

JMBAG	
Ime i prezime	

Programiranje i programsko inženjerstvo
Zimski ispitni rok 18. veljače 2016.

Rješenja svih zadataka osim 4. napisati na vlastitom papiru i predati u košuljici.

Rješenje 4. zadatka napisati na predviđeno mjesto uz zadatak.

1. (20 bodova) U tekstualnoj datoteci *prijenosi.txt* upisani su podaci o prijenosima sredstava s računa uplatitelja na račun primatelja koji su izvršavani od početka 1995. godine do kraja 2015. godine. Svaki zapis sadrži sljedeće podatke:

DD-MM-GGGG#hh:mm#broj_racuna_uplatitelja#broj_racuna_primatelja#iznos

Primjeri zapisa u datoteci *prijenosi.txt* :

```
1-1-1995#7:01#216002322#216000020#1523.34
...
31-12-2015#11:53#216002321#216000020#1523.34
31-12-2015#12:05#216102454#216300002#200.00
31-12-2015#23:58#216702334#216000005#12000.00
```

Binarna datoteka *saldo.bin* sadrži informacije o stanju računa klijenata banke na kraju 2013. godine. Podaci u jednom zapisu su: *brojRacuna* (int), *OIBklijenta* (char[11+1]), *stanjeRacuna* (double), *dozvoljeniMinus* (double). Prvi zapis u datoteci ima broj računa 216000000, drugi ima broj računa 216000001, itd. Za svaki broj računa koji se može pojaviti u datoteci *prijenosi.txt* sigurno postoji odgovarajući zapis u datoteci *saldo.bin*.

Ažurirajte datoteku *saldo.bin* podacima o prijenosima sredstava s računa uplatitelja na račun primatelja koji su izvršavani tijekom 2014. i 2015. godine kako bi zapisi u *saldo.bin* odražavali stanja računa klijenata na kraju 2015. godine. U rješenju trebate napisati definiciju funkcije i koristiti funkciju *azurirajSaldo* koja ima prototip:

```
void azurirajSaldo(FILE *saldo, int brojRacuna, double iznos)
```

(NAPOMENA: nije potrebno provjeravati uspješnost otvaranja datoteka, pozicioniranja u binarnu datoteku te čitanja i pisanja zapisa)

2. (25 bodova) Napisati funkciju *postaviMina* koja će u cjelobrojnu kvadratnu matricu reda n nasumično postaviti m „mina“. Mina se označava znamenkom 1, dok se polje na kojem nema mine označava znamenkom 0. Prije postavljanja mina funkcija mora inicijalizirati generator slučajnih brojeva, uz osiguranje da će se inicijalizacija provesti samo uz prvo pozivanje funkcije.

Napisati funkciju *staniNaPolje* koja prima kvadratnu matricu (popunjenu minama uz pomoć funkcije *postaviMina*) i par koordinata polja na koje korisnik "staje". Ako se radi o polju na kojem se nalazi mina, funkcija vraća -1. Ako na polju nema mine, funkcija vraća broj mina u okolici polja (prvi susjedi gledano horizontalno i vertikalno, ali ne i dijagonalno - pogledati sliku!).

	0	1	2	3	4	
0	0	0	1	0	0	(0,0) -> 1
1	1	0	1	1	1	(2,2) -> 3
2	0	0	0	1	0	(4,1) -> 0
3	1	0	1	0	0	(4,4) -> -1
4	0	0	0	0	1	

U glavnom programu učitati red matrice n ($5 \leq n \leq 10$) i broj mina m ($5 \leq m < n^2$). Učitavanje ponavljati sve dok se ne unesu ispravne vrijednosti. Potom započeti igru na sljedeći način: korisnik u svakom koraku igre unosi koordinate polja na koje "staje". Neispravne koordinate se ignoriraju. Ako na polju nema mine, korisnik dobiva informaciju o tome koliko mina se nalazi u okolici odabranog polja. Ako se na polju nalazi mina, igra završava uz odgovarajuću poruku.

