

## Programiranje i programsko inženjerstvo – 3. ispitni rok

3. 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. (23 boda)

Tekstualna datoteka **bodovi.txt** sadrži zapise o rezultatima studenata na predmetu *Programiranje i programsko inženjerstvo*. Svaki se redak u datoteci sastoji od šifre studenta (cijeli broj) te bodova iz dva međuispita i tri laboratorijske vježbe (realni brojevi s dvije decimale). Podatci u pojedinom retku datoteke su međusobno odvojeni jednom prazninom, npr:

1234 18.50 32.50 7.55 6.88 4.00

gdje je 1234 šifra studenta, 18.50 i 32.50 ostvareni broj bodova na prvom i drugom međuispitu a 7.55, 6.88 i 4.00 ostvareni broj bodova na prvoj, drugoj i trećoj laboratorijskoj vježbi.

- a) **(8 bodova)** Napisati funkciju `ocijeni` koja kao ulazni parametar prihvaća jednodimenzionalno realno polje s ostvarenim bodovima na međuispitima i laboratorijskim vježbama za jednog studenta i u pozivajući program vraća ukupan broj bodova (realni broj) i ocjenu (cijeli broj) koju je student ostvario. Ocjena se određuje na osnovu sljedećih pragova:

**50.0 (2); 58.0 (3); 76.0 (4); 92.0 (5)**

- b) **(15 bodova)** Napisati glavni program koji će stvoriti novu direktnu binarnu datoteku **studenti.bin** i u nju upisati zapise za sve studente iz datoteke **bodovi.txt** koji su ostvarili pozitivnu ocjenu. Zapis se sastoji od šifre studenta (cijeli broj), ukupnog broja bodova ostvarenog na predmetu (realni broj) i ocjene (cijeli broj). Redni broj zapisa u datoteci mora odgovarati šifri studenta. Ukupan broj bodova i ocjena određuju se pozivanjem funkcije `ocijeni`.

**Napomena:** nije potrebno provjeravati uspješnost otvaranja i zatvaranja datoteka te uspješnost pozicioniranja i pisanja u datoteku.

### Zadatak 2. (12 bodova)

- a) **(4 boda)** Dekadski broj je pohranjen prema standardu IEEE 754 u dvostrukoj preciznosti. Heksadekadski vrijednost sadržaja registra je 401C000000000000. Koji je to dekadski broj?

---

- b) **(4 boda)** Dekadski broj je pohranjen prema standardu IEEE 754 u dvostrukoj preciznosti. Heksadekadski vrijednost sadržaja registra je 0008000000000000. Koji je to dekadski broj?

---

- c) **(4 boda)** Prikažite, u heksadekadskom obliku, sadržaj registra u kojem je prema standardu IEEE 754 u dvostrukoj preciznosti pohranjen broj  $2^{-1022}$

---

### Zadatak 3. (15 bodova)

- a) **(10 bodova)** Napisati funkciju `brojNiz` koja zadani pozitivan cijeli broj (uključujući i broj 0) pretvara u znakovni niz. Prototip funkcije je:

```
void brojNiz(char *niz, unsigned int broj)
```

- b) **(5 bodova)** Napisati glavni program u kojem će se s tipkovnice učitati pozitivan cijeli broj, pozvati funkcija `brojNiz` i na ekran ispisati znakovni niz koji se dobije kao rezultat funkcije.

**Napomena:** Nije dozvoljeno korištenje ugrađenih funkcija za pretvaranje cijelog broja u niz znakova (`sprintf`, `itoa`).

### Zadatak 4. (30 bodova)

- a) **(15 bodova)** Napisati funkciju `provjeriSimetricnost` koja kao ulazni parametar prihvaća cjelobrojnu kvadratnu matricu proizvoljnih dimenzija  $n \times n$  te vraća 1 ako je matrica simetrična, odnosno 0 ako matrica nije simetrična. Matrica  $A = [a_{ij}]$  je simetrična ako vrijedi  $A^T = A$ , tj.  $a_{ij} = a_{ji}$  za svaki  $i, j \in [1, n]$ .

Npr. matrica  $A =$ 

1	0	1	-3
0	1	4	1
1	4	1	0
-3	1	0	2

 je simetrična jer vrijedi  $A^T = A$ .

- b) **(15 bodova)** Napisati glavni program u kojem će se s tipkovnice učitati dimenzija matrice iz intervala  $[10, 100]$  i zatim elementi matrice iz intervala  $[-10, 10]$ . Ako je korisnik unio broj izvan intervala potrebno je ponoviti učitavanje. Nakon toga potrebno je ispisati matricu, pozvati funkciju `provjeriSimetricnost` te na kraju ispisati je li unesena matrica simetrična ili nije, ovisno o rezultatu izvođenja funkcije `provjeriSimetricnost`.

