

JMBAG:		Ime i prezime:	
--------	--	----------------	--

## Programiranje i programsko inženjerstvo

Zimski ispitni rok 12. veljače 2015.

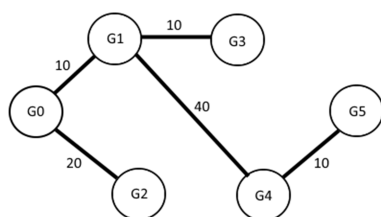
Odgovore na 1., 2., 3. i 4. pitanje napišite na svojim papirima, a na 5. i 6. pitanje na listu s zadacima.

### Zadatak 1. (25 bodova)

Udaljenosti između gradova zadane su dvodimenzijским poljem koje se naziva "matrica susjedstva". Ako je element dvodimenzijskog polja  $a_{ij} = 0$ , tada iz grada  $s$  indeksom  $i$  nije moguće doći do grada  $s$  indeksom  $j$ . Ako je element  $a_{ij} > 0$ , tada on sadrži udaljenost između grada  $s$  indeksom  $i$  i grada  $s$  indeksom  $j$  u kilometrima. Matrica je simetrična s obzirom na glavnu dijagonalu pa vrijedi  $a_{ij} = a_{ji}$ .

- Napisati funkciju `fUdaljenost` koja preko parametara prima dvodimenzijско polje s udaljenostima između gradova i jednodimenzijско polje s indeksima gradova koje je potrebno posjetiti redom kako su navedeni u polju (pretpostavite da su zadani indeksi unutar matrice sustava i da nema duplikata). Ako je moguće posjetiti sve gradove tim redoslijedom, funkcija mora vratiti ukupnu udaljenost koju je potrebno prijeći kako bi se posjetili zadani gradovi. U suprotnom funkcija mora vratiti vrijednost 0.
- Napisati glavni program u kojem je s tipkovnice potrebno učitati broj gradova iz intervala  $[3, 10]$  te njihovu matricu susjedstva. Nakon toga s tipkovnice učitati broj gradova koje je potrebno posjetiti (minimalno 3 grada) i njihove indekse. Zatim je potrebno pozvati funkciju iz a) dijela zadatka i ispisati udaljenost. Ako se zadani gradovi ne mogu posjetiti jedan za drugim ispisati poruku „Zadani gradovi ne mogu se posjetiti“.

**Primjer:** Zadano je 6 gradova i njihova matrica susjedstva. Na slici je vidljivo da udaljenost od grada G0 do grada G1 iznosi 10 km (element  $a_{01}$  odnosno element  $a_{10}$  matrice susjedstva), a udaljenost od grada G0 do grada G2 iznosi 20 km (element  $a_{02}$  odnosno element  $a_{20}$  matrice susjedstva), itd.



0	10	20	0	0	0
10	0	0	10	40	0
20	0	0	0	0	0
0	10	0	0	0	0
0	40	0	0	0	10
0	0	0	0	10	0

Ako je potrebno posjetiti gradove 0, 1, 4 i 5 tada funkcija `fUdaljenost` mora vratiti vrijednost 60 jer je ukupna udaljenost  $10 + 40 + 10 = 60$  km. Ako je potrebno posjetiti gradove 0, 1, 3 i 5 tada funkcija `fUdaljenost` mora vratiti vrijednost 0 jer iz grada G3 nije moguće direktno doći do grada G5.

### Zadatak 2. (10 bodova)

Napisati funkciju `odrediZadatke` koja određuje zadatke za laboratorijsku vježbu koja ima dvije grupe zadataka. Funkcija prima redne brojeve zadataka u obliku jednodimenzijских polja, a vraća po jedan redni broj zadatka za svaku grupu određen slučajnim odabirom. Pretpostaviti da je generator slučajnih brojeva inicijaliziran prije poziva funkcije.

**Primjer 1:**

Grupa1: 2, 1, 5, 4, 3

Grupa2: 10, 9, 8, 7, 11

Povratna vrijednost: 4, 8

**Primjer 2:**

Grupa1: 15, 2, 11, 22, 16, 3, 9, 1

Grupa2: 5, 3, 8, 10, 7

Povratna vrijednost: 16, 3

### Zadatak 3. (15 bodova)

- Napisati vlastitu implementaciju funkcije `strrchr` bez korištenja drugih ugrađenih funkcija iz zaglavlja `string.h`. Prototip funkcije je: `char *moj_strrchr(const char *s, int c);`
- Napisati glavni program u kojem je s tipkovnice potrebno učitati niz `s` (maksimalne duljine 30 znakova) i znak `c`. Pomoću funkcije `moj_strrchr` pronađite gdje se u nizu `s` nalazi zadnji znak `c`, te koristeći dobiveni rezultat ispišite sadržaj niza `s` od početka niza do pronađenog znaka `c`.

