *ulaz){

    double *rez = (double*) malloc (sizeof(double)), dec = 1;

    *rez = 0.0;

    int tocka = 0, pos = 0;

    for(int i=0;ulaz[i]!='\0';i++){
        if(isdigit(ulaz[i])){
            if(tocka == 0){
                *rez*=10;
                *rez+=(ulaz[i]-'0');
            }
            else if(tocka == 1){
                dec/=10.0;
                *rez+=dec*(ulaz[i]-'0');
            }
        }
    }
}
```

```
    else if(ulaz[i] == '.'){
        tocka = 1;
    }
    else return NULL;
}

return rez;
}

void* obradiInt(void *ulaz){
    return pretvoriUInt((char*) ulaz);
}

void* obradiDouble(void *ulaz){
    return pretvoriUDouble((char*) ulaz);
}

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

    int* broj1;
    double *broj2;

    pretvori(argv[1], obradiInt, (void**) &broj1);

    printf("Rezultat int: %d\n", *broj1);
    free(broj1);
    broj1 = NULL;

    pretvori(argv[2], obradiDouble, (void**) &broj2);

    printf("Rezultat double: %lg\n", *broj2);
    free(broj2);
    broj2 = NULL;
return 0;
}
```

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadataka

Zadatak 2 (2+12 bodova) Nakon održanog turnira društvenih igara, organizatori su odlučili napraviti rang ljestvicu igrača koji su sudjelovali na turniru. Svaka igra koja se igrala na turniru bila je igra za 4 igrača.

Za svakog igrača na turniru znamo njegovo ime koje ga jedinstveno određuje (string duljine najviše 30) i ukupan broj ostvarenih bodova na turniru (nenegativan cijeli broj). Podatke o igraču pamtimo u strukturi `player`.

Nadalje, za svaku igru koja se igrala na turniru znamo njezin naziv (string duljine najviše 50), težinu (prirodan broj između 1 i 10, uključivo) te niz `ekipa` od 4 igrača koji su odigrali partiju te igre (niz od 4 strukture `player`). Podatke o igri pamtimo u strukturi `game`. Poznato je da je niz `ekipa` **sortiran silazno** prema broju ostvarenih bodova u odigranoj partiji. Ako sa `tezina` označimo težinu igre, broj ostvarenih bodova u toj partiji igrača na indeksu `i` računamo prema formuli $tezina * (4 - i)$.

(a) Deklarirajte tip podatka `player` koji pamti navedene podatke o igraču. Dodatno, varijabla `x` tog tipa se treba moći definirati sa `player x`; . Deklarirajte tip podatka `game` koji pamti navedene podatke o igri. Dodatno, varijabla `y` tog tipa se treba moći definirati sa `game y`;

(b) Napišite funkciju

```
void rang_lista(game* odigrane, player* igraci, int m, int n)
```

koja kao argumente dobiva niz odigranih igara na turniru u nizu `odigrane`, niz svih igrača koji su sudjelovali na turniru u nizu `igraci` (s početnim bodovima postavljenim na 0) te prirodne brojeve `m` i `n` koji su redom duljine nizova `odigrane` i `igraci`. Funkcija treba sortirati niz `igraci` prema ukupnim bodovima igrača ostvarenim na turniru i na ekran ispisati rang listu igrača u formatu:

```
redni broj. ime igrača (broj bodova) \n.
```

Ako postoje igrači s istim brojem bodova, poredak među njima nije bitan, ali im se ispisuje isti redni broj.

Napomena. U nizu `odigrane` je moguće imati više od jedne strukture `game` s istim nazivom i težinom igre. Nije dozvoljeno korištenje dodatnih nizova.

Primjer. Pokažimo na primjeru kako ažuriramo bodove igrača u nizu `igraci`. Za niz igrača

```
igraci: P1={Petar, 0}, P2={Luka, 0}, P3={Iva, 0}, P4={Marija, 0}, P5={Lucija, 0}, P6={Ante, 0}
```

ako je niz `odigrane` jednak