3. (15 bodova) Napisati funkciju *brojPojava* koja će odrediti broj pojavljivanja pojedinih slova engleske abecede u znakovnom nizu te vratiti brojevni niz od 26 elemenata gdje svaki predstavlja broj pojavljivanja pojedinog slova gledano abecednim redom. Ako u nizu nema niti jednog slova abecede, funkcija vraća 0, inače vraća maksimalni broj pojavljivanja u nizu. Funkcija mora prebrojavati pojavu slova neovisno radi li se o velikom ili malom slovu.

U glavnom programu učitati sa tipkovnice niz znakova maksimalne duljine 100, te potom uz pomoć funkcije *brojPojava* ispisati slovo koje se u danom nizu pojavljuje najčešće. Pored slova ispisati frekvenciju pojavljivanja tog slova. U slučaju da postoji više slova koja su se pojavila najveći broj puta ispisati sva takva slova. U slučaju da u nizu nema slova, ispisati odgovarajuću poruku.

Primjeri učitanoz niza i ispisa:

<i>Peter Piper picked a peck of pickled peppers.</i> ↵ P 9	<i>AaBcDEe</i> ↵ A 2 E 2	27642!# ↵ U unesenom nizu nema slova
---	--------------------------------	---

4. (6 bodova) Prikazati kako se sljedeće konstante pohranjuju u memoriju računala (prikaz napisati u heksadekadskom obliku):

-12.625f	_____
'7'	_____
"ABC"	_____

5. (8 bodova) Napisati funkciju koja kao ulazne parametre prima skup od *n* osoba, a u pozivajući program vraća njihovu prosječnu visinu i težinu. Skup osoba treba predstaviti poljem struktura, pri čemu je struktura deklarirana sljedećom naredbom:

```
struct osoba {  
    char ime_prezime[30+1];  
    int visina;  
    float tezina;  
};
```

6. (6 bodova) Napisati što će biti ispisano na zaslon računala nakon izvođenja navedenog programskog koda, uz pretpostavku da je korisnik unio informaciju iz okvira desno (znak ↵ predstavlja tipku ENTER).

```
int a, b, rez;  
float f;  
char c, niz[3+1];  
rez = scanf("%2d%x%3f%c%s", &a, &b, &f, &c, niz);  
printf("%d %d %.2f %d %s %d", a, b, f, c, niz, rez);
```

12A.34↵ BCD↵

Rješenja:

1.

```
#include <stdio.h>
#include <string.h>
#define POSMAK 216000000

void azurirajSaldo(FILE *f, int broj, double iznos) {
    struct {
        int broj_racuna;
        char OIB_klijenta[11+1];
        double stanje_racuna;
        double dozv_minus;
    } z;
    fseek(f, (long) sizeof(z)*(broj-POSMAK), SEEK_SET);
    fread(&z, sizeof(z), 1, f);
    z.stanje_racuna += iznos;
    fseek(f, (long) -sizeof(z), SEEK_CUR);
    fwrite(&z, sizeof(z), 1, f);
}

int main(void) {

    char datum_lijevo[6+1], datum_desno[7+1];
    int uplatitelj, primatelj, godina;
    double iznos;
    char c;
    FILE *transfer, *saldo;

    transfer = fopen("prijenosi.txt", "r");
    saldo = fopen("saldo.bin", "r+b");

    while (fscanf(transfer, "%6s%d%7s%d%c%d%c%lf", datum_lijevo, &godina,
datum_desno, &uplatitelj, &c, &primatelj, &c, &iznos)==8) {

        if (godina = 2014 || godina = 2015) {
            azurirajSaldo(saldo, uplatitelj, -1*iznos);
            azurirajSaldo(saldo, primatelj, iznos);
        }

    }

    fclose(transfer);
    fclose(saldo);
    return 0;
}
```

2.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define MAX 10

void postaviMine(int p[], int n, int maksred, int m) {
    static int inic;
    int i,j,poz, postavljeno;

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

    /* inicijaliziramo polje bez mina */
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++)
            p[i*maksred+j] = 0;
```