**Primjeri:**

Za učitani znak `b` i učitani niz **Abra kad abra**, glavni program na zaslon ispisuje **Abra kad ab**

Za učitani znak `e` i učitani niz **Abra kad abra**, glavni program na zaslon ispisuje **U zadanom nizu nema znaka e**

#### Zadatak 4. (20 bodova)

Zadana je tekstualna datoteka "rezultati.txt" koja sadrži rezultate ispita za studente u obliku:

1000344#2#3.5#4#5

1000003#5.5#3#2.5#8

Podaci su razdvojeni znakom #. Prvi stupac predstavlja šifru studenta (šesteroznamenkasti cijeli broj), dok su ostali stupci ostvareni bodovi po zadacima (realni brojevi). Na ispitu je ukupno 4 zadataka.

Binarna datoteka "ukupno.dat" sadrži ostvarene bodove studenta po zadacima i ukupan ostvareni broj bodova na ispitu. Svaki zapis sadrži šifru studenta (šesteroznamenkasti cijeli broj), bodove po zadacima i ukupan broj bodova na ispitu (realan broj). Redni broj zapisa u datoteci odgovara šifri studenta umanjenoj za 1000000. Pretpostavlja se da u datoteci inicijalno postoje zapisi za sve studente u kojima su bodovi po zadacima i ukupni bodovi postavljeni na 0.

Napisati program koji će na temelju rezultata ispita za studente iz datoteke "rezultati.txt", pohraniti bodove po zadacima i ukupan broj bodova u datoteci "ukupno.dat". Na standardni izlaz potrebno je ispisati šifru studenta koji je pisao ispit i informaciju o rezultatu na ispitu na način da za one zapise gdje je ukupan rezultat veći od prosjeka piše "iznad", a za one ispod prosjeka "ispod".

1000344#ispod

1000003#iznad

#### Zadatak 5. (5 bodova)

Kolika je vrijednost pohranjena u varijablu tipa `float` ako je njezin sadržaj:

a) 01111111111111111111111111111111<sub>2</sub>

Rješenje: \_\_\_\_\_

b) 42150000<sub>16</sub>

Rješenje: \_\_\_\_\_

#### Zadatak 6. (5 bodova)

Zadana je funkcija koja računa iznos poreza  $p$  koji je potrebno platiti ovisno o cijeni artikla  $c$  po sljedećoj formuli:

$$p = \begin{cases} 0,1 * c, & c \leq 1000 \\ 100 + 0,2 * (c - 1000), & 1000 < c \leq 5000 \\ 500 + 0,3 * (c - 5000), & c > 5000 \end{cases}$$

U prostoru s desne strane napišite programski odsječak za skretnicu (switch) označen s `/*nadopuniti*/`

```
float f(float c) {  
    int kat; float p;  
    if (c <= 1000) {  
        kat = 1;  
    } else if (c <= 5000) {  
        kat = 2;  
    } else {  
        kat = 3;  
    }  
    switch (kat) {  
        /*nadopuniti*/  
    }  
    return p;  
}
```

## Zadatak 1

```
#include <stdio.h>
#include <stdlib.h>
#define MIN 3
#define MAX 10

int fUdaljenost(int *matrSus, int maxStup, int *gradovi, int nGP) {
    int k, trenG, sljedG, udaljenost = 0;
    trenG = gradovi[0];
    for (k = 1; k < nGP; k++) {
        sljedG = gradovi[k];
        if (matrSus[trenG * maxStup + sljedG] == 0) {
            udaljenost = 0;
            break;
        }
        udaljenost += matrSus[trenG * maxStup + sljedG];
        trenG = sljedG;
    }
    return udaljenost;
}

int main(void) {
    int i, j, nG, nGP, matrSus[MAX][MAX] = { 0 }, gradovi[MAX] = { 0 };
    int udaljenost;

    do {
        printf("\nUnesite broj gradova [%d, %d]: ", MIN, MAX);
        scanf("%d", &nG);
    } while (nG < MIN || nG > MAX);
    for (i = 0; i < nG; i++) {
        for (j = 0; j < i; j++) {
            if (i == j) {
                continue;
            }
            printf("\nUnesite udaljenost izmedju grada G%d i grada G%d: ", i, j);
            scanf("%d", &matrSus[i][j]);
            matrSus[j][i] = matrSus[i][j];
        }
    }
    do {
        printf("\nUnesite broj gradova koje je potrebno posjetiti [%d, %d]: ", MIN, nG);
        scanf("%d", &nGP);
    } while (nGP < MIN || nGP > nG);
    for (i = 0; i < nGP; i++) {
        printf("\nUnesite indeks %d. grada (%d/%d): ", i + 1, i + 1, nGP);
        scanf("%d", &gradovi[i]);
    }

    udaljenost = fUdaljenost(&matrSus[0][0], MAX, gradovi, nGP);

    if (udaljenost == 0) {
        printf("\nZadani gradovi ne mogu se posjetiti.");
    } else {
        printf("\nUkupna udaljenost posjecenih gradova iznosi %d.", udaljenost);
    }
    return 0;
}
```

