

Strukture podataka i algoritmi 1

Kolokvijum

22.05.2021.

Na Desktop-u u direktorijumu Rad kreirati direktorijum **ImePrezime_BrIndeksa** i 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**.

Napisati C program koji:

1. Sa standardnog ulaza učitava tekst (jedna linija) i transformiše na sledeći način:

U tekstu može da se nađe izraz oblika **broj1!** ili izraz **broj1^broj2**, gde su broj1 i broj2 celi brojevi, operacija **!** je faktoriyel, a **^** označava stepen. Ukoliko se u tekstu pojave izrazi ovog oblika, potrebno ih je zameniti rezultatom odgovarajućih operacija. Napisati rekursivne funkcije za izračunavanje faktoriijela i stepena.

Primer: Tekst: "5! " menja se sa "120".

Ulaz:

Temperatura za ponedeljak je 4!. Vazdusni pritisak se procenjuje na 2^{10} . Sansa za pojavu kise je !50%. Ponesite kisobran!

Izlaz:

Temperatura za ponedeljak je 24. Vazdusni pritisak se procenjuje na 1024. Sansa za pojavu kise je !50%. Ponesite kisobran!

Bodovanje: Ukoliko se zadatak uradi sa rekursivnim funkcijama zadatak nosi 14 poena. Ako se niz ne alokira dinamički broj poena se smanjuje za 3 poena, a ako se uradi sa iterativnom umesto rekursivne funkcije broj poena se smanjuje za 7 poena.

2. Sa standardnog ulaza učitava se najpre n , a potom n realnih brojeva koji se smeštaju u jednostruko povezanu listu. Zatim se unosi ceo broj k . Listu je potrebno formirati dodavanjem elemenata na početak liste.

- Ispisati kreiranu listu.
- Nakon toga potrebno je izbaciti svaki element iz liste čije su prve dve decimale manje od unetog broja k . Ispisati elemente liste nakon izbacivanja.
- Od izbačenih elemenata potrebno je formirati novu listu dodavanjem tih elemenata u nerastućem poretku. Ispisati novu listu.

Ulaz: 5

13.456 1.756 8.9342 2.3 15.23

34

Izlaz:

Učitana lista: 15.23 2.3 8.9342 1.756 13.456

Nakon izbacivanja: 8.9342 1.756 13.456

Nova lista: 15.23 2.3

Ulaz: 7

13.456 1.756 5.64 8.9342 3.98 2.3 15.23

60

Izlaz:

Učitana lista: 15.23 2.3 3.98 8.9342 5.64 1.756 13.456

Nakon izbacivanja: 3.98 8.9342 5.64 1.756

Nova lista: 15.23 13.456 2.3

Bodovanje: Ceo zadatak nosi 12 poena. Svaka stavka nosi po 4 poena.

3. Biraju se izvođači koji će nastupati na ovogodišnjem *Arsenal festu*. Za svakog izvođača se pamte sledeći podaci: ime benda (može sadržati više reči), broj članova (ceo broj) i prosečna ocena publike (realan broj). Publika je zadužena za izbor izvođača. Putem društvenih mreža obavlja se glasanje za izbor četiri najbolje ocenjena benda. Pravo glasa imaju samo kupci karata za predstojeći festival. Svaki kupac prilikom glasanja ostavlja sledeće podatke: broj karte (ceo broj), ime benda za koji glasa (može sadržati više reči) i ocena (ceo broj, koji je u opsegu od 1 do 10).

- Definirati odgovarajuće strukture podataka koje opisuju **izvođača** i **glasača**.
- Napisati funkciju koja učitava podatke o izvođačima iz fajla **izvodjaci.txt**. U prvoj liniji fajla nalazi se ceo broj koji predstavlja ukupan broj izvođača, a zatim je za svakog izvođača dato po 2 podataka: ime benda i broj članova, prosečna ocena

publike kod svakog izvođača je na početku nula. Od učitanih podataka potrebno je formirati niz struktura izvođača.

- Napisati funkciju koja učitava podatke o glasačima iz fajla **glasaci.txt**. U prvoj liniji fajla nalazi se ceo broj koji predstavlja ukupan broj glasača, a zatim je za svakog glasača dato po 3 podataka: broj karte, ime benda i ocena benda. Od učitanih podataka potrebno je formirati niz struktura glasača.
- Napisati funkciju koja određuje prosečnu ocenu svakog benda, na osnovu ocena glasača.
- Napisati funkciju koja određuje četiri najbolje ocenjena benda.
- Napisati funkciju koja određuje najaktivnijeg glasača, odnosno osobu koja je najviše puta glasala i ispisuje njen broj karte.

