

# Programiranje i programsko inženjerstvo

## 1. ispitni rok

13. veljače 2012.

**Nije dopušteno korištenje statičkih i eksternih varijabli i naredbe goto.**

### Zadatak 1. (15 bodova)

- a) Napisati funkciju `ispis_rijeci` koja kao formalne argumente prima niz znakova i tok podataka za kojeg se pretpostavlja da je prije poziva funkcije uspješno otvoren za čitanje slijedne formatirane datoteke i pozicioniran na početak datoteke. Datoteka u svakom retku sadrži riječ (ne dulju od 50 znakova). Funkcija na zaslon, svaku u svojem retku, ispisuje sve riječi iz datoteke koje sadrže zadani niz.
- b) Napisati glavni program koji će s tipkovnice učitati riječ (ne dulju od 50 znakova) te pomoću funkcije `ispis_rijeci` na zaslon ispisati sve riječi iz slijedne formatirane datoteke `baza_rijeci.txt` koje sadrže učitano riječ.

### Zadatak 2. (15 bodova)

- a) Napisati funkciju sa sljedećim prototipom:

```
int igra_RPS(int korisnik, int *racunalo);
```

Funkcija vraća rezultat jednog kruga igre kamen-papir-škare (eng. rock-paper-scissors) u kojoj korisnik igra protiv računala. Pri svakom pozivu funkcija generira pseudoslučajni broj (1, 2 ili 3) koji predstavlja „odabir“ protivnika, tj. računala i na temelju te vrijednosti i korisnikove vrijednosti predane parametrom `korisnik` (1 - kamen, 2 - papir, 3 - škare), određuje ishod jednog kruga igre (treba vratiti 1 ako je korisnik pobijedio računalo, -1 ako je izgubio, i 0 ako je rezultat izjednačen). Odabir računala funkcija vraća parametrom `racunalo`.

Pravila igre su: Kamen pobjeđuje škare, škare pobjeđuju papir, a papir pobjeđuje kamen.

- b) Napisati glavni program koji simulira 10 krugova igre. U svakom krugu igre korisnik unosi svoj odabir (1, 2, 3 ili 0 za prekid), a zatim se poziva funkcija `igra_RPS` koja odlučuje o pobjedniku tog kruga. U svakom krugu ponavljati unos sve dok učitani broj ne bude ispravan (u intervalu 0 do 3). Nakon svakog kruga igre potrebno je na zaslon ispisati odabir korisnika, odabir računala i ishod kruga. Nakon svih krugova ispisuje se konačan rezultat (1 ako je korisnik pobijedio računalo, -1 ako je izgubio ili 0 ako je rezultat izjednačen).

### Zadatak 3. (15 bodova)

Zadana je neformatirana datoteka **artikli.bin** s podacima o artiklima na skladištu. Svaki zapis sadrži šifru artikla (int), naziv artikla (50+1 znak) i količinu dostupnu na skladištu (int). Datoteka ne sadrži prazne zapise. Zadana je i slijedna formatirana datoteka **artikli.txt** čiji svaki redak sadrži šifru artikla (cijeli broj), naziv artikla i količinu dostupnu na skladištu (cijeli broj) odvojene znakom točka-zarez (;). Zapisi u slijednoj datoteci su poredani uzlazno po šifri artikla. Napisati program koji uspoređuje sadržaj datoteka `artikli.bin` i `artikli.txt` i ispisuje jesu li datoteke jednake ili ne. Smatra se da su datoteke jednake ako sadrže sve pripadne jednake zapise (jednake vrijednosti šifre, naziva i količine) poredane jednakim redoslijedom. Potrebno je provjeriti i uspješnost otvaranja datoteka, te ako otvaranje nije uspješno ispisati odgovarajuću poruku i završiti izvođenje programa.

#### Zadatak 4. (15 bodova)

- a) Napisati funkciju `najdulji_niz` koja će u zadanom znakovnom nizu pronaći najdulji niz u kojem se uzastopce ponavlja zadani znak. Funkcija treba vratiti broj uzastopnih ponavljanja najduljeg neprekinutog niza i poziciju prvog znaka u nizu. Ako u nizu ne postoji traženi znak, funkcija vraća duljinu 0 i poziciju -1, a ako ima više nizova iste duljine, vraća onaj sa manjim indeksom u nizu.
- b) Napisati glavni program u kojem se s tipkovnice učitava niz (ne dulji od 50 znakova) i znak, pozove funkciju `najdulji_niz`, te na zaslon ispiše duljinu i poziciju početka niza ili odgovarajuća poruka ako zadani znak ne postoji u nizu.

