

Programiranje i programsko inženjerstvo

Ispitni rok

14. veljače 2013.

Nije dopušteno korištenje statičkih i eksternih varijabli i naredbe goto.

Zadatak 1. (23 boda)

Tekstualna datoteka **ispitniRok1.txt** sadrži zapise o rezultatima studenata na ispitu iz *Programiranja i programskog inženjerstva*. Svaki se redak u datoteci sastoji od šifre studenta (cijeli broj) te bodova iz 5 zadataka (realni brojevi s jednom decimalom). Podatci u pojedinom retku datoteke su međusobno odvojeni jednom prazninom, npr.

1234 1.0 10.5 2.5 13.5 15.0

gdje je 1234 šifra studenta, 1.0 broj bodova koje je student ostvario u 1. zadatku, 10.5 broj bodova koje je student ostvario u 2. zadatku, itd.

U direktnoj binarnoj datoteci **studentiProg.bin** nalaze se zapisi o studentima koji su upisali predmet *Programiranje i programsko inženjerstvo*. Zapis se sastoji od šifre studenta (cijeli broj), imena studenta (najviše 30 znakova), prezimena studenta (najviše 50 znakova) i ocjene (cijeli broj). Redni broj zapisa u datoteci odgovara šifri studenta.

Napisati program koji će u datoteci **studentiProg.bin** ažurirati ocjene za studente koji su na ispitu dobili ocjenu veću od 1. Ocjena se određuje pozivanjem postojeće (nije ju potrebno pisati) funkcije `izracunajOcjenu` koja kao parametar prima broj zadataka i polje s bodovima po zadacima a vraća ocjenu. Prototip funkcije je:

```
int izracunajOcjenu(int brojZadataka, float bodovi[]);
```

Napomena: nije potrebno provjeravati uspješnost otvaranja datoteka te uspješnost pozicioniranja u datoteci.

Zadatak 2. (23 boda)

Igra se sastoji od izvlačenja brojeva iz intervala [1, 20]. U igri sudjeluje 20 igrača. Svaki igrač na početku ima 0 bodova. Ovisno o izvučenom broju, jedan od igrača dobiva bod i to na sljedeći način: ako je izvučen broj 1, onda prvi igrač dobiva 1 bod, ako je izvučen broj 2, onda drugi igrač dobiva 1 bod, itd. Igra završava nakon unaprijed zadanog broja izvlačenja. Pobjednik je igrač koji je prikupio najviše bodova. U slučaju da dva ili više igrača ostvare jednak najveći broj bodova, pobjednik je onaj igrač koji je prvi došao do tog broja bodova.

- Napisati funkciju `igra` koja predstavlja opisanu igru. Funkcija simulira izvlačenje brojeva unaprijed zadani broj puta, izračunava broj bodova za svakog igrača i određuje pobjednika. Funkcija mora vratiti redni broj pobjednika (ako je pobijedio prvi igrač funkcija treba vratiti 1, itd.) te bodove svih igrača ostvareni u toj igri.
- Napisati glavni program u kojem će se s tipkovnice učitati broj igara (n) i broj izvlačenja u jednoj igri (m). Funkciju `igra` potrebno je pozvati n puta. Nakon svakog poziva funkcije na zaslon ispisati redni broj igrača koji je pobijedio u igri, a zatim 20 brojeva koji predstavljaju u igri ostvareni broj bodova svakog pojedinog igrača (svi su brojevi međusobno odijeljeni prazninama), npr. za $n = 2$ i $m = 200$ ispisuje se:

```
7 6 12 6 6 13 6 15 8 11 12 12 12 12 13 10 10 10 5 12 9
8 6 6 10 9 9 7 8 18 11 15 14 12 9 5 18 6 6 16 6 9
```

Napomena: nije potrebno provjeravati uspješnost otvaranja datoteke.

Zadatak 3. (18 bodova)

Puni naziv datoteke je niz znakova proizvoljne duljine koji može sadržavati jednu ili više točaka (znak '.'). Ako naziv sadrži točku (ili više točaka) tada se ekstenzijom naziva podniz nakon posljednje točke u nazivu. Npr. datoteka imena **super.tajni.dokument.txt** je datoteka s ekstenzijom **txt**, a datoteka **naziv.dostaDugeEkstenzija** je datoteka s ekstenzijom **dostaDugeEkstenzija**.

- Napisati funkciju `odrediExt` koja pronalazi ekstenziju datoteke iz njezina punog naziva. Funkcija mora biti napisana tako da ispravno pronalazi ekstenziju datoteke bilo koje duljine. Duljina ekstenzije također nije ograničena. Funkcija ne smije promijeniti naziv datoteke.
- Napisati glavni program u kojem će se s tipkovnice učitati naziv datoteke, sigurno ne dulji od 100 znakova, pozvati funkciju `odrediExt`, te na osnovi rezultata funkcije na zaslon ispisati ili dobivenu ekstenziju imena datoteke ili poruku "*Datoteka nema ekstenziju*".

