

# Strukture podataka i algoritmi 1

## (zadatak - max 30 poena)

Jun, 2017

U koordinatnoj ravni date su tačke svojim koordinatama i oznakama i duži oznakama krajnjih tačaka. Napisati program u kome se iz datoteke učitava broj tačaka, zatim za svaku tačku oznaka (niz karaktera, jedna reč),  $x$  i  $y$  koordinata (realni brojevi), sva tri podatka u jednoj liniji. Nakon podataka o tačkama iz datoteke se učitava najpre broj duži, a zatim za svaku duž oznake krajnjih tačaka. Sve tačke imaju različite oznake, ali mogu imati iste koordinate. Duplikate tačaka izbaciti i pri tome ažurirati podatke u nizu duži. Za duži koje imaju iste obe krajnje tačke izbrisati duplikate iz niza duži. Svaku duž kojoj pripada neka tačka iz niza tačaka (ne računajući krajnje tačke), takođe, izbrisati iz niza duži. Ispitati da li date duži čine zatvorenu izlomljenu liniju, ako se kao kriterijum uzme da svaka tačka je krajnja tačka za tačno dve duži ili nije krajnja tačka ni jedne duži i u tom slučaju odrediti dužinu te linije.

**(3 poena)**

Za rešavanje problema napisati sledeće funkcije:

a) Napisati funkciju **UcitajTacke** koja iz u datoteke učitava i formira niz tačaka.

**(2 poena)**

b) Napisati funkciju **NadjiTacku** koja za zadatu oznaku tačke vraća pokazivač na tu tačku iz niza tačaka.

**(2 poena)**

c) Napisati funkciju **UcitajDuzi** koja iz datoteke učitava podatke o tačkama i formira niz duži čije su krajnje tačke predstavljene pokazivačima na odgovarajuće elemente niza tačaka.

**(2 poena)**

d) Napisati funkciju **DuplikatiTacaka** koja će iz niza tačaka izbaciti duplikate i pri tome ažurirati niz duži na odgovarajući način.

**(9 poena)**

e) Napisati funkciju **DuplikatiDuzi** koja će iz niza duži obrisati duplikate duži.

**(3 poena)**

f) Napisati funkciju **NepotrebneDuzi** koja iz niza duži briše sve duži koje sem krajnjih tačaka sadrže još neku tačku iz niza tačaka.

**(4 poena)**

g) Napisati funkciju **Poligon** koja ispituje da li dati niz duži formira zatvorenu izlomljenu liniju.

**(3 poena)**

h) Napisati funkciju **Obim** koja određuje obim poligona.

**(2 poena)**

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

Jednačina prave kroz tačke  $(x_1, y_1)$  i  $(x_2, y_2)$ :  $y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$

EksPLICITNI oblik jednačine prave:  $y = kx + n$ , gde je  $k$  koeficijent pravca prave, a  $n$  odsečak na  $Y$  osi

Strukture podataka i algoritmi 1  
Test – max 20 poena

Jun, 2017

1. Definirati **main** funkciju za zadatak o tačkama i dužima.
2. Definirati strukturu linija koja sadrži informaciju da li se radi o pravoj ili krugu i omogućava da se za pravu zapamte dve tačke, a za krug tri. Svaka tačka je određena sa dve realne koordinate u koordinatnoj ravni.
3. Odrediti vrednost promenljive  $x$  posle izvršenja sledećih nizova naredbi

a.

```
int x=1;
int y=0;
int z;
z = ((-x) ? 2 : -2) + 2;
x = z+++y;
```

b.

```
int x=1;
x <= 2;
x &= 0x345;
x *=x+2;
```

4. 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.

5. Definirati enumerisani tip podataka gde će konstante JEDAN, CETIRI, STO, HILJADU imati numeričke vrednosti koje odgovaraju njihovom imenu.
6. Dodati šta nedostaje u sledećem kodu

```
f(){
    x=2;
    printf("%d %d %d\n",x,a,*b);
    return x*2;
}
```

```
main(){
    int x,y,z;
    z=7;
    printf("%d %d %d\n",x,y,z);
    printf("%d %d %d\n",x,y,f(y,&z));
}
```

Tako da se nakon startovanja dobije rezultat

```
5    6    7
2    9    14
5    6    4
```

7. Navesti oznake ispred "brzih" algoritma za sortiranje niza

- a. Selection
- b. Quick
- c. Shell

- d. Stack
- e. Odd-even sort
- f. Insertion

- g. Merge
- h. Binary

8. Matematička funkcija je definisana na sledeći način  $y = \begin{cases} \frac{\sqrt{x^2+2}}{x+1} & |x-3| \geq 5 \\ \sqrt{x} & \text{inače} \end{cases}$

A odgovarajuća funkcija u C-u definisana na sledeći način

```
if (x<=-2 || x>=8)
    printf("%f\n",sqrt(x*x+2)/(x+1));
if (x>0)
    printf("%f\n",sqrt(x));
else
    printf("Nedefinisana vrednost\n");
```

Definisati test primere kojima bi se pokazala ispravnost/neispravnost funkcije u C-u.

9. Struktura je definisana na sledeći način

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

Definisati niz elemenat tipa date strukture i funkciju koja će sa ulaza učitati broj elemenata nizu (unapred nije poznato ograničenje za broj elementa u nizu) i podatke za sve članove niza. Funkcija treba da vrati pokazivač na prvi element niza.

10. Šta će biti rezultat rada sledećih kodova

a.

```
enum {false,true};
int main(){
    int i=1;
    do {
        printf("%d\n",i);
        i++;
        if(i < 15)
            continue;
    }while(false);
    return 0;
}
```

b.

```
#include <stdio.h>
int main()
{
    float a = 12.5;
    printf("%d\n", a);
    printf("%d\n", *(int *)&a);
    return 0;
}
```

11. Ispraviti grešku u sledećem kodu

```
#include <stdio.h>
int main()
{
    int a = 1,2;
    printf("a : %d\n",a);
    return 0;
}
```

12. Šta je opisano sledećim deklaracijama:

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

bonus

- d. char ((\*f())[])();
- e. int \*\*(\*f)(int\*\*,int\*\*(\*) (int \*\*,int \*\*));