Npr. Za niz `abbccbbccc` i znak `b` funkcija treba vratiti duljinu 2 i poziciju 1. Za isti niz i za znak `c` funkcija vraća duljinu 3 i poziciju 7.

#### Zadatak 5. (20 bodova)

- a) Napisati funkciju `pucaj` koja kao parametar prima kvadratnu znakovnu matricu reda  $n$ . Matrica predstavlja konfiguraciju "bojnog polja" u pojednostavljenoj igri potapanja brodova, i to tako da `~` predstavlja znak "voda", `O` dio broda, a `#` potopljenog broda. Funkcija kao parametar prima i dvije cjelobrojne vrijednosti (redak i stupac) koje predstavljaju "gađano polje". Ako se na "gađanom polju" nalazi brod, funkcija treba SVE znakove "O" koji predstavljaju pogođeni brod zamijeniti sa "#" (brod se potapa jednim pogotkom) i vratiti veličinu broda (tj. broj promijenjenih znakova iz "O" u "#"), a inače nulu. Svi su "brodovi" postavljeni vodoravno i ne protežu se na više od jednog retka.
- b) Napišite glavni program koji iz direktne neformatirane datoteke učitava polje znakova, s tipkovnice učitava koordinate „gađanog polja“, poziva funkciju `pucaj`, te u istu datoteku sprema novo polje znakova. Polje znakova je u datoteci spremljeno na sljedeći način: u prvom retku datoteke zapisan je broj  $n$  (`short`) koji predstavlja broj redaka odnosno stupaca matrice, nakon čega slijedi  $n \times n$  znakova (elemenata matrice). Matrica sigurno nije veća od  $150 \times 150$  elemenata.

Npr. Za matricu

~	~	~	~	O	O
~	~	~	~	~	~
~	~	#	#	#	~
O	~	~	~	~	~
~	~	O	O	O	O
~	~	~	~	~	~

i koordinate (4,3) rezultat je

~	~	~	~	O	O
~	~	~	~	~	~
~	~	#	#	#	~
O	~	~	~	~	~
~	~	#	#	#	#
~	~	~	~	~	~

i povratna vrijednost 4.

## Rješenja:

### Zadatak 1.

a)

```
#include <stdio.h>
#include <string.h>
#define MAX 50

void ispis_rijeci(char *trazena_rijec, FILE *datoteka){
    char rijec[MAX+1];
    while (fscanf(datoteka, "%s", rijec) == 1) {
        if (strstr(rijec, trazena_rijec) != NULL)
            printf("%s\n", rijec);
    }
}
```

b)

```
#include <stdio.h>
#define MAX 50

int main(){
    char trazena_rijec[MAX+1];
    FILE *dat;

    dat = fopen("baza_rijeci.txt", "r");

    printf("Unesite rijec:");
    scanf("%s", trazena_rijec);
    printf("Nadjene rijeci:\n");
    ispis_rijeci(trazena_rijec, dat);

    fclose(dat);
    return 0;
}
```

## Zadatak 2

a)

```
#include <stdlib.h>

int igra_RPS(int korisnik, int *racunalo){
    *racunalo = rand() % 3 + 1;
    if (korisnik == *racunalo) return 0;
    else if ((korisnik == 1 && *racunalo == 3)
        || (korisnik == 2 && *racunalo == 1)
        || (korisnik == 3 && *racunalo == 2))
        return 1;
    else
        return -1;
}
```

b)

