

JMBAG	
Ime i prezime	

## Programiranje i programsko inženjerstvo

### Završni ispit

30. siječnja 2014.

Odgovore na 1., 2. i 3. a) i 3. b) pitanje napišite na svojim papirima i predajte u košuljici.

Odgovore na 3. c), 4. i 5. pitanje napišite na za to predviđenom mjestu uz zadatak.

#### Zadatak 1. (12 bodova)

U rješenju ovog zadatka nije dozvoljeno koristiti funkcije iz zaglavne (*header*) datoteke `string.h`.

a) Napisati funkciju `pozZnak` koja kao parametre prima znak `znak` i niz znakova `niz`. Funkcija u nizu `niz` pronalazi prvo pojavljivanje znaka `znak` i vraća pokazivač na taj znak u nizu. Pri tome funkcija ne razlikuje mala i velika slova. Ako traženi znak ne postoji u nizu, funkcija vraća `NULL` pokazivač.

Npr. za niz: "Ananas je tropska biljka." i znak "a" funkcija treba vratiti pokazivač na 1. znak niza jer se znak "a" (uz uvjet da ne razlikuje mala i velika slova) prvi put pojavljuje na toj poziciji.

b) Napisati glavni program u kojem se s tipkovnice učitava niz znakova `niz` (ne dulji od 200 znakova) i znak `znak`. Uzastopnim pozivom funkcije iz a) dijela zadatka potrebno je odrediti koliko se puta znak pojavljuje u nizu. Rezultat ispisati na zaslon. Za primjer iz a) dijela zadatka glavnim programom treba ispisati:

U nizu "Ananas je tropska biljka."

znak "a" pojavljuje se 5 puta.

#### Zadatak 2. (12 bodova)

Tekstna datoteka `referendum.txt` smještena u direktoriju `c:\europa` sadrži pojedinačne glasove glasača na europskom referendumu. Svaki redak u datoteci sadrži vrijeme glasanja (niz znakova duljine 5), šifru glasačkog mjesta na kojem je glasač glasao (cijeli broj) i cijeli broj za koji vrijedi:

- ako je vrijednost 1, glasač se izjasnio ZA referendumsko pitanje,
- ako je vrijednost 0, glasač se izjasnio PROTIV referenduskog pitanja,
- bilo koja druga vrijednost predstavlja NEVAŽEĆI listić.

Pretpostaviti da je broj glasačkih mjesta jako velik i nije poznato koliko iznosi. Podaci u pojedinom retku datoteke međusobno su odvojeni znakom #, npr.

08:00#149689#0

pri čemu je 08:00 vrijeme glasanja, 149689 šifra glasačkog mjesta, a 0 ukazuje da je ovaj glasač glasao PROTIV referenduskog pitanja.

Postojeća binarna datoteka `rezGLMjesto.bin` smještena u direktoriju `c:\statistika` predviđena je za pohranu zbirnih rezultata referenduma po glasačkim mjestima. Svaki zapis datoteke `rezGLMjesto.bin` sadrži šifru glasačkog mjesta (int), naziv glasačkog mjesta (znakovni niz duljine 50 znakova) i tri cijela broja (sva tri s početnom vrijednošću 0) koji predstavljaju redom: broj glasova ZA, broj glasova PROTIV i broj nevažećih glasova. Redni broj zapisa u datoteci odgovara šifri glasačkog mjesta. Potrebno je ažurirati sadržaj datoteke `rezGLMjesto.bin` tako da statistika glasanja (broj glasova ZA, broj glasova PROTIV i broj nevažećih glasova) bude u skladu sa sadržajem datoteke `referendum.txt`.