```
{Splendor, 2, [P5,P2,P6,P1]}, {Catan, 7, [P3,P1,P5,P4]}, {Splendor, 2, [P3,P6,P2,P4]},
```

nakon poziva funkcije `rang_lista(odigrane,igraci,3,6)`, niz `igraci` treba biti (do na poredak igrača s istim bodovima):

```
igraci: P3={Iva, 36} P1={Petar, 23}, P5={Lucija, 15}, P2={Luka, 10}, P6={Ante, 10}, P4={Marija, 9},
```

a ispis:

1. Iva (36)
2. Petar (23)
3. Lucija (22)
4. Luka (10)
4. Ante (10)
6. Marija (9)

Rješenje:

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

```
typedef struct{
    char ime[31];
    int bodovi;
}player;
```

```
typedef struct{
    char naziv[51];
    int tezina;
    player ekipa[4];
}game;
```

```
void rang_lista(game* odigrane, player* igraci, int m, int n)
```

```

{
    int i,j,k,indeks,redni_broj=1;

    for(i=0;i<m;i++)
        for(j=0;j<4;j++)
            {
                indeks=0;
                while(strcmp(odigrane[i].ekipa[j].ime,igraci[indeks].ime)) indeks++;
                igraci[indeks].bodovi+=odigrane[i].tezina*(4-j);
            }

    for(i=0;i<n;i++)
        for(j=i+1;j<n;j++)
            if(igraci[i].bodovi<igraci[j].bodovi)
                {
                    player temp=igraci[i];
                    igraci[i]=igraci[j];
                    igraci[j]=temp;
                }

    printf("%d. %s (%d)\n", redni_broj,igraci[0].ime,igraci[0].bodovi);

    for(i=1;i<n;i++)
        {
            if(igraci[i-1].bodovi>igraci[i].bodovi) redni_broj=i+1;

            printf("%d. %s (%d)\n", redni_broj,igraci[i].ime,igraci[i].bodovi);
        }
}

//nije dio zadatka
int main()
{
    player igraci[6];
    game odigrane[3];

    int i;
    for(i=0;i<6;i++) igraci[i].bodovi=0;

    strcpy(igraci[0].ime,"Petar");
    strcpy(igraci[1].ime,"Luka");
    strcpy(igraci[2].ime,"Iva");
    strcpy(igraci[3].ime,"Marija");
    strcpy(igraci[4].ime,"Lucija");
    strcpy(igraci[5].ime,"Ante");

    strcpy(odigrane[0].naziv,"Splendor");
    odigrane[0].tezina=2;
    odigrane[0].ekipa[0]=igraci[4]; odigrane[0].ekipa[1]=igraci[1];
    odigrane[0].ekipa[2]=igraci[5]; odigrane[0].ekipa[3]=igraci[0];

    strcpy(odigrane[1].naziv,"Catan");
    odigrane[1].tezina=7;
    odigrane[1].ekipa[0]=igraci[2]; odigrane[1].ekipa[1]=igraci[0];
    odigrane[1].ekipa[2]=igraci[4]; odigrane[1].ekipa[3]=igraci[3];

    strcpy(odigrane[2].naziv,"Splendor");
    odigrane[2].tezina=2;
    odigrane[2].ekipa[0]=igraci[2]; odigrane[2].ekipa[1]=igraci[5];
    odigrane[2].ekipa[2]=igraci[1]; odigrane[2].ekipa[3]=igraci[3];