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

int main() {
    int i, kor, rac, rez, ukRez=0;
    srand((unsigned)time(NULL));
    printf ("Igra Rock - Paper - Scissors \n1=Rock 2=Paper 3=Scissors 0=Kraj\n");

    for (i=0; i<10; i++) {
        do {
            printf("\n[%d. krug] [1,2,3,0]: ", i+1);
            scanf ("%d", &kor);
        } while (kor<0 || kor>3);

        if (kor==0) break;

        rez = igra_RPS(kor, &rac);
        ukRez += rez;

        printf ("Korisnik: %d, Racunalo: %d, Pobjednik: %s\n"
            , kor, rac, (rez==1)?"Korisnik":(rez==-1)?"Racunalo":"Izjednaceno");
    }

    if (ukRez > 0) printf("1\n");
    else if (ukRez < 0) printf("-1\n");
    else printf ("0\n");

    return 0;
}
```

## Zadatak 3

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

void main() {
    FILE *f1,*f2;
    struct{
        int sifra;
        char naziv[50+1];
        int kolicina;
    }z;

    int sifra, kolicina, jednake = 1;
    char naziv[50+1], c;
    int brprocl, brproc2;

    f1 = fopen("artikli.bin", "rb");
    if (f1 == NULL) {
        printf("Doslo je do pogreske prilikom učitavanja datoteke artikli.bin");
        exit(-1);
    }
    f2 = fopen("artikli.txt", "r");
    if (f2 == NULL) {
        printf("Doslo je do pogreske prilikom učitavanja datoteke artikli.txt");
        exit(-1);
    }

    do{
        brprocl = fread(&z, sizeof(z), 1, f1);
        brproc2 = fscanf(f2, "%d%c%[^;]%c%d", &sifra, &c, naziv, &c, &kolicina);
        if (brprocl == 1 && brproc2 == 5){
            if (z.sifra != sifra || strcmp(z.naziv, naziv) != 0 || z.kolicina != kolicina){
                jednake = 0;
                break;
            }
        }
    } while (brprocl == 1 && brproc2 == 5);

    if (jednake && brprocl == 0 && brproc2 == -1)
        printf ("Datoteke su jednake");
    else
        printf ("Datoteke nisu jednake");
    return 0;
}
```

## Zadatak 4

a)

```
#include <string.h>

void najdulji_niz(char *niz, char znak, int *maxPoz, int *maxDulj) {

    char *kazaljka, *pocNovogNiza;

    *maxPoz = -1;
    *maxDulj = 0;
    kazaljka = niz;

    while (*kazaljka != '\0') {
        if (*kazaljka == znak) {
            pocNovogNiza = kazaljka;
            while (*kazaljka == znak)
                kazaljka++;
            if (kazaljka - pocNovogNiza > *maxDulj) {
                *maxPoz = pocNovogNiza - niz;
                *maxDulj = kazaljka - pocNovogNiza;
            }
        }
        else
            kazaljka++;
    }
}
```

b)

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX 50

int main(){
    char niz[MAX + 1];
    char znak;
    int poz, d;

    printf("Unesite niz: ");
    gets(niz);
    printf("Unesite znak: ");
    scanf("%c", &znak);

    najdulji_niz(niz, znak, &poz, &d);

    if (d > 0) printf("Najduzi niz %d znakova, počinje na poziciji %d.\n", d, poz);
    else printf ("Nema niti jednog znaka %c.\n", znak);

    return 0;
}
```

## Zadatak 5

a)

```
int pucaj(char *mat, int n, int maxstup, int koord_x, int koord_y){
    int br=0, j;

    if (mat[koord_x * maxstup + koord_y]!='~' || mat[koord_x * maxstup + koord_y]!='#')
        return 0;

    mat[koord_x * maxstup + koord_y]='#';
    br++;

    /* lijevo */
    j = koord_y -1;
    while (j > 0) {
        if (mat[koord_x * maxstup + j] == '~') break;
        mat[koord_x * maxstup + j] = '#';
        br++;
        j--;
    }

    /* desno */
    j = koord_y +1;
    while (j < n) {
        if (mat[koord_x * maxstup + j] == '~') break;
        mat[koord_x * maxstup + j] = '#';
        br++;
        j++;
    }

    return br;
}
```

b)

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 150

int main()
{
    FILE *f;
    short n;
    char mat[MAX][MAX];
    int i, j, x, y, ret;

    f = fopen("brodovi.bin", "r+b");

    fread(&n, sizeof(short), 1, f);

    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            fread(&mat[i][j], sizeof(char), 1, f);

    printf("Unesite koordinate koje gadjate: ");
    scanf("%d %d", &x, &y);

    ret = pucaj(&mat[0][0], n, MAX, x, y);

    fseek(f, 0, SEEK_SET);
    fwrite(&n, sizeof(short), 1, f);
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            fwrite(&mat[i][j], sizeof(char), 1, f);

    fclose(f);
    return 0;
}
```