## Rješenja:

### Zadatak 1. (23 boda)

```
#include <stdio.h>
#define BR_PROVJERA 5
```

```
typedef struct {
    int sifStud;
    float bodovi;
    int ocjena;
} zapis;
```

a)

```
/*Ukupan broj bodova vraća se preko parametra a ocjena kao povratna vrijednost*/
int ocijeni(float bodoviProvjere[], int brProvjera, float* bodoviUkupno){
    int i;
    *bodoviUkupno = 0;
    /*Zbroji bodove*/
    for(i = 0; i < brProvjera; i++){
        *bodoviUkupno += bodoviProvjere[i];
    }
    /*Odredi ocjenu*/
    if(*bodoviUkupno < 50.0){
        return 1;
    }else if(*bodoviUkupno < 58.0){
        return 2;
    }else if(*bodoviUkupno < 76.0){
        return 4;
    }else{
        return 5;
    }
}
```

b)

```
int main(){
    FILE *fBodovi = NULL, *fStudenti = NULL;
    float bodovi[BR_PROVJERA], bodoviUkupno;
    int sifStudent, ocjena;
    zapis stud;
    /*Otvranje datoteka*/
    fBodovi = fopen("bodovi.txt", "r");
    fStudenti = fopen("studenti.bin", "wb");
    /*Čitamo jedan po jedan zapis u datoteci s bodovima*/
    while(fscanf(fBodovi, "%d %f %f %f %f %f", &sifStudent, &bodovi[0], &bodovi[1], &bodovi[2],
        &bodovi[3], &bodovi[4]) == BR_PROVJERA + 1){

        ocjena = ocijeni(bodovi, BR_PROVJERA, &bodoviUkupno); /*Ocijeni studenta*/
        if(ocjena > 1){
            /*Ako je ocjena pozitivna zapisujemo podatke o studentu u datoteku*/
            stud.sifStud = sifStudent;
            stud.bodovi = bodoviUkupno;
            stud.ocjena = ocjena;
            fseek(fStudenti, (sifStudent - 1) * sizeof(zapis), SEEK_SET);
            fwrite(&stud, sizeof(zapis), 1, fStudenti);
        }
    }
    /*Zatvaranje datoteka*/
    fclose(fBodovi);
    fclose(fStudenti);
    return 0;
}
```

## Zadatak 2. (12 bodova)

- a) 7 (K=1024, M=11000....0)
- b)  $2^{-1023} \approx 1.1125369 \cdot 10^{-308}$  (K=0, M=1000...0)
- c) 0x0010 0000 0000 0000 (K=1, M=0)

## Zadatak 3. (15 bodova)

a)

```
void brojNiz(char *niz, unsigned int broj){
    int pom = broj;
    /*Pomakni pokazivac za onoliko mjesta koliko ima znamenaka*/
    do {
        pom = pom / 10;
        niz++;
    } while(pom);
    /*Na kraj niza stavljamo \0*/
    *niz = '\0';
    /*Pretvaranje broja u niz znakova*/
    do {
        *--niz = broj % 10 + '0';
        broj = broj / 10;
    } while(broj);
}
```

b)

```
#include <stdio.h>

int main() {
    char niz[10];
    unsigned int broj;
    /*Ucitaj broj*/
    scanf("%u", &broj);
    /*Pretvori broj u niz*/
    brojNiz(niz, broj);
    /*Ispis niza*/
    printf("%s", niz);
    return 0;
}
```

## Zadatak 4. (30 bodova)

a) (15 bodova)

```
int provjeriSimetricnost(int *mat, int n, int maxStup) {
    int i, j;
    for (i = 0; i < n; i++) {
        for (j = 0; j < i; j++){
            if ( *(mat + i*maxStup + j) != *(mat + j*maxStup + i)) {
                return 0;
            }
        }
    }
    return 1;
}
```

b) (15 bodova)

```
#include <stdio.h>
#define MAX 100

int main() {
    int mat[MAX][MAX];
    int n, i, j, k;
    /*Ucitaj dimenzije matrice*/
    do{
        /*Ponavljaj učitavanje dok se ne unese broj iz intervala [10, 100]*/
        printf("\nUnesite dimenziju matrice: ");
        scanf("%d", &n);
    }while(n < 10 || n > 1000);
    /*Ucitaj elemente matrice*/
    for (i = 0; i < n; i++){
        for (j = 0; j < n; j++) {
            do {
                /*Ponavljaj učitavanje dok se ne unese broj iz intervala [-10, 10]*/
                printf("Element [%d][%d]: ", i,j);
                scanf ("%d", &mat[i][j]);
            } while (mat[i][j] < -10 || mat[i][j] > 10);
        }
    }
    /*Ispis matrice*/
    printf("\n Ispis matrice:\n-----\n");
    for (i = 0; i < n; i++){
        for (j = 0; j < n; j++){
            printf("%d ", mat[i][j]);
        }
        printf("\n");
    }
    /*Poziv funkcije za provjeru simetricnosti*/
    if (provjeriSimetricnost(&mat[0][0], n, MAX)){
        printf ("\n\nMatrica je simetricna\n");
    } else {
        printf ("\n\nMatrica NIJE simetricna\n");
    }
    return 0;
}
```