

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

Programiranje i programsko inženjerstvo

Izvanredni ispitni rok 21. rujna 2016.

Rješenja zadataka 1, 2 i 3 potrebno je napisati na vlastitim papirima i predati u košuljici.

Rješenja zadataka 4, 5 i 6 napisati na za to predviđeno mjesto uz zadatak i predati u košuljici.

Zadatak 1. (22 boda)

U binarnoj datoteci `prolasci.dat` pohranjeni su podaci o zabilježenim prolascima djelatnika Fakulteta kroz neki od ulaza u zgradu. Svaki zapis sadrži šifru zaposlenika čiji je prolazak zabilježen (`short`), datum kada je zaposlenik prošao (`int`), vrijeme u danu kada je prošao (`int`), te šifru vrata kroz koja je zabilježen prolazak (`char`). Datum je pohranjen kao broj dana od 1. siječnja 1970. godine, a vrijeme kao broj sekundi od ponoći tekućeg dana.

U direktnoj binarnoj datoteci `zaposlenici.dat` pohranjeni su osnovni podaci o djelatnicima gdje svaki zapis sadrži šifru djelatnika (`short`), jmbag zaposlenika (`char[10+1]`) te ime i prezime (`char[100+1]`) djelatnika, a redni broj zapisa u datoteci odgovara šifri zaposlenika. Potrebno je napisati funkciju:

```
int brminuta(char *jmbag, int dan);
```

koja izračunava koliko je na traženi dan punih minuta zaposlenik proveo unutar Fakulteta (119 sekundi broji se kao 1 minuta, 120 sekundi kao 2 minute). Prvi argument funkcije je jmbag djelatnika u istom obliku u kojem su brojevi pohranjeni u `zaposlenici.dat`, a drugi je dan u istom formatu kao datum u datoteci `prolasci.dat`. Pretpostaviti da za svaki dan u datoteci `prolasci.dat` postoji paran broj zapisa za svakog zaposlenika gdje prvi zapis predstavlja ulaz u fakultet, drugi izlaz iz fakulteta, treći ulaz itd. Također pretpostaviti da za svakog zaposlenika postoji barem jedan par zapisa u `prolasci.dat` za svaki dan te da svaki zaposlenik ima jedinstveni JMBAG. Nije potrebno provjeravati uspješnost otvaranja datoteka.

Zadatak 2. (15 bodova)

Potrebno je napisati funkciju `spoji` zadanu prototipom:

```
void spoji(char *niz1, char *niz2, *char rez);
```

Funkcija prima dva niza znakova na više mjesta (`niz1` i `niz2`) koje spaja u znakovni niz `rez`. Spajanje se obavlja naizmjeničnim dodavanjem jednog po jednog znaka iz ulaznih nizova i to redom iz `niz1`, `niz2`, `niz1`, `niz2` itd. Ako su ulazni nizovi različitih duljina, na kraj rezultatnog niza kopiraju se preostali znakovi iz duljeg niza.

Primjeri:

`niz1: abcdefghijkl` `niz2: 1234567` `rez: alb2c3d4e5f6g7hijkl`

Zadatak 3. (22 boda)

a) Napisati funkciju `kopiraj` koja prima niz cijelih brojeva i dvodimenzijско polje cijelih brojeva, oboje *proizvoljnih dimenzija*. Funkcija redom čita elemente iz ulaznog niza te ih kopira u zadano dvodimenzijско polje popunjavajući redak po redak polja *u obliku matrice zadanih dimenzija* (pogledati primjer) koje je također potrebno predati kao argumente funkcije. Funkcija kopira elemente sve dok se ne popune svi elementi zadane matrice. U slučaju kad je broj elemenata ulaznog niza manji od raspoloživog broja elemenata polja za pohranu, nepopunjene elemente polja funkcija treba postaviti na vrijednost 0 (pogledati primjer). Funkcija vraća ukupan broj nepopunjenih elemenata.

Primjer:

Sadržaj zadanog polja (7x7) nakon kopiranja
elemenata u obliku matrice zadanih dimenzija 4x5:

Ulazni niz (12 elemenata):

1	15	99	14	7	15	2	54	60	32	21	4
---	----	----	----	---	----	---	----	----	----	----	---

1	15	99	14	7
15	2	54	60	32
21	4	0	0	0
0	0	0	0	0
...
...
...

b) Napisati glavni program u kojemu se unosi broj elemenata (`n`) ulaznog niza (maksimalno 30 elemenata). Učitavanje je potrebno ponavljati dok se ne unese ispravna vrijednost. Nakon toga učitati `n` elemenata niza. Kreirati i dvodimenzijско polje dimenzija 10x10 te u njega pomoću funkcije implementirane pod a) kopirati elemente iz ulaznog niza u obliku matrice dimenzija 4x5 (pogledati primjer). Na zaslon ispisati sadržaj ažuriranog dijela polja nakon kopiranja i broj nepopunjenih elemenata polja.

Zadatak 4. (8 bodova)

Nadopuniti funkciju `brojiSlovaBrojeve` koja broji slova i brojeve u ulaznom znakovnom niz `ulaz` te njihov broj vraća preko parametara `brSlova` i `brBroj`.

```
void brojiSlovaBrojeve(char *ulaz, int *brSlova, int *brBroj){
    *brSlova = *brBroj = 0;
    while (*ulaz != _____) {

        if ( _____ ) {
            (*brSlova)++;
        } else if ( _____ ) {
            (*brBroj)++;
        }
        _____;
    }
}
```

Zadatak 5. (8 bodova)

Nadopuniti funkciju `baciKocku` tako da simulira bacanje zamišljene Rubikove kocke koja ima 12 različitih strana s oznakama iz intervala [1, 12]. Nadopuniti glavni program na način da se simulira slučajno odabrani broj bacanja kocke (iz intervala [100, 200]) te da se broj pojavljivanja pojedine strane kocke ispravno pohrani u polje `brojac`.

