

JMBAG:		Ime i prezime:	
--------	--	----------------	--

Programiranje i programsko inženjerstvo

Ljetni ispitni rok 2. srpnja 2015.

Odgovore na 1., 2. i 3. pitanje napišite na svojim papirima, a na 4., 5. i 6. pitanje na listu s zadacima.

Zadatak 1. (20 bodova)

U binarnoj datoteci su pohranjeni zapisi o osobama i njihovom zdravstvenom osiguranju definirani strukturom:

```
struct osoba {
    char oib[12];
    char ime_prezime[51];
    int osigurana; /* 1 ako je osoba osigurana, 0 inace */
    char mat_broj_osigurane_osobe[10];
};
```

Također, zadana je funkcija

```
int treba_na_pregled(char *broj_osiguranika);
```

(nije potrebno implementirati), koja za osiguranika zadanog matičnim brojem osigurane osobe pri osiguravatelju vraća vrijednost 1 ako osiguranik treba pristupiti pregledu, odnosno 0 ako ne treba.

Potrebno je napisati funkciju

```
void sastavi_popis(char *dat_osobe, char *dat_popis, int kapacitet_klinike);
```

koja će kreirati novu slijednu tekstualnu datoteku s popisom OIB-a osoba koje su osigurane i trebaju pristupiti pregledu. Funkcija prima ime ulazne binarne datoteke s osobama, ime nove datoteke (pretpostaviti da datoteka ne postoji i da ju je potrebno stvoriti) te broj osoba koje je moguće primiti na pregled (kapacitet klinike). Zapisi (OIB-i) u izlaznoj datoteci su navedeni po jedan u svakom retku. Ako je u ulaznoj datoteci više od kapacitet_klinike osoba koje trebaju na pregled, u izlaznu datoteku treba upisati samo prvih kapacitet_klinike OIB-a.

Zadatak 2. (15 bodova)

Napisati funkciju `ispisi_max_srednjih` koja za zadano cjelobrojno dvodimenzijско polje na zaslon ispisuje za svaki stupac najveću srednju vrijednosti svakog para susjednih elemenata u tom stupcu. Stupac od N elemenata ima $(N-1)$ parova susjednih elemenata (prvi i drugi, drugi i treći, ..., predzadnji i zadnji).

Svaku vrijednost ispisati u jednom retku tako da budu desno poravnate s jednom znamenkom iza decimalne točke.

	2	13	11	2		6.0
Primjer: za zadano polje	-4	6	2	5	na zaslon je potrebno ispisati:	9.5
	16	12	10	17		6.5
						11.0

jer je najveća srednja vrijednost susjednih elemenata u prvom stupcu $\frac{-4+16}{2}$, u drugom $\frac{13+6}{2}$, a u trećem $\frac{11+2}{2}$.

Zadatak 3. (20 bodova)

Napisati funkciju koja generira nove riječi. Zadan je prototip funkcije:

```
void generiraj_rijec(char *rijec, char *novarijec)
```

Funkcija generira novu riječ i vraća ju preko argumenta `novarijec`. Nova riječ se sastoji od slučajno odabranih znakova iz niza `rijec`. Duljina nove riječi je jednaka duljini ulazne riječi. U novoj riječi moraju se naizmjenično pojavljivati suglasnici i samoglasnici, počevši sa suglasnikom. Isti samoglasnik ili suglasnik mogu se pojaviti više puta u novoj riječi.

Npr. za riječ "daaa" generirana nova riječ je "dada", a za riječ "rijec" generirane nove riječi mogu biti: "jicij", "rejij", "cerij", "ririj"...

Pretpostaviti da lokacija u memoriji na koju pokazuje `novarijec` ima dovoljno mjesta za pohranu nove riječi i da ulazna riječ sadrži samo slova i to i samoglasnike i suglasnike.