Zadatak 4. (6 bodova)

Kolika je vrijednost pohranjena u varijablu tipa *double* ako je njen binarni sadržaj:

a) 00000000 00010000 00000000 00000000 00000000 00000000 00000000 10000001₂

Rješenje: _____

b) 01111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000₂

Rješenje: _____

c) 01111111 11111000 00000000 00000000 00000000 00000000 00000000 00000000₂

Rješenje: _____

Zadatak 5. (4 boda)

Nadopuniti funkciju `zamijeniZnamenkom`, tako da se u zadanom nizu `niz` svako pojavljivanje znaka `z` zamjenjuje slučajnom dekadskom znamenkom.

Pretpostaviti da će generator slučajnih brojeva biti inicijaliziran prije poziva funkcije `zamijeniZnamenkom`.

Nije dopušteno koristiti dodatne programske petlje.

```
#include <string.h>
#include <stdlib.h>
...
int zamijeniZnamenkom (char *niz, char z) {
    while ( _____ ) {

        *niz = _____;
    }
    return 0;
}
```

Zadatak 6. (6 bodova)

Funkcija `fibonacci` zadanog prototipa (deklaracije) vraća *n*-ti član Fibonaccijevog niza. Nadopunite funkciju `prvihN` tako da se pozivajući funkciju `fibonacci` ispisuje prvih *n* elemenata Fibonaccijevog niza obrnutim redoslijedom u sljedećem formatu (primjer za argument 10):

```
fibonacci(00010)= 55
```

```
fibonacci(00009)= 34
```

```
...
```

```
fibonacci(00002)= 1
```

```
fibonacci(00001)= 1
```

Nije dopušteno koristiti dodatne varijable i programske petlje.

```
#include <string.h>
...
long fibonacci(int x);
...
void prvihN(int n) {
    do {
        printf("fibonacci( _____ ) = %d\n", _____ , _____ );
        _____ ;
    } while( _____ );
}
```

Rješenja:

Zadatak 1.

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

typedef struct s {
    int sifStudent;
    char imeStudent[30 + 1];
    char prezStudent[50 + 1];
    int ocjena;
} zapis;

int main() {
    FILE *fIspitRok = NULL, *fStudent = NULL;
    float bodovi[BRZAD] = {0.};
    int sifStudent;
    zapis student;
    int ocjena, i = 0;

    fIspitRok = fopen("ispitniRok1.txt", "r");
    fStudent = fopen("studentiProg.bin", "r+b");

    // učitavanje zapisa o ispitnom roku
    while (fscanf(fIspitRok, "%d %f %f %f %f %f", &sifStudent, &bodovi[0], &bodovi[1], &bodovi[2],
        &bodovi[3], &bodovi[4]) == 6) {
        ocjena = izracunajOcjenu(BRZAD, bodovi);
        if(ocjena > 1) {
            fseek(fStudent, (sifStudent - 1) * sizeof(zapis), SEEK_SET);
            fread(&student, sizeof(zapis), 1, fStudent);
            student.ocjena = ocjena;
            fseek(fStudent, sizeof(zapis) * (-1L), SEEK_CUR);
            fwrite(&student, sizeof(zapis), 1, fStudent);
        }
    }

    fclose(fIspitRok);
    fclose(fStudent);
    return 0;
}
```

Zadatak 2.

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

void igra(int m, int *pobjednik, int *bodovi) {
    int i, izvucen;
    *pobjednik = 1;
    for (i = 0; i < 20; i++) {
        bodovi[i] = 0;
    }
    for (i = 0; i < m; i++) {
        izvucen = rand() % 20 + 1;
        ++bodovi[izvucen - 1];
        if (bodovi[izvucen - 1] > bodovi[*pobjednik - 1]) {
            *pobjednik = izvucen;
        }
    }
}

int main() {
    int n, m;
    int i, j;
    int bodovi[20], pobjednik;

    scanf("%d %d", &n, &m);
    srand((unsigned)time(NULL));
    for (i = 0; i < n; i++) {
        igra(m, &pobjednik, bodovi);
        printf("%d", pobjednik);
        for (j = 0; j < 20; j++) {
            printf(" %d", bodovi[j]);
        }
        printf("\n");
    }
    return 0;
}
```

Zadatak 3.

```
#include <stdio.h>

char *odrediExt(char *ime) {
    char *pom = NULL;

    if(ime == NULL)
        return NULL;
    do {
        if(*ime == '.')
            pom = ime + 1;
        ime++;
    } while(*ime != '\0');
    return pom;
}

int main() {
    char ime[ 100 + 1 ];
    char *ext;
    gets(ime);
    ext = odrediExt(ime);
    if(ext != NULL) {
        printf("Ekstenzija je %s\n", ext);
    } else {
        printf("Datoteka nema ekstenziju");
    }
    return 0;
}

/*alternativa:
void odrediExt(char *ime, char ekst[]);
int main() {
    ...
    char ekstenzija[100];
    ...
    odrediExt(ime, ekstenzija);
    ...
}
*/
```

Zadatak 4.

- a) $2^{-1022} + 2^{-(1022+45)} + 2^{-(1022+52)} \approx 2^{-1022}$
- b) $+\infty$
- c) NaN

Zadatak 5.

```
int zamijeniZnamenkom (char *niz, char z) {
    while (niz = strchr(niz, z)) {
        *niz = rand() % 10 + '0';
    }
    return 0;
}
```

Zadatak 6.

```
do {
    printf( "fibonacci(   %05d   ) = %d\n",   n   ,   fibonacci(n)   );
      --n  ;
} while(   n > 0   );
```