

Strukture podataka i algoritmi 1

(zadatak - max 30 poena)

Avgust, 2020

Lanac restorana IMIMeal omogućava zakazivanje večere unapred. Za svaki restoran se zna njegov ID (ceo broj, jedinstven), njegovo ime (niz karaktera), broj različitih vrsta stolova (ceo broj), a za svaki vrstu stola broj komada i broj gostiju koji može da sedne (celi brojevi). Ukoliko grupa koja dolazi ne može da sedne za jedan sto, moguće je spojiti više stolova, uz pretpostavku da se ne gubi mesto za sedenje, a nije obavezno da sva mesta za stolovima budu popunjena. Jednom raspoređeni stolovi se ne pomeraju. Prilikom zakazivanja večere grupa daje svoje ime (niz karaktera, jedinstven), broj članova grupa i ID restorana za koji žele da zakažu. Rezervacija se vrši za čitavo veče, tako da nema smenjivanja gostiju za stolom. Ukoliko je grupa veća od 10 članova, njena rezervacija se automatski odbija. Ukoliko u željenom restoranu nema mesta, grupi se nudi rezervacija u restoranu u kome ima mesta za grupu i pri tome ima najviše slobodnih mesta. Grupa može da prihvati ili odbije ponudu. Prioritet pri rezervisanju mesta je redosled prijavljivanja. Napisati program u kome se učitavaju podaci o restoranima, zatim podaci o rezervacijama, realizuju rezervacije, a potom se za dva uneta imena grupa ispisuje da li su u istom restoranu ili ne. Na kraju ispisati koliko rezervacija nije realizovano.

(4 poena)

Za rešavanje problema napisati sledeće funkcije:

a) Definirati sve potrebne složene tipove podataka neophodne za rešavanje opisanog problema.

(3 poena, obavezno)

b) Napisati funkciju **UcitajRestorane** koja iz datoteke *Restorani.txt* učitava podatke o restoranima i formira listu/niz restorana.

(3+2 poena, obavezno)

c) Napisati funkciju **NadjiRestoran** koja za dati ID restorana vraća pokazivač na određeni restoran, odnosno NULL ukoliko restoran ne postoji.

(3 poena, obavezno)

d) Napisati funkciju **UcitajRezervacije** koja iz datoteke *Rezervacije.txt* učitava podatke o grupama gostiju i formira listu/niz grupa.

(4 poena)

e) Napisati funkciju **Rezervisanje** koja za određenu grupu vrši rezervaciju restorana.

(4+3 poena)

f) Napisati funkciju **DveGrupe** koja za date nazive dve grupe ispisuje da li su u istom restoranu i ispisuje odgovarajući komentar.

(4 poena)

Pri učitavanju podataka podrazumevati da su svi podaci korektno zadati

Dozvoljeno je proširivanje struktura i definisanje novih, kao i definisanje dodatnih funkcija.

Zadatak se može rešavati korišćenjem povezanih lista, nizova ili kombinacijom.

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

Strukture podataka i algoritmi 1
Test – max 20 poena

Avgust, 2020

| Ime i prezime | Broj indeksa | Broj poena |
|---------------|--------------|------------|
|---------------|--------------|------------|

1. (1.5) Šta je rezultat sledećih kodova?

```
int a = 51;  
printf("%d", a | 8);
```

59

```
int a = 91;  
printf("%d", a ^ 12)
```

87

```
int a = 27;  
printf("%d", (~a) ^ 16);
```

-12

2. (2.0) Sortiraj niz brojeva 17 3 22 13 78 30 54 47 6 28 u rastućem poretku koristeći Insertion sort i ispisati svaki korak prilikom sortiranja brojeva.

```
17 3 22 13 78 30 54 47 6 28  
3 17 22 13 78 30 54 47 6 28  
3 17 22 13 78 30 54 47 6 28  
3 13 17 22 78 30 54 47 6 28  
3 13 17 22 78 30 54 47 6 28  
3 13 17 22 30 78 54 47 6 28  
3 13 17 22 30 54 78 47 6 28  
3 13 17 22 30 47 54 78 6 28  
3 6 13 17 22 30 47 54 78 28  
3 6 13 17 22 28 30 47 54 78  
3 6 13 17 22 28 30 47 54 78
```

3. (2.5) Data je struktura:

```
struct node{  
    float x;  
    struct node* next;  
};
```

```
struct node* head;
```