Zadatak 4. (5 bodova)

- a) Napisati dekadski zapis broja čiji je oktalni zapis u IEEE754 formatu jednostruke preciznosti zadan s:
37777777767₍₈₎

Rješenje: _____

- b) Napisati sadržaj registra u kojem je, prema standardu IEEE 754 za prikaz brojeva u dvostrukoj preciznosti, pohranjen broj 101.101₍₈₎. Sadržaj registra napisati u heksadekadskom obliku.

Rješenje: _____

Zadatak 5. (10 bodova)

Nadopuniti funkciju `provjeri_palindrom` koja za parametar `rijec` vraća vrijednost 1 ako je riječ palindrom, inače 0, pri čemu ne razlikujemo mala i velika slova

Primjeri palindroma: radar, Rotor, dud, kisik, PoTOp

```
#include <ctype.h>
```

```
...
```

```
int provjeri_palindrom(char * rijec){
```

```
    int i, duljina, palindrom;
```

```
    palindrom = 1;
```

```
    duljina = _____;
```

```
    for (i = 0; i < duljina / 2; i++){
```

```
        if (_____){
```

```
            palindrom = 0;
```

```
        }
```

```
    }
```

```
    return palindrom;
```

```
}
```

Zadatak 6. (10 bodova)

U prostoru označenom pravokutnikom napišite što će se ispisati na ekranu izvođenjem sljedećeg programskog odsječka:

```
char polje[4] = { 10, '7', 0x0A  };  
printf("%d %d %c", *(polje+1)+1, polje[2], polje[1] + 1);
```

Rješenja

Zadatak 1

```
void sastavi_popis( char *dat_student, char *dat_popis, int kapacitet_klinike ) {
    FILE *fi, *fo;
    int i = 0, prvi = 1;
    struct student s;
    fi = fopen( dat_student, "rb" ) ;
    fo = fopen( dat_popis, "w" ) ;
    while( ( i < kapacitet_klinike ) && fread( &s, sizeof( s ), 1, fi ) ) {
        if( s.osigurana && treba_na_pregled( s.mat_broj_osigurane_osobe ) ) {
            if( !prvi ) {
                fprintf( fo, "\n" );
            }
            prvi = 0;
            i++;
            fprintf( fo, "%s", s.jmbag );
        }
    }
    fclose( fo );
    fclose( fi );
}
```

Zadatak 2

```
void ispis_max_srednjih( int *mat, int r, int s, int max_stu) {
    int i, j;
    float trenmax, trenavg;
    for( i = 0; i < s; i ++ ) {
        for( j = 0; j < r - 1; j++ ) {
            trenavg = ( mat[j * max_stu + i] + mat[ (j + 1) * max_stu + i] ) / 2.;
            if( j == 0 || trenavg > trenmax )
                trenmax = trenavg;
        }
        printf( "%8.1f\n", trenmax );
    }
}
```

Zadatak 3

```
void generiraj_rijec(char *rijec, char *novarijec){
    int duljina, i, index, samoglasnik;
    char slovo;

    i = index = samoglasnik = 0;
    duljina = strlen(rijec);

    for (i = 0; i < duljina; i++){
        do {
            index = rand() % duljina;
            slovo = tolower(*(rijec + index));

            samoglasnik = 0;
            if (slovo == 'a' || slovo == 'e' || slovo == 'i' || slovo == 'o' || slovo == 'u'){
                samoglasnik = 1;
            }
        } while ((i % 2 == 0 && samoglasnik) || (i % 2 == 1 && !samoglasnik));

        *(novarijec + i) = slovo;
    }

    *(novarijec + i) = '\0';
}
```

Zadatak 4

- a) NaN
- b) 4050482000₍₁₆₎

Zadatak 5

```
int provjeri_palindrom(char * rijec){
    int i, duljina, palindrom;

    palindrom = 1;
    duljina = strlen(rijec);

    for (i = 0; i < duljina / 2; i++){
        if (toupper(rijec[i]) != toupper(rijec[duljina - i - 1])) {
            palindrom = 0;
            break;
        }
    }

    return palindrom;
}
```

Zadatak 6

56 10 8