    rang_lista(odigrane,igraci,3,6);
    return 0;
}

```

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadataka

Zadatak 3 (2+11 bodova) Svi automobili dostupni za prodaju u nekoj firmi nalaze se u vezanoj listi. Za svaki automobil pamtimo naziv (string s najviše 20 znakova), cijenu (u prirodnim brojevima) i količinu prijeđenih kilometara.

- (a) Napišite deklaraciju odgovarajućeg tipa podatka za automobil na način da bude moguće definirati varijablu naredbom `auto a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- (b) Napišite funkciju `auto* izbor(auto* auti, char** nazivi, int n, int maxcijena, int maxkm)` koja prima pokazivač na listu automobila, niz stringova u kojem se nalaze nazivi automobila, duljinu tog niza te prirodne brojeve koji predstavljaju cijenu i kilometražu. Funkcija kreira novu vezanu listu u kojoj se nalaze automobili iz vezane liste `auti` čiji se naziv nalazi u nizu `nazivi`, a čija je cijena manja ili jednaka od `maxcijena` i broj prijeđenih kilometara manji ili jednak od `maxkm`.
- (c) Napišite funkciju `auto* invertiraj(auto* izbor)` koja vraća pokazivač na invertiranu vezanu listu liste `izbor`.

Napomena. Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**. Dozvoljeno je korištenje funkcija iz `stdlib.h` i `string.h`.

Rješenje:

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

typedef struct _automobil{
    char naziv[21];
    int cijena;
    int km;
    struct _automobil* next;
}automobil;

automobil* insert(automobil* first, char* naziv, int cijena, int km){
    automobil* novi=(automobil*)malloc(sizeof(automobil));
    strcpy(novi->naziv, naziv);
    novi->cijena=cijena;
    novi->km=km;
    novi->next = first;
    return novi;
}

void ispisi(automobil* first){
    int i;
    while(first){
        printf("Naziv: %s\n", first->naziv);
        printf("Cijena: %d\n", first->cijena);
        printf("Kilometraza: %d\n", first->km);
        first = first->next;
    }
}

automobil* izbor(automobil* first, char** nazivi, int n, int maxcijena, int maxkm){
    int i;
    automobil* pom;
    automobil* ifirst=NULL;
    automobil* ilast;
    automobil* pom2;
    for(pom=first ; pom ; pom=pom->next){
        for(i=0 ; i<n ; i++){
            if(strcmp(nazivi[i], pom->naziv)==0 && pom->cijena <= maxcijena && pom->km <= maxkm){
                if(ifirst==NULL){
                    ifirst=pom;
                }
            }
        }
    }
    return ifirst;
}
```

```

        ilast=ifirst;
    }
    else {
        ilast->next=pom;
        ilast=ilast->next;
    }
}
}
if(ifirst) ilast->next=NULL;
return ifirst;
}

automobil* invertiraj(automobil* izbor){
    automobil* first=NULL, *pom, *pom2;
    for(pom=izbor ; pom ; pom=pom2){
        pom2=pom->next;
        pom->next=first;
        first=pom;
    }
    return first;
}

int main(){
    int n=3;
    char** nazivi=(char**)malloc(n*sizeof(char*));
    for(int i=0 ; i<n ; i++){
        nazivi[i]=(char*)malloc(21*sizeof(char));
    }
    strcpy(nazivi[0], "BMW");
    strcpy(nazivi[1], "Mercedes");
    strcpy(nazivi[2], "Toyota");
    automobil* auti=NULL;
    auti=insert(auti, "BMW", 35000, 120000);
    auti=insert(auti, "Citroen", 6000, 250000);
    auti=insert(auti, "Toyota", 10000, 200000);
    auti=insert(auti, "VW", 20000, 50000);
    auti=insert(auti, "Mercedes", 30000, 80000);
    //ispisi(auti);
    automobil* shortlist=izbor(auti, nazivi, n, 35000, 150000);
    ispisi(shortlist);
    printf("-----\n");
    automobil* novi_shortlist=invertiraj(shortlist);
    ispisi(novi_shortlist);
    return 0;
}

```

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadatka

Zadatak 4 (bodova)

Zapis o kupovini nekog proizvoda u trgovini se čuva u strukturi `proizvod` koja se sastoji od sljedećih podataka: ime proizvoda (string od najviše 50 znakova), cijena proizvoda (double) i broj kupljenih proizvoda (int). Podaci o svim kupovinama se čuvaju u binarnoj datoteci. Isti proizvod se može pojaviti više puta u datoteci. Možete pretpostaviti da trgovina može imati maksimalno 50 različitih proizvoda. Ne moraju se svi proizvodi pojaviti u datoteci.

