

JMBAG		Prezime, ime	
-------	--	--------------	--

## Uvod u programiranje

Izvanredni ispitni rok - 14. rujna 2021.

### Napomene:

- Programska rješenja moraju biti usklađena sa stilom pisanja programa koji je propisan na predmetu
- Rješenja zadataka 1 - 3 napisati na vlastitim papirima i predati ih u košuljici
- Rješenja zadataka 4 - 6 napisati na za to predviđeno mjesto uz tekst zadatka te list sa zadacima predati u košuljici

### 1. (21 bod)

Svaki zapis *binarne* datoteke `bodovi.bin` sadrži JMBAG studenta (niz znakova 10+1 znak), te niz od 5 cijela broja (polje tipa *int*) u rasponu [0, 10] koji predstavljaju ostvarene bodove na pojedinom zadatku ispita. U *tekstualnu* datoteku `prosli.txt` potrebno je u svaki redak zapisati JMBAG-ove studenata koji su na zadacima ostvarili sumu bodova jednaku ili veću od 25, te njihovu ostvarenu sumu bodova.

Također, u zadnji redak datoteke `prosli.txt` potrebno je u obliku decimalnog broja s dva decimalna mjesta ispisati ukupan prosjek bodova svih studenata koji su navedeni u toj datoteci. U slučaju da nije bilo studenata koji su zadovoljili uvjet, prosjek bodova se smatra 0.

Datoteke je dozvoljeno otvoriti **samo jednom**. Nije dozvoljeno korištenje **pomoćnih polja** za pohranu međurezultata.

bodovi.bin

0064532171	4	5	6	8	9
0122542329	1	1	3	4	7
5258063877	10	8	9	2	4
3085353870	3	10	7	8	9
1692637190	4	4	7	10	2
...	...	...	...	...	...

prosli.txt

```
0064532171 • 32.
5258063877 • 33.
3085353870 • 37.
1692637190 • 27.
Prosjek: • 32.25
```

### 2. (21 bod)

Napišite funkciju koja u zadanom nizu traži zadani podniz te vraća broj pojavljivanja podniza. Za implementaciju potrebno je koristiti odgovarajuće funkcije iz standardne biblioteke. Funkcija ne smije koristiti nijedan pomoćni niz.

Prototip je:

```
int brojPojavljivanja(char *niz, char *podniz)
```

U glavnom programu učitajte niz i podniz sa standardnog ulaza te pomoću funkcije **brojPojavljivanja** pronađite broj pojavljivanja podniza. Ako se podniz nalazi u nizu, na standardni izlaz ispišite broj pojavljivanja podniza, u suprotnom ispišite poruku "*Ne postoji*". Niz i podniz može sadržavati razmake i sigurno neće imati više od 80 znakova.

Primjeri izvođenja programa:

```
Ulaz: >Godina ima 365 dana, a dan
24 sata.
dan.
Izlaz: >Uzorak se pojavljuje 2 puta.
```

```
Ulaz: >Zagreb.
zag.
Izlaz: >Ne postoji.
```

### 3. (21 bod)

Kvadratna matrica sadrži nenegativne cjelobrojne vrijednosti (tipa *int*).

Napisati funkciju **zbrojiZnamenke** koja kao parametar prima cjelobrojnu vrijednost (tipa *int*), a kao rezultat vraća sumu znamenki zadanog broja.

Napisati funkciju **nadjiSume** koja kao parametar prima matricu i broj, a zadatak joj je unutar zadane matrice pronaći broj elemenata čija suma znamenki je jednaka zadanom broju. Za pronalazak sume znamenki funkcija **nadjiSume** treba koristiti funkciju **zbrojiZnamenke**.

Napisati funkciju glavnog programa (main) u kojoj se od korisnika traži unos dimenzije matrice, unos njezinih elemenata i broj koji predstavlja traženu sumu, te se uz pomoć funkcije **nadjiSume** izračunava i na ekran ispisuje broj elemenata matrice čija suma znamenki je jednaka zadanom broju. Primjer: za zadanu matricu i broj 9, u glavnom programu se treba ispisati 4 jer 4 elementa matrice imaju zbroj znamenki jednak 9.

<u>153</u>	29	1	345
13	0	24	759
4	<u>2223</u>	7	<u>54</u>
29	10	17	<u>630</u>

### 4. (6 bodova)

Definirane je sljedeća struktura:

```
struct Osoba {
    char oib[12];
    char ime[21];
    char prezime[21];
};
```

```
...
printf("%s", _____);
```

A large rectangular area with a grid of dashed lines, intended for drawing a picture. The grid consists of 10 columns and 4 rows of squares.