## Zadatak 2

```
#include <stdlib.h>
void odrediZadatke(int *grupa1, int duljina1, int *grupa2, int duljina2, int *zadaci){
    *(zadaci + 0) = *(grupa1 + rand() % duljina1);
    *(zadaci + 1) = *(grupa2 + rand() % duljina2);
}
```

## Zadatak 3 - funkcija

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

```
char *moj_strchr(const char *s, int c) {
    char *rez = NULL;
    while (*s) {
        if (*s == c) {
            rez = s;
        }
        s++;
    }
    if (rez != NULL) {
        return rez;
    } else {
        /*null-character smatra se dijelom znakovnog niza pa se ova funkcija može
        koristiti kako bi se dobio pokazivač na kraj znakovnog niza*/
        return c == '\\0' ? s : NULL;
    }
}
```

/\*Alternativno rješenje\*/

```
char *moj_strchr_1(const char *s, int c) {
    char *rez = NULL;
    do {
        if (*s == c) {
            rez = s;
        }
    } while (*s++);
    return rez;
}
```

### Zadatak 3 – glavni program

```
int main(void) {
    char c, s[30 + 1], *p;
    int i = 0;

    printf("\nUnesite niz znakova: ");
    gets(s);
    printf("\nUnesite znak c: ");
    scanf("%c", &c);

    p = moj_strchr(s, c);

    if (p == NULL) {
        printf("U zadanom nizu nema znaka %c", c);
    } else {
        do {
            printf("%c", *(s + i));
        } while ((s + i++) != p);
    }

    /*Alternativno rješenje 1*/
    /*
    if (p == NULL) {
        printf("U zadanom nizu nema znaka %c", c);
    } else {
        do {
            printf("%c", *(s + i));
            i++;
        } while ((s + i) != p);
        printf("%c", *(s + i));
    }
    */
    /*Alternativno rješenje 2*/
    /*
    if (p == NULL) {
        printf("U zadanom nizu nema znaka %c", c);
    } else {
        while ((s + i) != p) {
            printf("%c", *(s + i));
            i++;
        }
        printf("%c", *(s + i));
    }
    */

    return 0;
}
```

#### Zadatak 4

```
#include <stdio.h>
#define BR_ZAD 4
int main(void){
    FILE *fin, *foutst;
    float sumauk, prosjekuk;
    int br, i;
    char tmp;
    struct {
        int sifra;
        float bodovi[BR_ZAD];
        float suma;
    } zapis;

    fin = fopen("rezultati.txt", "r");
    foutst = fopen("ukupno.dat", "w+b");

    br = sumauk = 0;
    while (fscanf(fin, "%d%c%f%c%f%c%f%c\n", &zapis.sifra, &tmp, &zapis.bodovi[0],
        &tmp, &zapis.bodovi[1], &tmp, &zapis.bodovi[2], &tmp, &zapis.bodovi[3])
        == BR_ZAD + 5){
        zapis.suma = 0;
        for (i = 0; i < BR_ZAD; i++){
            zapis.suma += zapis.bodovi[i];
        }

        fseek(foutst, (long)(zapis.sifra - 1000000) * sizeof(zapis), SEEK_SET);
        fwrite(&zapis, sizeof(zapis), 1, foutst);
        sumauk += zapis.suma;
        br++;
    }
    prosjekuk = sumauk / br;

    fseek(foutst, 0L, SEEK_SET);
    while (fread(&zapis, sizeof(zapis), 1, foutst) == 1) {
        printf("%d#%s\n", zapis.sifra, zapis.suma > prosjekuk ? "iznad" : "ispod");
    }

    fclose(fin);
    fclose(foutst);

    return 0;
}
```

#### Zadatak 5

- a) NaN
- b) 37.25

#### Zadatak 6

```
case 1:
    p = 0.1 * c;
    break;
case 2:
    p = 100 + 0.2 * (c - 1000);
    break;
case 3:
    p = 500 + 0.3 * (c - 5000);
```