```

/* stavljamo mine */
while (m!=0) {
    i = rand() % n;
    j = rand() % n;
    if (!p[i*maksred+j]) {
        p[i*maksred+j]++;
        m--;
    }
}
}

int staniNaPolje(int p[], int n, int maksred, int x, int y) {
    int brojac = 0;
    if (p[x*maksred+y]== 1) brojac = -1;
    else {
        if (x!=0 && p[(x-1)*maksred+y]) brojac++; /* gore */
        if (x!=n-1 && p[(x+1)*maksred+y]) brojac++; /* dolje */
        if (y!=0 && p[x*maksred+y-1]) brojac++; /* lijevo */
        if (y!=n && p[x*maksred+y+1]) brojac++; /* desno */
    }
    return brojac;
}

int main(void) {
    int matrica[MAX][MAX];
    int m, n, x, y, broj_mina, game_over=0;

    do {
        printf("Upisi red matrice i broj mina:");
        scanf("%d %d", &n, &m);
    } while ((n < 5) || (n > MAX) || (m < 5) || (m >= n * n));

    postaviMine(matrica[0], n, MAX, m);

    while (!game_over) {
        printf("Upisi koordinate x i y (od 0 do %d):", n-1);
        scanf("%d %d", &x, &y);
        if (x < 0 || x >= n || y < 0 || y >= n) continue;

        broj_mina = staniNaPolje(matrica[0], n, MAX, x, y);
        if (broj_mina == -1 ) {
            printf("BUUUUM!!!!\n");
            game_over = 1;
        } else printf("Broj mina u okolici tog polja: %d\n", broj_mina);
    }

    return 0;
}

```

ALTERNATIVNO RJEŠENJE za funkciju *PostaviMine*

```

void postaviMine(int p[], int n, int maksred, int m) {

    static int inic;
    int i,j, preostaloMina, poz, postavljeno;

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

```

```

preostaloMina = m;

/* inicijaliziramo polje bez mina */
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++)
        p[i*maksred+j] = 0;

/* stavljamo mine */
while (preostaloMina) {
    postavljeno = 0;
    poz = rand() % (n*n - m + preostaloMina);
    for (i = 0; i < n; i++) {
        for (j = 0; j < n; j++) {
            if (p[i*maksred+j]) poz++; /* pomikemo poziciju
                                         za dosad postavljene mine */
            if (i*n+j == poz) {
                p[i*maksred+j] = 1;
                preostaloMina--;
                postavljeno = 1;
                break;
            }
        }
    }
    if (postavljeno) break;
}
}
}

```

3.

```

#include <stdio.h>
#include <ctype.h>
#define MAX 100

int brojPojava(char *niz, int *frek) {
    int max = 0;
    for (; *niz; niz++)
        isalpha(*niz) && max < ++frek[(int) (toupper(*niz) - 'A')] && max++;
    return max;
}

int main(void) {
    char niz[MAX+1];
    int frek[26] = {0}, i, max; /* moze i frekv['Z' - 'A' + 1] */
    printf("Upisite niz:");
    gets(niz);

    if (max = brojPojava(niz, frek)) {
        for (i = 0; i < 26; i++)
            if (frek[i] == max)
                printf("%c %d\n", i+'A', frek[i]);
    }
    else
        printf("U unesenom nizu nema slova\n");

    return 0;
}

```

4.

-12.625f	C1 4A 00 00
'7'	37
"ABC"	41 42 43 00

5.

```
void prosjMjere(struct osoba skup[], int n, float *prosVisina, float *prosTezina)
{
    int i;
    *prosVisina = *prosTezina = 0.f;
    for (i = 0; i < n; i++) {
        *prosVisina += skup[i].visina;
        *prosTezina += skup[i].tezina;
    }
    *prosVisina /= n;
    *prosTezina /= n;
}
```

-

alternativno rješenje:

```
struct rezultat {
    float prosVisina;
    float prosTezina;
};

struct rezultat prosjMjere(struct osoba skup[], int n) {
    int i;
    struct rezultat r;
    r.prosVisina = r.prosTezina = 0.f;
    for (i = 0; i < n; i++) {
        r.prosVisina += skup[i].visina;
        r.prosTezina += skup[i].tezina;
    }
    r.prosVisina /= n;
    r.prosTezina /= n;
    return r;
}
```

6.

12 10 0.34 10 BCD 5