

Programiranje i programsko inženjerstvo – izvanredni ispitni rok*18. rujna 2013.*

Nije dopušteno korištenje globalnih i statičkih varijabli te naredbe **goto**. Ispit nosi maksimalno 80 bodova, a prag za prolaz pismenog ispita je **40** bodova. Ovaj obrazac morate predati.

Zadatak 1. (20 bodova)

Napisati funkciju **rasporedi** koja će elemente iz zadanog jednodimenzionalnog polja pozitivnih cijelih brojeva nasumično rasporediti u dvodimenzionalno cjelobrojno polje pri čemu se elementi ne smiju stavljati preko već raspoređenih elemenata. Elementi dvodimenzionalnog polja na kojima nije raspoređen niti jedan element iz jednodimenzionalnog polja moraju imati vrijednost -1.

Funkcija preko ulaznih parametara prihvaća jednodimenzionalno polje pozitivnih cijelih brojeva, dvodimenzionalno cjelobrojno polje i ostale parametre potrebne za rad s jednodimenzionalnim i dvodimenzionalnim poljem.

Primjer: neka je zadano jednodimenzionalno polje 2, 3, 4, 7, 8, 9 koje je potrebno rasporediti u dvodimenzionalno polje s 4 retka i 3 stupca. Jedan mogući rezultat poziva funkcije može biti:

4	9	-1
-1	2	8
-1	-1	3
7	-1	-1

Napomena: Možete pretpostaviti da je broj elementa jednodimenzionalnog polja manji od maksimalnog broja elemenata dvodimenzionalnog polja zadanog brojem redaka i brojem stupaca.

Zadatak 2. (20 bodova)

Direktna binarna datoteka **studenti.bin** sadrži zapise o studentima koji pristupaju ispitu iz predmeta *Programiranje i programsko inženjerstvo*. Zapis se sastoji od šifre studenta (cijeli broj), šifre dvorane (cijeli broj) te oznake reda (cijeli broj) i oznake stupca (cijeli broj) kojima je određeno mjesto na kojemu student sjedi u dvorani. Redni broj zapisa u datoteci odgovara šifri studenta pri čemu nema praznih zapisa u datoteci.

Tekstualna datoteka **dvorane.txt** sadrži popis svih dvorana u kojima se održava ispit. Svaki se redak u datoteci sastoji od šifre dvorane (cijeli broj), broja redova (cijeli broj) i broja stupaca (cijeli broj) u dvorani. Podatci u pojedinom retku datoteke su međusobno odvojeni znakom #:

12#10#5

gdje je 12 šifra dvorane, 10 broj redova a 5 broj stupaca u dvorani. Ukupan broj mjesta u toj dvorani je 50 (10 redova x 5 stupaca).

Napisati program koji će studente iz datoteke **studenti.bin** rasporediti u dvorane zadane u datoteci **dvorane.txt**. Šifru dvorane te redak i stupac u koje je student raspoređen potrebno je zapisati u datoteku **studenti.bin** za svakog studenta. Studente je unutar pojedine dvorane potrebno rasporediti slijedno – prvi student u prvi redak (indeks 0) i prvi stupac (indeks 0), drugi student u prvi redak (indeks 0) i drugi stupac (indeks 1) itd.

Primjer: Ako u datoteci **dvorane.txt** imamo definirane 3 dvorane ukupnog kapaciteta 120 mjesta (50 + 30 + 40):

1#10#5

2#5#6

3#4#10

Neka se u datoteci **studenti.bin** nalaze zapisi za 100 studenata. Prvih 50 studenata iz datoteke **studenti.bin** program mora rasporediti u prvu dvoranu, sljedećih 30 studenata program mora rasporediti u drugu dvoranu a zadnjih 20 studenata program mora rasporediti u treću dvoranu.

Napomene:

Možete pretpostaviti da je u datoteci **dvorane.txt** definirano dovoljno dvorana s dovoljnim brojem mjesta da se u njih mogu rasporediti svi studenti iz datoteke **studenti.bin**.

Nije potrebno provjeravati uspješnost otvaranja i zatvaranja datoteka te uspješnost pozicioniranja i pisanja u datoteku.

