

**Programiranje i programsko inženjerstvo**  
**Završni ispit**  
**23.01.2009**

**Napomena za sve zadatke:**

**Nije dopušteno korištenje goto naredbe, te statičkih i globalnih varijabli.**

1. (8 bodova). Napisati funkciju čiji je prototip:

```
void zamijeni (char *nizStari, char *nizNovi);
```

koja će niz znakova nizStari zadan u formatu "PREZIME, IME" prepisati u niz nizNovi, i to tako da niz nizNovi bude formata "IME PREZIME". Nije dozvoljeno mijenjanje niza nizStari.

Napisati glavni program u kojem se učitava niz znakova u formatu "PREZIME, IME" (pretpostavite da će niz biti unesen u tom formatu) te ispisuje u formatu "IME PREZIME".

**Primjer:** Za učitani niz "ANIC, ANA" potrebno je ispisati "ANA ANIC".

2. (7 bodova)

a) Napisati funkciju statistika\_matrice koja za zadanu kvadratnu znakovnu matricu **proizvoljne veličine** analizira znakove pohranjene iznad glavne dijagonale. Funkcija vraća broj znamenki i broj slova.

Npr. za matricu

1	2	Z	9	z
m	*	a	a	6
0	0	b	k	\$
3	H	0	D	#
p	K	g	!	D

Funkcija vraća 3 za broj znamenki i 5 za broj slova.

b) U priloženom glavnom programu označeno je mjesto na kojem se treba nalaziti naredba za poziv funkcije statistika\_matrice. Kao rješenje **b)** dijela zadatka potrebno je napisati poziv funkcije statistika\_matrice (ne treba prepisivati program).

```
#define MAXM 100
```

```
int main() {  
    char mat[MAXM][MAXM] = {{'1', '2', 'Z', '9', 'z'},  
                             {'m', '*', 'a', 'a', '6'},  
                             {'0', '0', 'b', 'k', '$'},  
                             {'3', 'H', '0', 'D', '#'},  
                             {'p', 'K', 'g', '!', 'D'}};
```

```
    int broj_znamenki, broj_slova;
```

```
    ***OVDJE DOLAZI POZIV FUNKCIJE statistika_matrice***
```

```
    printf("Broj znamenki iznad dijagonale je %d, a broj slova je %d",  
           broj_znamenki, broj_slova);
```

```
    return 0;
```

```
}
```

### 3. (7 bodova)

a) Napisati funkciju koja će vratiti pseudoslučajni broj u intervalu [-100, 100]. Funkcija treba imati prototip: `int fun1(void);`

b) Napisati funkciju koja će testirati ponašanje funkcije iz **a)** dijela zadatka. Funkciju `fun1` treba pozvati 1000 puta te za tako generirane brojeve izračunati aritmetičku sredinu i standardnu devijaciju. Izračunate vrijednosti vratiti u glavni program. Standardna devijacija računa se po formuli:

$$StDev = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}, \text{ gdje je } \bar{x} \text{ aritmetička sredina}$$

4. (8 bodova) Studenti procjenjuju kvalitetu nastave iz predmeta odgovaranjem na 20 postavljenih pitanja ocjenama 1 do 5. Studentske procjene spremaju se u neformatiranu datoteku **anketa.bin**. Jedan zapis neformatirane datoteke **anketa.bin** sadrži šifru predmeta (int) i 20 ocjena (int).

U direktnoj neformatiranoj datoteci **predmeti.bin** svaki zapis sadrži šifru (long) i naziv predmeta (50+1 znak). Redni broj zapisa datoteke **predmeti.bin** odgovara šifri predmeta. Prazan zapis prepoznaje se po šifri predmeta 0.

Napišite glavni program u kojem će se učitati šifra predmeta te pomoću podataka iz datoteka **predmeti.bin** i **anketa.bin** ispisati tablica učestalosti pojavljivanja pojedinih ocjena za zadani predmet u sljedećem formatu (prva dva reda nije potrebno ispisivati i služe za lakše formatiranje ispisa):

```
          2          3          4          5          6          7
1234567890123456789012345678901234567890123456789012345678901234567890
```

```
Predmet          1    2    3    4    5
-----
Programiranje i programsko inženjerstvo  xxxx  xxxx  xxxx  xxxx  xxxx
```

Ukoliko je zadana šifra predmeta koja ne postoji u datoteci **predmeti.bin** ispisati odgovajuću poruku i završiti s izvođenjem programa.