Napisati niz komandi kojima se formira lista od prvih 10 elemenata Fibonačijevog niza.

```
struct node *temp, *novi;  
head = (struct node*)malloc(sizeof(struct node));  
head->x = 1.0;  
temp = head;  
float prvi = 0, drugi = 1;  
int i;  
for(i=1;i<10;i++){  
    novi=(struct node*)malloc(sizeof(struct node));  
    novi->x = prvi + drugi;  
    novi->next = NULL;  
    prvi = drugi;  
    drugi = novi->x;  
    temp->next=novi;  
    temp=novi;  
}
```

4. (3.0) Za strukturu iz zadatka 3 napisati funkciju koja za dati pokazivač na početak liste briše element koji je po vrednosti najbliži srednjoj vrednosti svih elemenata liste.

```
struct node* obrisi(struct node *p){
    struct node *temp;
    int n;
    float avg = 0.0;
    temp = p;
    while(temp){
        avg += temp->x;
        n++;
        temp = temp->next;
    }
    avg /=n;

    float min;
    struct node *preth;
    preth = NULL;
    temp = p;
    min = abs(temp->x - avg);
    while(temp->next){
        if(abs(temp->next->x - avg) < min){
            preth = temp;
            min = abs(temp->next->x - avg);
        }
        temp = temp->next;
    }
    if(preth){
        temp = preth->next;
        preth->next = temp->next;
        free(temp);
    }
    else{
        temp = p;
        p = temp->next;
        free(temp);
    }
    return p;
}
```

5. (1.0) Napisati uslovni izraz koji odgovara sledećem kodu

```
if(!x/4)
    y = x * 4;
else if(x%2==0)
    y = sqrt(x, 2);
else
    y = x * x;
```

```
y = (!x/4)? x*4 : (x%2 ? x*x: sqrt(x));
```

6. (1.0) Zaokružiti izraze/izraz kojim treba zameniti x tako da izraz $a[i][j][k][l]=x$ bude tačan

a. $(**(*(*a+i)+j)+k)+l$

b. $*(**(*(*a+i)+j)+k)+l$

c. $*(((a+i)+j)+k+l)$

d. $*(a+i)+j+k+l$

7. (0.5) Koliko memorijskog prostora (u bajtovima) zauzima promenljiva koja je tipa struct geom_figura ukoliko promenljiva tipa float zauzima 4 bajta i double 8 bajtova?

```
struct geom_figura {
    union {
        struct { double r; } kruznicica;
        struct { double a; } kvadrat;
        struct { double a, b; } pravougaonik;
        struct { double a, b, c; } trougao;
    } figura;
    union{
        float x;
        double y;
    }krd;
};
```

32

8. (3.0) Napisati program koji sadrži rekurzivnu funkciju koja izvršava bitovsku operaciju pomeranja u levo za dva zadata broja x i y, tako što broj x pomeri u levo y puta. Brojevi x i y su predati prilikom pokretanja programa na sledeći način:

```
./pomeri 4 5
```

```
#include <stdio.h>
int pomeri(int x, int y){
    if (y)
        return pomeri(x,y-1)<<1;
    return x;
}
main(int argc, char **argv)
{
    int x,y;
    x = atoi(argv[1]);
    y = atoi(argv[2]);
    printf("%d %d %d",x,y, pomeri(x,y));
}
```

9. (1.0) Šta je rezultat sledećeg koda?

```
#include <stdio.h>
main()
{
    int w=10;
    float y=4.5;
    if (y/w) printf("%d\n", (int)(--w?y*y:-w));
    w?printf("Test 1\n"):printf("%d\n", (2+w)/++w);
}
```

20

Test 1

10. (0.5) Šta je rezultat sledećeg koda?

```
#include <stdio.h>
#define STAR(x,y,z) (x/y+z)
main() {
    int a = 3, b = 5;
    float c = 7.5;
    printf("%.2f\n", STAR(a+b, b-a, a+c));
}
```

11.50

11. (2.0) Šta je rezultat sledećeg koda?

```
#include <stdio.h>
main(){
    unsigned x, y;
    scanf("%d%d", &x, &y);

    while (x>0)
    {
        y <<= (x>=1)/3;
        printf("%d %d\n", x, y);
    }
}
```

Ako se kao vrednosti promenljivih x i y unesu 31 i 5

15 160

7 640

3 1280

1 1280

0 1280

12. (2.0) Šta je rezultat sledećeg koda?

```
char *str[] = { "februar", "mart", "april", "maj"};
char **s1 = str;
int n = sizeof(str)/sizeof(str[0]);
printf("%d\n", n);
while(n--)
    printf("%c \n", *(++++(*s1++)));
```

b

r

r

j