- Definirajte strukturu `proizvod` koja pamti navedene podatke o jednom proizvodu. Struktura mora biti definirana tako da bude moguća deklaracija oblika `proizvod p`; te smije sadržavati samo zadane podatke.
- Napišite funkciju `void obradiPodatke(FILE* in, FILE* out)` koja kao argumente prima pokazivač `in` na binarnu datoteku otvorenu za čitanje te pokazivač `out` na tekstualnu datoteku otvorenu za pisanje. Funkcija u tekstualnu datoteku upisuje ukupnu količinu kupljenih proizvoda iz binarne datoteke u sljedećem obliku "ime cijena ukupnaKolicina". Nakon svakog proizvoda ide skok u novi red. Redoslijed proizvoda nije bitan. Možete pretpostaviti da binarna datoteka nije prazna.

Primjer:

Početna binarna datoteka in:

```
{Mlijeko, 5.2, 3}, {Sir, 8.7, 1}, {Jaja, 0.5, 10}, {Mlijeko, 5.2, 1},  
{Mlijeko, 5.2, 2}, {Sir, 8.7, 5}
```

Završna tekstualna datoteka out:

```
Mlijeko 5.2 6  
Sir 8.7 6  
Jaja 0.5 10
```

Napomena: Od dodatnih nizova, dozvoljeno je **jedino korištenje jednog niza struktura proizvod duljine 50**. Nije dozvoljeno korištenje dodatnih datoteka. Dozvoljeno je korištenje zaglavlja `stdio.h`, `stdlib.h` i `string.h`.

Rješenje:

```
#include<stdio.h>  
#include<stdlib.h>  
#include<string.h>  
  
typedef struct  
{  
    char ime[51];  
    double cijena;  
    int br;  
} proizvod;  
  
int azurirajNiz(proizvod niz[], proizvod p, int n)  
{  
    int i;  
    for(i=0; i<n; i++)  
        if(strcmp(niz[i].ime,p.ime)==0)  
        {  
            niz[i].br+=p.br;  
            return n;  
        }  
  
    if(i==n)  
        niz[i] = p;  
  
    return n+1;  
}  
  
void obradiPodatke(FILE* in, FILE* out)  
{  
    proizvod niz[50];  
    proizvod temp;
```

```
int n=0;

while(fread(&temp, sizeof(proizvod),1,in)==1)
    n = azurirajNiz(niz,temp,n);

for(int i=0; i<n; i++)
    fprintf(out,"%s %g %d\n", niz[i].ime, niz[i].cijena, niz[i].br);
}
```


Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadataka

Zadatak 1 (16 bodova) Napišite program koji kao prvi argument komandne linije prima nenegativan cijeli broj, a kao drugi argument nenegativni realni broj zapisan u znanstvenoj notaciji (s eksponentom, kao kod formata %e ili %E). Program treba spremi učitane brojeve u odgovarajuće varijable tipa `int` i `double`, te ispisati njihove vrijednosti. Pretvorba za oba broja se vrši pozivom funkcije **pretvori** koja kao argumente prima string reprezentaciju učitano broja, a preko varijabilnog argumenta vraća `int` ili `double` reprezentaciju tog broja (ovisno o ulaznom broju). Napišite sve dodatne funkcije i definirajte formalne argumente funkcije **pretvori** (može ih biti i više od nabrojanih) da bi mogli ostvariti navedenu funkcionalnost. Obavezno napišite implementaciju funkcije **main**. Pripazite da izbjegnute upozorenja prevodioca uzrokovana konverzijama tipova argumenata funkcija.