Zadatak 3. (10 bodova)

Napisati **macro definiciju** NTICLAN koja vraća vrijednost za n -ti član reda kojim se računa broj $\frac{1}{4}$. n -ti član reda računa se prema formuli $\frac{1}{n*(n+1)*(n+2)}$.

Koristeći macro definiciju NTICLAN napišite program koji će s tipkovnice pročitati broj članova reda $1 \leq n \leq 1000$ (broj članova reda potrebno je učitavati sve dok se ne unese ispravna vrijednost), a zatim izračunati broj $\frac{1}{4}$. Na zaslon ispisati vrijednost izračunatu pomoću sume n članova reda, te apsolutnu vrijednost pogreške proračuna izračunate kao razliku između vrijednosti dobivene sumom reda i vrijednosti $\frac{1}{4}$.

Zadatak 4. (10 bodova)

Napisati vlastitu implementaciju ugrađene funkcije strcmp bez korištenja drugih ugrađenih funkcija iz zaglavlja string.h. Osim ulaznih argumenata nije dozvoljeno korištenje drugih pomoćnih varijabli. Prototip funkcije je:

```
int *moj_strcmp(const char *str1, const char *str2)
```

Zadatak 5. (20 bodova)

- a) **(10 bodova)** Napisati funkciju transponiraj koja kao ulazni parametar prihvaća cjelobrojnu matricu proizvoljnih dimenzija $m \times n$ a u pozivajući program vraća transponiranu matricu. Transponirana matrica matrice A je matrica A^T koja je definirana sa $[A^T]_{ij} = A_{ji}$. Npr. neka je zadana matrica A dimenzija 2×3

$$A = \begin{bmatrix} 2 & 3 & 7 \\ 1 & 0 & 1 \end{bmatrix}. \text{ Transponirana matrica matrice A je } A^T = \begin{bmatrix} 2 & 1 \\ 3 & 0 \\ 7 & 1 \end{bmatrix}$$

- b) **(10 bodova)** Napisati program u kojem će se s tipkovnice učitati broj redaka i broj stupaca iz intervala $[1, 10]$ (učitavanje je potrebno ponavljati dok se ne unesu ispravne vrijednosti). Zatim je matricu potrebno popuniti nasumičnim vrijednostima iz intervala $[0, 999]$, pozvati funkciju transponiraj i dobivenu transponiranu matricu ispisati na zaslon.

Rješenja:

Zadatak 1. (20 bodova)

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

void rasporedi(unsigned int *ulaz, int nUlaz, int izlR, int izlS, int izlMaxStup, int *izlaz){
    int i,j, k;
    /*Postavi vrijednost -1 u 2D polje zadano brojem redaka i brojem stupaca*/
    for(i = 0; i < izlR; i++){
        for(j = 0; j < izlS; j++){
            izlaz[i * izlMaxStup + j] = -1;
        }
    }
    /*Prolaz kroz sve elemente 1D polja*/
    for(k = 0; k < nUlaz; k++){
        do{
            /*Generiramo nasumičnu poziciju dok ne nađemo slobodno mjesto*/
            i = rand() % izlR;
            j = rand() % izlS;
        } while (izlaz[i * izlMaxStup + j] != -1);
        /*Upis vrijednosti u 2D polje*/
        izlaz[i * izlMaxStup + j] = ulaz[k];
    }
}
```

Zadatak 2. (20 bodova)

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

typedef struct {
    int sifStud;
    int sifDvorana;
    int redak;
    int stupac;
} student;

int main(){
    FILE *fStudenti, *fDvorane;
    int i, imaStud = 1;
    int sifDvorane, brRedaka, brStupaca, stupac, redak;
    int brMjestaUDvorani;
    student s;

    fStudenti = fopen("studenti.bin", "r+b");
    fDvorane = fopen("dvorane.txt", "r");

    /*Citamo podatke o dvoranama*/
    while(fscanf(fDvorane, "%d#%d#%d", &sifDvorane, &brRedaka, &brStupaca) == 3){
        brMjestaUDvorani = brRedaka * brStupaca; /*Odredi broj mjesta u dvorani*/
        /*Dohvacamo onoliko studenata koliko ima mjesta u dvorani*/
        stupac = redak = 0;
        for(i = 0; i < brMjestaUDvorani; i++){
            /*Procitaj sljedeći zapis iz datoteke studenti.bin*/
            if(fread(&s, sizeof(student), 1, fStudenti) == 1){
                /*Smjesti studenta u sljedeće slobodno mjesto*/
                s.sifDvorana = sifDvorane;
                s.stupac = stupac;
                s.redak = redak;
                /*Zapisi podatke u datoteku*/
                fseek(fStudenti, sizeof(student) * (-1), SEEK_CUR);
                fwrite(&s, sizeof(student), 1L, fStudenti);
                /*Povećaj brojce redaka i stupaca*/
                stupac++;
                if(stupac == brStupaca){
                    redak++;
                    stupac = 0;
                }
            }else{
                /*Došli smo do kraja datoteke studenti.bin tj. nema više studenata*/
                imaStud = 0;
                break;
            }
        }
        if(imaStud == 0){
            break;
        }
    }

    fclose(fStudenti);
    fclose(fDvorane);

    return 0;
}
```