**Nije potrebno provjeravati uspješnost obavljanja operacija nad datotekama.**

## Rješenja

1.

```
void zamijeni (char *nizStari, char *nizNovi) {
    int duljina;
    char *p;
    p = strchr(nizStari, ',');
    duljina = p - nizStari + 1;
    strcpy(nizNovi, nizStari + duljina + 1);
    strcat(nizNovi, " ");
    strncat(nizNovi, nizStari, duljina - 1);
}

int main() {
    char nizStari [100], nizNovi [100];
    gets(nizStari);
    zamijeni(nizStari, nizNovi);
    printf("%s %s\n", nizStari, nizNovi);
    return 0;
}
```

Rješenje s pomoćnim poljima:

```
void zamijeni (char *nizStari, char *nizNovi) {
    char pom1[100], pom2[100];
    int i = 0, j = 0;

    while(nizStari[i] != ',') {
        pom1[i] = nizStari[i];
        i++;
    }
    pom1[i] = '\\0';

    i++;
    while(nizStari[i] != '\\0') {
        if (nizStari[i] != ' ') {
            pom2[j] = nizStari[i];
            j++;
        }
        i++;
    }
    pom2[j] = '\\0';

    strcpy(nizNovi, pom2);
    strcat(nizNovi, " ");
    strcat(nizNovi, pom1);
}
```

2.

```
#define MAXN 10
```

```
void statistika_matrice(char *matrica, int rang, int max_rang, int* broj_znamenki,
int* broj_slova){
    int i=0; int j=0; char trenutni_element;
    *broj_znamenki=0; *broj_slova=0;
    for(i=0; i<rang; i++){
        for(j=0; j<rang; j++){
            trenutni_element = *(matrica + i*max_rang + j);
            if(j>i){
                if(isdigit(trenutni_element)){
                    (*broj_znamenki)++;
                }
                else if(isalpha(trenutni_element)){
                    (*broj_slova)++;
                }
            }
        }
    }
}
```

```

    }
}

int main() {
    char mat[MAXN][MAXN] = {{'1', '2', 'z', '9', 'z'},
                            {'m', '*', 'a', 'a', '6'},
                            {'0', '0', 'b', 'k', '$'},
                            {'3', 'H', '0', 'D', '#'},
                            {'p', 'K', 'g', '!', 'D'}};

    int broj_znamenki, broj_slova;
    statistika_matrice(&mat[0][0], 5, MAXN, &broj_znamenki, &broj_slova);
    printf("Broj znamenki iznad dijagonale je %d, a broj slova je %d",
           broj_znamenki, broj_slova);
    return 0;
}

```

### 3.

```

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

int fun1(void) {
    return rand()%201 - 100;
}

void fun2(float *as, float *sd) {
    int i, x;
    int polje[1000];

    *as = *sd = 0;
    srand(time(NULL));

    for (i=0; i<1000; i++) {
        polje[i] = fun1();
        *as += polje[i];
    }
    *as /= 1000;

    for (i=0; i<1000; i++) {
        *sd += pow(polje[i] - *as, 2);
    }
    *sd = sqrt(*sd/1000);
}

```

### 4.

```

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

int main () {
    FILE *fPredmet, *fAnketa;
    int sifPred, i;
    int ocjena[5] = {0};

    struct predmetZapis {
        long sifPred;
        char nazPred[50+1];
    } predmet;

    struct anketaZapis {
        int sifPred;
        int ocjena[20];
    } anketa;

    fAnketa = fopen ("anketa.bin", "rb");
    fPredmet = fopen ("predmeti.bin", "rb");
}

```

```

printf ("Upisite sifru predmeta: ");
scanf ("%d", &sifPred);
fseek (fPredmet, (long) (sifPred-1) * sizeof (predmet), SEEK_SET);
fread(&predmet, sizeof (predmet), 1, fPredmet);

if (sifPred != predmet.sifPred) {
    printf ("Ne postoji predmet: %d \n", sifPred);
    exit(3);
}

while (fread(&anketa, sizeof (anketa), 1, fAnketa) == 1){
    if (anketa.sifPred == sifPred)
        for (i=0; i<20;i++)
            ocjena[anketa.ocjena[i]-1]++;
}

printf ("Predmet                1      2      3      4      5 \n");
printf ("-----\n");
printf ("%50s %4d %4d %4d %4d %4d \n", predmet.nazPred, ocjena[0], ocjena[1]
        , ocjena[2], ocjena[3], ocjena[4]);

fclose(fAnketa);
fclose(fPredmet);

return 0;
}

```