Napomene: zabranjeno korištenje funkcija `atof`, `atoi`, `atol`, `strtod`, `strtoul`, `strtoul`. Dopušteno je korištenje ostalih funkcija definiranih u datotekama zaglavlja `stdlib.h` i `ctype.h`. Pretpostavite da je ulazni realni broj oblika cjelobrojnidioe(ili E)eksponent (npr. 12e3, 1234E56). Eksponent broja je u ovom zadatku uvijek nenegativan cijeli broj.

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

void pretvori(char* broj, void *(*f)(void* ulaz), void **izlaz){

    *izlaz = f(broj);

    return;
}

int *pretvoriUInt(char *ulaz){

    int *rez = (int*) malloc (sizeof(int));

    *rez = 0;

    for(int i=0;ulaz[i]!='\0';i++){

        if(isdigit(ulaz[i])){
            *rez*=10;
            *rez+=(ulaz[i]-'0');
        }
        else return NULL;
    }
    return rez;
}

double *pretvoriUDoubleZ(char *ulaz){

    double *rez = (double*) malloc (sizeof(double)), exp = 0, pot = 1;

    *rez = 0.0;

    int eznak = 0, pos = 0;

    for(int i=0;ulaz[i]!='\0';i++){
        if(isdigit(ulaz[i])){
            if(eznak == 0){
                *rez*=10;
                *rez+=(ulaz[i]-'0');
            }
            else if(eznak == 1){
                exp*=10;
                exp+=(ulaz[i]-'0');
            }
        }
    }
}
```

```

    }
    else if(ulaz[i] == 'e' || ulaz[i] == 'E'){
        eznak = 1;
    }
    else return NULL;
}

for(int i=0;i<exp;i++) pot*=10;
*rez*=pot;

    return rez;
}

void* obradiInt(void *ulaz){
    return pretvoriUInt((char*) ulaz);
}

void* obradiDoubleZ(void *ulaz){
    return pretvoriUDoubleZ((char*) ulaz);
}

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

    int* broj1;
    double *broj2;

    pretvori(argv[1],obradiInt, (void**) &broj1);

    printf("Rezultat int: %d\n",*broj1);
    free(broj1);
    broj1 = NULL;

    pretvori(argv[2],obradiDoubleZ, (void**) &broj2);

    printf("Rezultat double znanstvena: %lg\n",*broj2);
    free(broj2);
    broj2 = NULL;

return 0;
}

```

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadataka

Zadatak 2 (2+12 bodova) Nakon održanog turnira društvenih igara, organizatori su odlučili napraviti rang ljestvicu igrača koji su sudjelovali na turniru. Svaka igra koja se igrala na turniru bila je igra za 3 igrača.

Za svakog igrača na turniru znamo njegovo ime koje ga jedinstveno određuje (string duljine najviše 30) i ukupan broj ostvarenih bodova na turniru (nenegativan cijeli broj). Podatke o igraču pamtimo u strukturi `player`.

Nadalje, za svaku igru koja se igrala na turniru znamo njezin naziv (string duljine najviše 50), težinu (prirodan broj između 1 i 10, uključivo) te niz `ekipa` od 3 igrača koji su odigrali partiju te igre (niz od 3 strukture `player`). Podatke o igri pamtimo u strukturi `game`. Poznato je da je niz `ekipa` **sortiran silazno** prema broju ostvarenih bodova u odigranoj partiji. Ako sa `tezina` označimo težinu igre, broj ostvarenih bodova u toj partiji igrača na indeksu `i` računamo prema formuli $tezina * (3 - i)$.

(a) Deklarirajte tip podatka `player` koji pamti navedene podatke o igraču. Dodatno, varijabla `x` tog tipa se treba moći definirati sa `player x`; . Deklarirajte tip podatka `game` koji pamti navedene podatke o igri. Dodatno, varijabla `y` tog tipa se treba moći definirati sa `game y`;

(b) Napišite funkciju