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

### Zadatak 3. (12 bodova)

- a) Napisati funkciju `bacajKocku` koja simulira bacanje kocke zadani broj puta od strane zadanog broja igrača. Jedno bacanje kocke svodi se na generiranje pseudoslučajnog broja iz intervala  $[1,6]$ . Osigurati da funkcija `bacajKocku` inicijalizira generator pseudoslučajnih brojeva samo pri prvom pozivu. Funkcija rezultate bacanja kocke pohranjuje u polje koje je definirano u pozivajućem programu kao dvodimenzijско polje. Jedan redak polja namijenjen je pohrani rezultata bacanja igrača kocke za jednog igrača.
- b) Napisati funkciju `ispisi2Dpolje` koja ispisuje sadržaj polja koje je definirano u pozivajućem programu kao dvodimenzijско polje. Funkcija svaki redak polja treba ispisati u novom retku.
- c) Nadopuniti modul **igra.c** nadopisivanjem po jedne naredbe na svaku crtu. U modulu je potrebno deklarirati funkcije `bacajKocku` i `ispisi2Dpolje`. U glavnom programu je potrebno definirati dvodimenzijско polje maksimalnih dimenzija 10 retka i 8 stupca koje će služiti za pohranu rezultata bacanja kocke. S pomoću funkcija iz a) i b) dijela zadatka potrebno je obaviti sljedeće:
1. ispisati rezultat po 5 bacanja kocke od strane 3 igrača
  2. ispisati rezultat po 8 bacanja kocke od strane 4 igrača

igra.c

```
#include <stdio.h>
/*deklaracije funkcija bacajKocku i ispisi2Dpolje*/

_____  
_____  
  
int main(void) {  
    _____ /*definicija dvodimenzijskog polja: 10 redaka, 8 stupaca*/  
    _____ /*poziv funkcije bacajKocku: */  
    printf("\n 3 igraca po 5 bacanja."); /*3 igrača baca kocku po 5 puta*/  
    _____ /*ispis rezultata - ispisi2Dpolje*/  
    _____ /*poziv funkcije bacajKocku: */  
    printf("\n 4 igraca po 8 bacanja."); /*4 igrača baca kocku po 8 puta*/  
    _____ /*ispis rezultata- ispisi2Dpolje */  
    return 0;  
}
```

### Zadatak 4. (1 bod)

Napisati macro definiciju `ARITM_NIZ` kojom se izračunava suma prvih  $n$  članova aritmetičkog niza za zadane parametre:  $a$  – prvi član niza,  $d$  – diferencija niza,  $n$  – broj članova. Suma prvih  $n$  članova aritmetičkog niza računa se prema formuli:  $n/2 * (2 * a + (n - 1) * d)$ . Naredbu napisati na crtu:

\_\_\_\_\_

### Zadatak 5. (3 boda)

Nadopuniti funkciju `sumaGLISpDijagonale`, tako da funkcija izračuna sumu vrijednosti na glavnoj (`sumGLDij`) i sporednoj (`sumSpDij`) dijagonali cjelobrojne kvadratne matrice `mat` s  $n$  redaka, odnosno stupaca. Na svaku crtu napisati po jednu naredbu.

```
void sumaGLISpDijagonale (int *mat, int maxStup, int n, int *sumaGLDij, int *sumaSpDij) {  
    int i;  
    *sumaGLDij=0;  
    *sumaSpDij=0;  
    for (i=0; i<n; ++i) {  
        _____  
        _____  
    }  
    return;  
}
```

## Rješenja

### Zadatak 1. (12 bodova)

```
a)
#include <ctype.h>
char *pozZnaka(char znak, char *niz) {
    int i = 0;

    while (*(niz + i) != '\0') {
        if (toupper(*(niz + i)) == toupper(znak))
            return (niz + i);
        ++i;
    }
    return NULL;
}

b)
#include <stdio.h>
int main(void){
    char niz[200+1];
    char znak;
    char *pozZnakaUNizu = niz;
    int brZnakovaUNizu = 0;

    printf("\nUpisi niz:\n");
    gets(niz);
    printf("\nUpisi znak:\n");
    scanf("%c", &znak);

    while (pozZnakaUNizu != NULL) {
        pozZnakaUNizu = pozZnaka(znak, pozZnakaUNizu);
        if (pozZnakaUNizu != NULL){
            ++brZnakovaUNizu;
            pozZnakaUNizu += 1;
        }
    }

    printf("\nU nizu \"%s\" \nznak \"%c\" se pojavljuje %d puta.", niz, znak, brZnakovaUNizu);
    return 0;
}
```

