

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

Programiranje i programsko inženjerstvo

Dekanski ispitni rok 16. rujna 2015.

Odgovore na 1., 2., 3. pitanje napišite na svojim papirima, a na 4., 5. i 6. pitanje na listu s zadacima. Ispit nosi 80 bodova, za prolazak je potrebno ostvariti 40.

Zadatak 1. (20 bodova)

U direktnoj binarnoj datoteci `igraci.dat` pohranjeni su podatci o igračima tenisa: šifra igrača (cijeli broj), ime i prezime igrača (70 + 1 znak). Redni broj zapisa u datoteci odgovara šifri igrača.

U tekstualnoj datoteci `rezultati.txt` pohranjeni su podatci o rezultatima teniskih mečeva. Svaki redak datoteke sadrži zapis o jednome meču u obliku:

```
123 7089 3 2 01 02 2015
4045 123 2 0 11 10 2015
257 123 0 3 06 07 2015
```

...

Podatci u retku razdvojeni su bjelinom, a redom predstavljaju šifru prvog igrača (cijeli broj), šifru drugog igrača (cijeli broj), broj setova koje je osvojio prvi igrač (cijeli broj), broj setova koje je osvojio drugi igrač (cijeli broj) te datum (znakovni niz oblika *dd mm gggg*).

Potrebno je napisati funkciju u kojoj će se za zadanu šifru igrača (cijeli broj) i kalendarsku godinu (cijeli broj) ispisati izvješće o mečevima koje je igrač sa zadanom šifrom odigrao u zadanoj godini. Pretpostaviti da za zadanu šifru igrača i godinu postoji barem jedan zapis u datoteci `rezultati.txt`. Izvješće treba ispisati u sljedećem obliku (primjer je naveden kada je za šifru igrača zadano 123, a za godinu 2015, a koriste se podatci iz gornjeg primjera za datoteku `rezultati.txt`):

```
Protivnik          | Rezultat
-----
7089 John Smith-Smith | 3:2
4045 Ivo Ivic        | 0:2
257 Jacques Martin   | 3:0
```

Npr. igrač sa šifrom 123 pobijedio je Johna Smith-Smitha (čija je šifra zapisa 7089) rezultatom 3:2 u setovima, a izgubio je od Ive Ivica (čija je šifra zapisa 4045) rezultatom 0:2 u setovima te je pobijedio igrača Martina Jacquesa rezultatom 3:0 u setovima.

Zadatak 2. (16 bodova)

Napisati funkciju čiji je prototip:

```
char *kombiniraj (char *novi, char *prvi, char *drugi, int n);
```

u kojoj se u niz `novi` dodaju naizmjenično po `n` znakova iz nizova `prvi` i `drugi` sve dok se u `novi` ne dodaju svi znakovi iz nizova `prvi` i `drugi` (u posljednjem dodavanju znakova iz nizova `prvi` i `drugi` može biti dodano manje od `n` znakova).

Pretpostaviti da je niz `novi` dovoljno velik da u njega stanu svi znakovi iz nizova `prvi` i `drugi`. Funkcija treba vratiti pokazivač na početni znak niza `novi`.

Primjer: ako je niz `prvi` "ABCD EFGHIJ", a niz `drugi` "abcd" i `n` je 3, onda niz `novi` treba biti "ABCabcD EdFGHIJ".

Zadatak 3. (18 bodova)

Napišite funkciju `prosiri` čiji je prototip:

```
int prosiri(long *mat, int r, int s, int novir, int novis, int maxr, int maxs);
```

Funkcija kao argument prima matricu dimenzija $r \times s$ (za koje vrijedi $r \leq \text{maxr}$ i $s \leq \text{maxs}$), a proširuje je na nove dimenzije $\text{novir} \times \text{novis}$. Maksimalni broj redaka i stupaca matrice zadani su argumentima `maxr` i `maxs`. Matrica se proširuje generiranjem slučajnih brojeva iz intervala $[\text{min}, \text{max}]$ gdje su `min` i `max` najmanji i najveći broj u originalnoj matrici. Ako je nemoguće proširiti matricu na zadane dimenzije jer je $\text{novir} > \text{maxr}$ ili $\text{novis} > \text{maxs}$, funkcija treba vratiti vrijednost koja predstavlja logičku laž, a inače vraća vrijednost koja predstavlja logičku istinu. Napomena: u funkciji je potrebno inicijalizirati generator slučajnih brojeva, ali samo kada je pozvana prvi puta tijekom izvođenja programa.

Primjer mogućeg proširenja matrice dimenzija 4×3 na 4×4 (novododane vrijednosti su iz intervala $[-1, 4]$, `min` i `max` elementi matrice su potcrtani):

<u>4</u>	<u>4</u>	<u>-1</u>
-1	2	3
1	-1	3
2	0	2

4	4	-1	3
-1	2	3	1
1	-1	3	-1
2	0	2	4

Zadatak 4. (10 bodova)

Koristeći prostor nad crtama, nadopunite dijelove programa kako bi funkcija `zamijeni` zamijenila prvih `n` znakova nizova `a` i `b`. Funkcija vraća logičku istinu, ako je uspješno zamijenjeno `n` znakova ili logičku laž, ako je neki od nizova bio kraći od `n`:

```
int zamijeni( _____ ) {
    char pomocni_znak;
    while( n && *a && *b ) {
        pomocni_znak = _____ ;
        _____ = _____ ;
        _____ = _____ ;
        _____; _____; n --;
    }
    return !n;
}
```

Zadatak 5. (6 bodova)

Koja je vrijednost pohranjena u varijablu tipa *float* ako je njezin sadržaj:

a) (3 boda) 20 000 000 000₈

Rješenje: _____

b) (3 boda) C1A0 0000₁₆

Rješenje: _____

Zadatak 6. (10 bodova)

Što će se ispisati na ekranu izvođenjem sljedećeg programskog odsječka:

```
#define MACRO(A,B,C) A/4-3*B+C
int main() {
    float f = 1;
    float *p = &f;
    f = 3 * MACRO(1.+3,3-2,4+3.);
    printf( "%06.2f", *p );
    return 0;
}
```

Rješenja:

Zadatak 1.