Ulaz

izvodjaci.txt

6

Artan Lili

4

Dubioza Kolektiv

7

Buc Kesidi

2

Hladno pivo

5

Lollobrigida

4

Brkovi

5

glasaci.txt

11

111

Artan Lili

9

235

Dubioza Kolektiv

6

345

Buc Kesidi
7
111
Hladno pivo
9
234
Lollobrigida
9
111
Brkovi
5
555
Artan Lili
8
233
Lollobrigida
6
557
Buc Kesidi
2
889
Hladno pivo
3
889
Brkovi
6

Izlaz:

Artan Lili 8.5
Lollobrigida 7.5
Hladno pivo 6
Dubioza kolektiv 6
111

Bodovanje: Ceo zadatak nosi 14 poena. Prva stavka nosi 1 poen, druga i treća nose po 2 poena, a ostale stavke nose po 3 poena. Ukoliko se nizovi ne formiraju dinamički oduzima se 2 poena.

Vreme za izradu kolokvijuma je 150 minuta.

Rešenja:

```
1. #include<stdio.h>
   int ucitajbroj(int *broj){
       int c;
       c=getchar();
       while(c>='0' && c<='9'){
           *broj=(*broj*10)+ c-'0';
           c=getchar();
       }
       return c;
   }

   int faktorijel(int n){
       if(n==0 || n==1) return 1;
       else return n*faktorijel(n-1);
   }

   int stepen(int a, int b){
       if (b==0) return 1;
       else return a*stepen(a,b-1);
   }

   int main(){
       int c,a;
       while ((c=getchar())!='\n'){
           if(c>='0' && c<='9'){
               int broj=c-'0';
               c = ucitajbroj(&broj);
               a=broj;

               if(c=='!'){
                   printf("%d",faktorijel(a));
               }
               else if(c=='^'){
                   broj =0;
                   broj=getchar()-'0';
                   c=ucitajbroj(&broj);

                   printf("%d", stepen(a,broj));
               }
           }
       }
   }
```

```

        putchar(c);
    }
    else{printf("%d", broj);
        putchar(c);}

    }
    else putchar(c);
}
printf("\n");

}

```

```

2. #include<stdio.h>
#include<stdlib.h>
/* 15 12 11 null
13
*/

typedef struct el{
    float vrednost;
    struct el *sledeci;
}el;
el *alociraj(float v){
    el *nov=(el*)malloc(sizeof(el));
    if(nov==NULL){
        printf("GRESKA\n");
        exit(1);
    }
    nov->vrednost=v;
    nov->sledeci=NULL;
    return nov;
}
void dodajNaPocetak(el **glava,float v){
    el *nov=alociraj(v);
    if(*glava==NULL){
        *glava=nov;
    }else{
        nov->sledeci=*glava;
        *glava=nov;
    }
}

```

```

}
void dodajUNerastuci(el **glava,float v){
    el *nov=alociraj(v);
    if(*glava==NULL){
        *glava=nov;
    }else{
        el *pom=*glava,*pret=*glava;
        if(v>(*glava)->vrednost){
            nov->sledeci=*glava;
            *glava=nov;
        }else{
            for(pom=*glava;pom!=NULL &&
(pom->vrednost)>v;pret=pom,pom=pom->sledeci);
            nov->sledeci=pom;
            pret->sledeci=nov;
        }
    }
}

void stampaj(el *glava){
    while(glava!=NULL){
        printf("%f ",glava->vrednost);
        glava=glava->sledeci;
    }
    putchar('\n');
}

void obradi(el **glava,el **nova,int k){
    el *pom=*glava,*pret=*glava;
    int n;
    while(pom!=NULL){
        n=((int)(pom->vrednost *100))%100;
        if(n<k){
            if(pom==(*glava))
            {
                dodajUNerastuci(nova,pom->vrednost);
                (*glava)=(*glava)->sledeci;
                pret=(*glava);
                free(pom);
                pom=*glava;
                continue;
            }else{
                pret->sledeci=pom->sledeci;
                dodajUNerastuci(nova,pom->vrednost);
                free(pom);
            }
        }
    }
}

```

```

        }
    }
    pret=pom;
    pom=pom->sledeci;
}
}
int main(){
    int n,k;
    float v;
    el *glava=NULL,*nova=NULL;
    scanf("%d",&n);
    for(k=0;k<n;k++){
        scanf("%f",&v);
        dodajNaPocetak(&glava,v);

    }
    scanf("%d",&k);
    stampaj(glava);
    obradi(&glava,&nova,k);
    stampaj(glava);
    stampaj(nova);

}