Zadatak 3. (10 bodova)

```
#define NTICLAN(n) (1.0/((n)*((n) + 1)* ((n) + 2)))
#define MIN_CLANOVA 1
#define MAX_CLANOVA 1000

int main(){
    int brClanova, i;
    float rez = 0;
    do{
        printf("Unesite broj clanova niza iz intervala [1, 1000]:\n");
        scanf("%d", &brClanova);
    }while(brClanova < MIN_CLANOVA || brClanova > MAX_CLANOVA);

    for(i = 1; i <= brClanova; i++){
        rez += NTICLAN(i);
    }

    printf("Izracunata vrijednost: %f\n", rez);
    printf("Aposultna pogreska: %f\n", rez - 0.25);
}
```

Zadatak 4. (10 bodova)

```
int moj_strcmp(const char *str1, const char *str2){
    /*Petlja ide do kraja prvog niza*/
    while(*str1 != '\0'){
        if(*str1 != *str2){
            if(*str1 > *str2){
                /*str1 je veci*/
                return 1;
            }else{
                /*str1 je manji*/
                return -1;
            }
        }
        /*Pomakni pokazivace*/
        str1++;
        str2++;
    }
    if(*str2 == '\0'){
        /*Nizovi su jednaki*/
        return 0;
    }else{
        /*str1 je manji*/
        return -1;
    }
}
```

Zadatak 5. (20 bodova)

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

#define MAX_RED 10
#define MAX 1000

void transponiraj(int *mat, int brRed, int brStup, int maxStup, int *izlaz){
    int i,j;
    for(i = 0; i < brRed; i++){
        for(j = 0; j < brStup; j++){
            izlaz[j * maxStup + i] = mat[i * maxStup + j];
        }
    }
}

int main(){
    int mat[MAX_RED][MAX_RED];
    int trans[MAX_RED][MAX_RED];
    int i, j, brStup, brRed;

    /*Ucitaj dimenzije matrice*/
    do{
        printf("Upisite broj redaka:\n");
        scanf("%d", &brRed);
    }while(brRed < 1 || brRed > MAX);

    do{
        printf("Upisite broj stupaca:\n");
        scanf("%d", &brStup);
    }while(brStup < 1 || brStup > MAX_RED);

    /*Generiraj elemente matrice*/
    srand((unsigned)time(NULL));
    for(i = 0; i < brRed; i++){
        for(j = 0; j < brStup; j++){
            mat[i][j] = rand() % MAX;
        }
    }

    /*Poziv funkcije*/
    transponiraj(&mat[0][0], brRed, brStup, MAX_RED, &trans[0][0]);
    /*Ispis transponirane matrice*/
    for(i = 0; i < brStup; i++){
        for(j = 0; j < brRed; j++){
            printf("%4d ", trans[i][j]);
        }
        printf("\n");
    }
}
```