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

typedef struct sIgrac {
    int sifra;
    char ime_prezime[70 + 1];
} zIgrac;

void izvjesce(int sifIgrac, int god) {
    FILE *fb, *ft;
    int brPoraza = 0, brPobjeda = 0;
    int prvi, drugi, set1, set2, godina;
    int sifProtivnik, setProtivnik, setIgrac;
    zIgrac protivnik;

    ft = fopen("rezultati.txt", "r");
    fb = fopen("igraci.dat", "rb");

    printf(" Protivnik           | Rezultat\n");
    printf("-----\n");
    while (fscanf(ft, "%d %d %d %d %d %d", &prvi, &drugi, &set1, &set2, &godina) == 5) {
        if (godina == god && (prvi == sifIgrac || drugi == sifIgrac)) {
            if (prvi == sifIgrac) {
                sifProtivnik = drugi; setProtivnik = set2; setIgrac = set1;
            }
            else {
                sifProtivnik = prvi; setProtivnik = set1; setIgrac = set2;
            }
            fseek(fb, sifProtivnik * sizeof(zIgrac), SEEK_SET);
            fread(&protivnik, sizeof(zIgrac), 1, fb);
            printf("%4d %-20.20s| %d:%d\n", sifProtivnik, protivnik.ime_prezime, setIgrac,
                setProtivnik);
        }
    }
    /* end while */
    fclose(fb);
    fclose(ft);
}
```

Zadatak 2.

```
char *kombiniraj(char *novi, char *prvi, char *drugi, int n) {
    int i = 0, j = 0;
    char *p = novi;
    while (*prvi || *drugi) {
        for (i = 0; *prvi && i < n; i++) {
            *novi = *prvi;
            ++novi;
            ++prvi;
        }
        for (i = 0; *drugi && i < n; i++) {
            *novi = *drugi;
            ++novi;
            ++drugi;
        }
    } /* end while */
    *novi = '\\0';
    return p;
}

/* alternativno, skracena verzija: */
char* kombiniraj(char *novi, char *prvi, char *drugi, int n) {
    int i = 0, j = 0;
    char *p = novi;
    while (*prvi || *drugi) {
        for (i = 0; *prvi && i < n; i++) {
            *novi ++ = *prvi ++;
        }
        for (i = 0; *drugi && i < n; i++) {
            *novi ++ = *drugi ++;
        }
    } /* end while */
    *novi = '\\0';
    return p;
}
```

Zadatak 3.

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

/* popuni ostatak polja do dimenzija novir x novis sa slucajnim brojevima */
int prosiri( long *mat, int r, int s, int maxr, int maxs, int novir, int novis ) {
    int i, j;
    long maxval, minval;
    static int inicijalizirano = 0;
    if( novir > maxr || novis > maxs ) {
        return 0;
    }
    if( !inicijalizirano ) {
        srand( time( NULL ) );
        inicijalizirano = 1;
    }
    maxval = minval = mat[ 0 ];
    for( i = 0; i < r; i++ ) {
        for( j = 0; j < s; j++ ) {
            if( mat[ i * maxs + j ] > maxval ) {
                maxval = mat[ i * maxs + j ];
            }
        }
    }
}
```

```

        else if( mat[ i * maxs + j ] < minval ) {
            minval = mat[ i * maxs + j ];
        }
    }
}
/* osnovni nacin: */
for( i = 0; i < r; i++ ) {
    for( j = s; j < novis; j++ ) {
        mat[ i * maxs + j ] = minval
            + ( long )( rand() % ( maxval - minval + 1 ) );
    }
}
for( i = r; i < novir; i++ ) {
    for( j = 0; j < novis; j++ ) {
        mat[ i * maxs + j ] = minval
            + ( long )( rand() % ( maxval - minval + 1 ) );
    }
}
}
/* kraj osnovnog nacina */

/* alternativni nacin (manje efikasno, pogotovo za mala prosirenja velikih matrica*/
for( i = 0; i < novir; i++ ) {
    for( j = 0; j < novis; j++ ) {
        if( i >= r || j >= s ) {
            mat[ i * maxs + j ] = minval
                + ( long )( rand() % ( maxval - minval + 1 ) );
        }
    }
}
}
/* kraj alternativnog nacina */

return 1;
}

```

Zadatak 4.

```

int zamijeni( char *a, char *b, int n ) {
    char temp;
    while( n && *a && *b ) {
        temp = *a;
        *a = *b;
        *b = temp;
        a++; b++; n--;
    }
    return !n;
}

```

Zadatak 5.

- a) -0
- b) -20

Zadatak 6.

-01.00