```

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

```

typedef struct izvodjac{
    char ime_benda[31];
    int broj_clanova;
    float prosecna_ocena;
}Izvodjac;

```

```

typedef struct glasac{
    int broj_karte;
    char ime_benda[31];
    int ocena;
}Glasac;

```

```

struct izvodjac* unosIzvodjaca(int n,FILE *f){

```

```

    int i;
    Izvodjac *izvodjac;

    izvodjac = (struct izvodjac*)malloc(n*sizeof(struct izvodjac));

    for(i = 0; i < n; i++){
        fgetc(f);
        fgets(izvodjac[i].ime_benda,31,f);
        fscanf(f,"%d",&izvodjac[i].broj_clanova);
        izvodjac[i].prosecna_ocena = 0;
    }
    return izvodjac;
}

void stampalzovjdaca(Izvodjac *iz,int n){
    int i;
    for(i = 0; i < n; i++){
        printf("%s %d\n",iz[i].ime_benda,iz[i].broj_clanova,iz[i].prosecna_ocena);
    }
}

struct glasac* unosGlasaca(int n,FILE *g){
    int i;
    Glasac *glasaci;
    glasaci = (struct glasac*)malloc(n*sizeof(struct glasac));

    for(i = 0; i < n; i++){
        fscanf(g,"%d",&glasaci[i].broj_karte);
        fgetc(g);
        fgets(glasaci[i].ime_benda,31,g);
        fscanf(g,"%d",&glasaci[i].ocena);
    }

    return glasaci;
}

void stampaGlasaca(Glasac *g,int n){
    int i;
    for(i = 0; i < n; i++){
        printf("%d %s %d\n",g[i].broj_karte,g[i].ime_benda,g[i].ocena);
    }
}

```

```
}
```

```
void oceni(Izvodjac *iz,Glasac *g,int duzinaGlasaca,int duzinalzvodjaca){  
    int i,j,br = 0;  
  
    for(i = 0; i < duzinalzvodjaca; i++){  
        for(j = 0; j < duzinaGlasaca; j++){  
            if(strcmp(iz[i].ime_benda,g[j].ime_benda) == 0){  
                iz[i].prosecna_ocena += g[j].ocena;  
                br++;  
            }  
        }  
        iz[i].prosecna_ocena /= br;  
        br = 0;  
    }  
}
```

```
void cetiriNajbolja(Izvodjac *iz,int duzinalzvodjaca){  
    int i,j;  
    Izvodjac tmp;  
  
    for(i = 0; i < duzinalzvodjaca; i++){  
        for(j = i; j < duzinalzvodjaca; j++){  
            if(iz[i].prosecna_ocena < iz[j].prosecna_ocena){  
                tmp = iz[i];  
                iz[i] = iz[j];  
                iz[j] = tmp;  
            }  
        }  
    }  
    stampalzvovjdaca(iz,4);  
}
```

```
void prebrojGlasaca(Glasac *g,int duzinaGlasaca){  
    int maxi,max = 0,i,j,niz[duzinaGlasaca];  
  
    for(i = 0; i < duzinaGlasaca; i++){  
        niz[i] = 0;  
    }  
  
    for(i = 0; i < duzinaGlasaca; i++){  
        for(j = i + 1; j < duzinaGlasaca; j++){  
            if(g[i].broj_karte == g[j].broj_karte){  
                niz[i]++;  
            }  
        }  
    }  
}
```

```

        }
    }
    for(i = 0; i < duzinaGlasaca; i++){
        if(max < niz[i]){
            max = niz[i];
            maxi = i;
        }
    }

    printf("%d\n",g[maxi].broj_karte);
}

int main(){
    FILE *f,*g;
    Izvodjac *izvodjaci;
    Glasac *glasaci;
    int duzinaGlasaca,duzinalzvodjaca;

    f = fopen("izvodjaci.txt","r");
    g = fopen("glasaci.txt","r");

    fscanf(f,"%d",&duzinalzvodjaca);
    fscanf(g,"%d",&duzinaGlasaca);

    izvodjaci = unosIzvodjaca(duzinalzvodjaca,f);
    stampaIzvodjaca(izvodjaci,duzinalzvodjaca);

    printf("\n-----\n");

    glasaci = unosGlasaca(duzinaGlasaca,g);
    stampaGlasaca(glasaci,duzinaGlasaca);

    printf("\n-----\n");

    oceni(izvodjaci,glasaci,duzinaGlasaca,duzinalzvodjaca);

    cetiriNajbolja(izvodjaci,duzinalzvodjaca);

    printf("\n-----\n");

    prebrojGlasaca(glasaci,duzinaGlasaca);

```

```
    return 0;  
}
```