### Zadatak 2. (12 bodova)

```
int main(void){
    FILE *fReferendum, *fRezGlmjesto;
    int sifGlmjesto, glas, i, n1;
    char c, vrijeme[5 + 1];

    struct sRezGlmjesto {
        int sifGlmjesto;
        char nazGlmjesto[50+1];
        int za;
        int protiv;
        int nevazecih;
    } rezGlmjestoDat;

    fReferendum = fopen("C:/europa/referendum.txt", "r");
    fRezGlmjesto = fopen("C:/statistika/rezGlmjesto.bin", "r+b");

    while (fscanf(fReferendum, "%5s%c%d%c%d", vrijeme, &c, &sifGlmjesto, &c, &glas) == 5) {
        fseek(fRezGlmjesto, (long)(sifGlmjesto-1)* sizeof (rezGlmjestoDat), SEEK_SET);
        fread(&rezGlmjestoDat, sizeof(rezGlmjestoDat), 1, fRezGlmjesto);

        if (glas == 1) ++rezGlmjestoDat.za;
        else if (glas == 0) ++rezGlmjestoDat.protiv;
        else ++rezGlmjestoDat.nevazecih;

        fseek(fRezGlmjesto, (long)(-1) * sizeof (rezGlmjestoDat), SEEK_CUR);
        fwrite(&rezGlmjestoDat, sizeof(rezGlmjestoDat), 1, fRezGlmjesto);
    }

    fclose(fReferendum);
    fclose(fRezGlmjesto);
    return 0;
}
```

### Zadatak 3. (12 bodova)

a)

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

void bacaJKocku(int brIgraca, int brBacanjaPoIgracu, int maxBacanja, int *rezBacanja){

    int i, j;
    static int generatorInicijaliziran = 0;

    if (!generatorInicijaliziran) {
        srand((unsigned)time(NULL));
        generatorInicijaliziran = 1;
    }

    for (i = 0; i < brIgraca; ++i)
        for (j = 0; j < brBacanjaPoIgracu; ++j)
            *(rezBacanja + i*maxBacanja + j) = rand() % 6 + 1;

    return;
}

b)
void ispisi2dPolje(int brRedaka, int brStupaca, int maxBrStupaca, int *rezBacanja){
    int i, j;

    for (i = 0; i < brRedaka; ++i) {
        printf("\n");
        for (j = 0; j < brStupaca; ++j)
            printf("%3d", *(rezBacanja + i*maxBrStupaca + j));
    }
    return;
}

c)
```

```
#include <stdio.h>

void bacaJKocku(int brIgraca, int brBacanjaPoIgracu, int maxBacanja, int *rezBacanja);
void ispisi2dPolje(int brRedaka, int brStupaca, int maxBrStupaca, int *rezBacanja);

int main(void) {
    int rezultatibacanja[10][8];
    bacaJKocku(3, 5, 8, &rezultatibacanja[0][0]);
    printf("\n 3 igrača po 5 bacanja.");
    ispisi2dPolje(3, 5, 8, &rezultatibacanja[0][0]);

    bacaJKocku(4, 8, 8, &rezultatibacanja[0][0]);
    printf("\n 4 igrača po 8 bacanja.");
    ispisi2dPolje(4, 8, 8, &rezultatibacanja[0][0]);
    return 0;
}
```

igra.c

### Zadatak 4. (1 bod)

```
#define ARITM_NIZ(a,d,n) ((n)/2. *(2*(a)+((n)-1)*(d)))
```

### Zadatak 5. (3 boda)

```
*sumGlDij += *(mat+i*maxStup+i);
*sumSpDij += *(mat+i*maxStup+n-1-i);
```