```
void rang_lista(game* odigrane, player* igraci, int m, int n)
```

koja kao argumente dobiva niz odigranih igara na turniru u nizu `odigrane`, niz svih igrača koji su sudjelovali na turniru u nizu `igraci` (s početnim bodovima postavljenim na 0) te prirodne brojeve `m` i `n` koji su redom duljine nizova `odigrane` i `igraci`. Funkcija treba sortirati niz `igraci` prema ukupnim bodovima igrača ostvarenim na turniru i na ekran ispisati rang listu igrača u formatu:

`redni broj.ime igrača.(broj bodova).\n`. Ako postoje igrači s istim brojem bodova, poredak među njima nije bitan, ali im se ispisuje isti redni broj.

Napomena. U nizu `odigrane` je moguće imati više od jedne strukture `game` s istim nazivom i težinom igre. Nije dozvoljeno korištenje dodatnih nizova.

Primjer. Pokažimo na primjeru kako ažuriramo bodove igrača u nizu `igraci`. Za niz igrača

```
igraci: P1={Vedran, 0}, P2={Nikola, 0}, P3={Matej, 0}, P4={Ana, 0}, P5={Ema, 0}, P6={Laura, 0}
```

ako je niz `odigrane` jednak

```
{Carcassonne, 6, [P3,P1,P2]}, {Monopoly, 3, [P6,P5,P4]}, {Carcassonne, 6, [P5,P2,P1]},
```

nakon poziva funkcije `rang_lista(odigrane,igraci,3,6)`, niz `igraci` treba biti (do na poredak igrača s istim bodovima):

```
igraci: P5={Ema, 24} P2={Nikola, 18}, P3={Matej, 18}, P1={Vedran, 18}, P6={Laura, 9}, P4={Ana, 3},
```

a ispis:

1. Ema (24)
2. Nikola (18)
2. Matej (18)
2. Vedran (18)
5. Laura (9)
6. Ana (3)

Rješenje:

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

```
typedef struct{
    char ime[31];
    int bodovi;
}player;
```

```
typedef struct{
    char naziv[51];
    int tezina;
    player ekipa[3];
}game;
```

```
void rang_lista(game* odigrane, player* igraci, int m, int n)
```

```

{
    int i,j,k,indeks,redni_broj=1;

    for(i=0;i<m;i++)
        for(j=0;j<3;j++)
        {
            indeks=0;
            while(strcmp(odigrane[i].ekipa[j].ime,igraci[indeks].ime)) indeks++;
            igraci[indeks].bodovi+=odigrane[i].tezina*(3-j);
        }

    for(i=0;i<n;i++)
        for(j=i+1;j<n;j++)
            if(igraci[i].bodovi<igraci[j].bodovi)
            {
                player temp=igraci[i];
                igraci[i]=igraci[j];
                igraci[j]=temp;
            }

    printf("%d. %s (%d)\n", redni_broj,igraci[0].ime,igraci[0].bodovi);

    for(i=1;i<n;i++)
    {
        if(igraci[i-1].bodovi>igraci[i].bodovi) redni_broj=i+1;

        printf("%d. %s (%d)\n", redni_broj,igraci[i].ime,igraci[i].bodovi);
    }
}

//nije dio zadatka
int main()
{
    player igraci[6];
    game odigrane[3];

    int i;
    for(i=0;i<6;i++) igraci[i].bodovi=0;

    strcpy(igraci[0].ime,"Vedran");
    strcpy(igraci[1].ime,"Nikola");
    strcpy(igraci[2].ime,"Matej");
    strcpy(igraci[3].ime,"Ana");
    strcpy(igraci[4].ime,"Ema");
    strcpy(igraci[5].ime,"Laura");

    strcpy(odigrane[0].naziv,"Carcassonne");
    odigrane[0].tezina=6;
    odigrane[0].ekipa[0]=igraci[2]; odigrane[0].ekipa[1]=igraci[0];
    odigrane[0].ekipa[2]=igraci[1];

    strcpy(odigrane[1].naziv,"Monopoly");
    odigrane[1].tezina=3;
    odigrane[1].ekipa[0]=igraci[5]; odigrane[1].ekipa[1]=igraci[4];
    odigrane[1].ekipa[2]=igraci[3];