## Rješenja

1.

```
#include <stdio.h>

struct Bodovi {
    char jmbag[11];
    int bodovi[5];
};

// za potrebe testiranja
void createBodoviBin() {
    FILE *bin = fopen("bodovi.bin", "wb");

    struct Bodovi bodovi[5] = {{ "0064532171", {4, 5, 6, 8, 9}},
                                { "0122542329", {1, 1, 3, 4, 7}},
                                { "5258063877", {10, 8, 9, 2, 4}},
                                { "3085353870", {3, 10, 7, 8, 9}},
                                { "1692637190", {4, 4, 7, 10, 2}}};

    for (int i = 0; i < 5; i++) {
        fwrite(&bodovi[i], sizeof(struct Bodovi), 1, bin);
    }

    fclose(bin);
}

int main(void) {
    FILE *f_bodovi = fopen("bodovi.bin", "rb");
    FILE *f_prosli = fopen("prosli.txt", "w");

    struct Bodovi zapis;
    int ukBodova = 0;
    int brZapisa = 0;

    while (fread(&zapis, sizeof(struct Bodovi), 1, f_bodovi)) {

        int zapisUkBodova = 0;
        for (int i = 0; i < 5; i++)
            zapisUkBodova += zapis.bodovi[i];

        if (zapisUkBodova >= 25) {
            ukBodova += zapisUkBodova;
            brZapisa++;

            fprintf(f_prosli, "%s %d\n", zapis.jmbag, zapisUkBodova);
        }
    }

    float prosjek = brZapisa == 0 ? 0 : ukBodova / (1. * brZapisa);

    fprintf(f_prosli, "Prosjek: %.2f", prosjek);

    fclose(f_bodovi);
    fclose(f_prosli);
    return 0;
}
```

2.

```
int brojPojavljivanja(char *niz, char *uzorak){
    int br = 0;
    for(int i = 0; niz[i] != '\0'; i++){
        int nadjen = 1;

        for(int j=0; uzorak[j] != '\0'; j++){
            if(niz[i] != uzorak[j]){
                nadjen = 0;
                break;
            }
            i++;
        }
        if(nadjen){
            br++;
        }
    }
    return br;
}

---- ili ----

int brojPojavljivanja(char *niz, char *uzorak){
    int br = 0;
    int len = strlen(uzorak);
    for(; niz = strstr(niz, uzorak); niz+=len, br++){
        return br;
    }

int main(void){
    char niz[MAX + 1], uzorak[MAX+1];
    scanf("%[^\\n]*c%[^\\n]", niz, uzorak);
    int br = brojPojavljivanja(niz, uzorak);
    if (br > 0 ) {
        printf("Uzorak se pojavljuje %d puta\\n", br);
    } else {
        printf("Ne postoji\\n");
    }
    return 0;
}
```

3.

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

int zbrojiZnamenke(int x) {
    int sum = 0;
    while (x != 0) {
        sum += x % 10;
        x = x / 10;
    }
    return sum;
}

int nadjiSume(int m, int a[m][m], int br) {
    int brojac = 0;
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < m; j++) {
            if (zbrojiZnamenke(a[i][j]) == br) {
                brojac++;
            }
        }
    }
    return brojac;
}

int main(void) {
    int m, rez, suma = 0;
    printf("Unesite dimenziju matrice: ");
    scanf("%d", &m);
    printf("Unesite trazenu sumu: ");
    scanf("%d", &suma);
    int a[m][m];
    for (int i = 0; i < m; i++) {
        for (int j = 0; j < m; j++) {
            scanf("%d", &a[i][j]);
        }
    }
    rez = nadjiSume(m, a, suma);

    printf("%d", rez);

    return 0;
}
```

4.

```
printf("%s", p_osobe[2].prezime);
printf("%s", (p_osobe + 2)->prezime);
printf("%s", (*(p_osobe + 2)).prezime);
```

5.

```
switch (znak) {  
    case 'A':  
    case 'a':  
        rezultat = znak;  
        break;  
    case 90:  
    case 91:  
    case 92:  
    case 93:  
    case 94:  
    case 95:  
    case 96:  
        rezultat = znak * 10;  
        znak = 'X';  
        break;  
    default:  
        rezultat = 0;  
        break;  
}
```

6.

```
srand((unsigned int) time(NULL));  
double v = (double)rand() / RAND_MAX * (210 - 120) + 120  
ili rand() % (210-120+1) +120
```

7.

X=3 Y=4

8.

uci