    strcpy(odigrane[2].naziv,"Carcassonne");
    odigrane[2].tezina=6;
    odigrane[2].ekipa[0]=igraci[4]; odigrane[2].ekipa[1]=igraci[1];
    odigrane[2].ekipa[2]=igraci[0];

    rang_lista(odigrane,igraci,3,6);
    return 0;
}

```

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadataka

Zadatak 3 (2+11 bodova) Svi automobili dostupni za prodaju u nekoj firmi nalaze se u vezanoj listi. Za svaki automobil pamtimo naziv (string s najviše 20 znakova), cijenu (u prirodnim brojevima) i godinu proizvodnje.

- Napišite deklaraciju odgovarajućeg tipa podatka za automobil na način da bude moguće definirati varijablu naredbom `auto a;`. Definirajte samo podatke koji su nužni za čuvanje takve liste u memoriji.
- Napišite funkciju `auto* izbor(auto* auti, char** nazivi, int n, int maxcijena, int godina)` koja prima pokazivač na listu automobila, niz stringova u kojem se nalaze nazivi automobila, duljinu tog niza te prirodne brojeve koji predstavljaju cijenu i godinu proizvodnje. Funkcija kreira novu vezanu listu u kojoj se nalaze automobili iz vezane liste `auti` čiji se naziv nalazi u nizu `nazivi`, a čija je cijena manja ili jednaka od `maxcijena` i godina proizvodnje veća ili jednaka od `godina`.
- Napišite funkciju `auto* invertiraj(auto* izbor)` koja vraća pokazivač na invertiranu vezanu listu liste `izbor`.

Napomena. Možete definirati dodatne (pomoćne) funkcije, ali **dodatni nizovi nisu dozvoljeni**. Dozvoljeno je korištenje funkcija iz `stdlib.h` i `string.h`.

Rješenje:

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

typedef struct _automobil{
    char naziv[21];
    int cijena;
    int godina;
    struct _automobil* next;
}automobil;

automobil* insert(automobil* first, char* naziv, int cijena, int godina){
    automobil* novi=(automobil*)malloc(sizeof(automobil));
    strcpy(novi->naziv, naziv);
    novi->cijena=cijena;
    novi->godina=godina;
    novi->next = first;
    return novi;
}

void ispisi(automobil* first){
    int i;
    while(first){
        printf("Naziv: %s\n", first->naziv);
        printf("Cijena: %d\n", first->cijena);
        printf("Godina proizvodnje: %d\n", first->godina);
        first = first->next;
    }
}

automobil* izbor(automobil* first, char** nazivi, int n, int maxcijena, int godina){
    int i;
    automobil* pom;
    automobil* ifirst=NULL;
    automobil* ilast;
    automobil* pom2;
    for(pom=first ; pom ; pom=pom->next){
        for(i=0 ; i<n ; i++){
            if(strcmp(nazivi[i], pom->naziv)==0 && pom->cijena <= maxcijena && pom->godina >= godina){
                if(ifirst==NULL){
                    ifirst=pom;
                }
            }
        }
    }
    return ifirst;
}
```

```

        ilast=ifirst;
    }
    else {
        ilast->next=pom;
        ilast=ilast->next;
    }
}
}
}
if(ifirst) ilast->next=NULL;
return ifirst;
}

automobil* invertiraj(automobil* izbor){
    automobil* first=NULL, *pom, *pom2;
    for(pom=izbor ; pom ; pom=pom2){
        pom2=pom->next;
        pom->next=first;
        first=pom;
    }
    return first;
}

int main(){
    int n=3;
    char** nazivi=(char**)malloc(n*sizeof(char*));
    for(int i=0 ; i<n ; i++){
        nazivi[i]=(char*)malloc(21*sizeof(char));
    }
    strcpy(nazivi[0], "BMW");
    strcpy(nazivi[1], "Mercedes");
    strcpy(nazivi[2], "Toyota");
    automobil* auti=NULL;
    auti=insert(auti, "BMW", 35000, 2020);
    auti=insert(auti, "Citroen", 6000, 2010);
    auti=insert(auti, "Toyota", 10000, 2018);
    auti=insert(auti, "VW", 20000, 2022);
    auti=insert(auti, "Mercedes", 30000, 2021);
    //ispisi(auti);
    automobil* shortlist=izbor(auti, nazivi, n, 35000, 2020);
    ispisi(shortlist);
    printf("-----\n");
    automobil* novi_shortlist=invertiraj(shortlist);
    ispisi(novi_shortlist);
    return 0;
}

```

Programiranje 2 – drugi kolokvij, 28. 6. 2024.

Rješenja zadataka

Zadatak 4 (5+10=15 bodova)

Zapis o kupovini nekog proizvoda u voćarni se čuva u strukturi `voce` koja se sastoji od sljedećih podataka: ime voća (string od najviše 50 znakova), cijena voća (double) i težina kupljenog voća (double). Podaci o svim kupovinama se čuvaju u binarnoj datoteci. Isti proizvod se može pojaviti više puta u datoteci. Možete pretpostaviti da trgovina može imati maksimalno 50 različitih vrsta voća. Ne moraju se svi proizvodi pojaviti u datoteci.

- Definirajte strukturu `voce` koja pamti navedene podatke o jednom proizvodu. Struktura mora biti definirana tako da bude moguća deklaracija oblika `voce v`; te smije sadržavati samo zadane podatke.
- Napišite funkciju `void obradiPodatke(FILE* in, FILE* out)` koja kao argumente prima pokazivač `in` na binarnu datoteku otvorenu za čitanje te pokazivač `out` na tekstualnu datoteku otvorenu za pisanje. Funkcija u tekstualnu datoteku upisuje ukupnu težinu kupljenih proizvoda iz binarne datoteke u sljedećem obliku "ime: cijena, ukupnaTezina". Nakon svakog proizvoda ide skok u novi red. Redoslijed proizvoda nije bitan. Možete pretpostaviti da binarna datoteka nije prazna.

Primjer:

Početna binarna datoteka `in`:

```
{Jabuka, 5.2, 3.5}, {Banana, 8.7, 1.2}, {Jagoda, 10, 0.7}, {Jabuka, 5.2, 1.2},  
{Jabuka, 5.2, 2}, {Banana, 8.7, 0.8}
```

Završna tekstualna datoteka `out`:

```
Jabuka: 5.2, 6.7  
Banana: 8.7, 2.0  
Jagoda: 10, 0.7
```

Napomena: Od dodatnih nizova, dozvoljeno je **jedino korištenje jednog niza struktura `voce` duljine 50**. Nije dozvoljeno korištenje dodatnih datoteka. Dozvoljeno je korištenje zaglavlja `stdio.h`, `stdlib.h` i `string.h`.

Rješenje:

```
#include<stdio.h>  
#include<stdlib.h>  
#include<string.h>  
  
typedef struct  
{  
    char ime[51];  
    double cijena;  
    double tezina;  
} voce;  
  
int azurirajNiz(voce niz[], voce p, int n)  
{  
    int i;  
    for(i=0; i<n; i++)  
        if(strcmp(niz[i].ime,p.ime)==0)  
        {  
            niz[i].tezina+=p.tezina;  
            return n;  
        }  
  
    if(i==n)  
        niz[i] = p;  
  
    return n+1;  
}  
  
void obradiPodatke(FILE* in, FILE* out)  
{  
    voce niz[50];  
    voce temp;
```

```
int n=0;

while(fread(&temp, sizeof(voce),1,in)==1)
    n = azurirajNiz(niz,temp,n);

for(int i=0; i<n; i++)
    fprintf(out,"%s: %g, %g\n", niz[i].ime, niz[i].cijena, niz[i].tezina);
}
```