<pre>#include <stdlib.h> #include <time.h> int main(void) { int brojac[12] = {0}; int i, n; _____; n = _____; for (i = 0; i < n; ++i) { _____; } return 0; }</pre>	<pre>int baciKocku(void) { int rez; rez = _____; return rez; }</pre>
--	--

Zadatak 6. (5 bodova)

Na označenom mjestu napisati koji je ispis zadanog koda.

```
#include<stdio.h>

#define CIKLICKI(i) A[i%4]

int main(void) {
    int i, A[6] = { 1 };
    CIKLICKI(1) = 2;
    CIKLICKI(5) = 3;
    CIKLICKI(4 + 2) = CIKLICKI(1 + 3 + 1)--;
    for (i = 0; i < 6; i++) {
        printf("%d ", A[i]);
    }
    return 0;
}
```

Rješenja:

Zadatak 1.

```
#include <stdio.h>
#include <string.h>
typedef struct zap {
    short sifra;
    char JMBAG[10 + 1];
    char ime_prez[100 + 1];
} zaposlenik;
typedef struct prol {
    short sifra;
    int datum;
    int vrijeme;
    char vrata;
} prolaz;
int brminuta(char *jmbag, int dan) {
    int brsek = 0, trazena_sifra = -1, ulaz = 1, zadnji_ulaz;
    zaposlenik z;
    prolaz p;
    FILE *fprol, *fzap;
    fprol = fopen("prolasci.dat", "rb");
    fzap = fopen("zaposlenici.dat", "rb");
    while (fread(&z, sizeof(z), 1, fzap)) {
        if (!strcmp(z.JMBAG, jmbag)) {
            trazena_sifra = z.sifra;
            break;
        }
    }
    fclose(fzap);
    while (fread(&p, sizeof(p), 1, fprol)) {
        if (trazena_sifra == p.sifra && dan == p.datum) {
            if (ulaz) {
                zadnji_ulaz = p.vrijeme;
            }
            else {
                brsek += (p.vrijeme - zadnji_ulaz);
            }
            ulaz = 1 - ulaz;
        }
    }
    fclose(fprol);
    return brsek % 60;
}
```

Zadatak 2.

```
void spoji(char niz1[], char niz2[], char rez[]){
    int i, j, k, duljuk, dulj1, dulj2;
    dulj1 = strlen(niz1);
    dulj2 = strlen(niz2);
    duljuk = dulj1 + dulj2;
    for (i = 0, j = 0, k = 0; i < duljuk; i++){
        if (i % 2 == 1 || k >= dulj2){
            rez[i] = niz1[j++];
        } else if (i % 2 == 0 || j >= dulj1) {
            rez[i] = niz2[k++];
        }
    }
    rez[i] = '\0';
}

/* alternativno */
void spoji(char niz1[], char niz2[], char rez[]){
    int i, paran;

    for (i = 0; *niz1 != '\0' || *niz2 != '\0'; i++){
        paran = i % 2 == 1;

        if (paran && *niz2 != '\0' || *niz1 == '\0'){
```

```

        *(rez++) = *(niz2++);
    }
    else if (!paran && *niz1 != '\0' || *niz2 == '\0'){
        *(rez++) = *(niz1++);
    }
}

*rez = '\0';
}

```

Zadatak 3.

```

#include <stdio.h>
#define MAXR 10
#define MAXS 10
#define MAXNIZ 30
int kopiraj(int n, int *niz, int r, int s, int maxstu, int *polje){
    int i,j,red,stu,br=0;
    /* Inicijalizirati elemente polja na 0 */
    for(i=0; i<r; i++)
        for(j=0; j<s; j++)
            polje[i*maxstu+j]=0;
    red = 0;
    stu = 0;
    for (i=0; i<n; i++) {
        polje[red * maxstu + stu]=niz[i];
        br++;

        /* Popunjeni svi raspoloživi elementi? */
        if (br == r*s) break;

        stu++;
        if (stu == s) {
            /* Redak popunjen - resetiraj stup i skoci u novi red */
            stu = 0;
            red++;
        }
    }

    return r*s - br;
}

int main (void) {
    int niz[MAXNIZ], polje[MAXR][MAXS];
    int n, i, j, br;

    printf ("Ucitajte broj elemenata ulaznog niza: ");
    do {
        scanf("%d", &n);
    } while (n<=0 || n>MAXNIZ);

    for (i=0; i<n; i++) {
        printf("Unesite %d. element niza: ", i+1);
        scanf("%d", &niz[i]);
    }

    br = kopiraj(n, niz, 4, 5, MAXS, &polje[0][0]);

    for(i=0; i<4; i++){
        for(j=0; j<5; j++){
            printf("%d ", polje[i][j]);
        }
        printf("\n");
    }
    printf("Broj nepopunjenih elemenata: %d", br);

    return 0;
}

```

Zadatak 4.

```
void brojiSlovaBrojeve(char *ulaz, int *brSlova, int *brBroj){
    *brSlova = *brBroj = 0;

    while (*ulaz != _____){
        if ( _____){
            (*brSlova)++;
        }
        else if ( _____){
            (*brBroj)++;
        }
        _____;
    }
}

/* alternativno */
void brojiSlovaBrojeve(char *ulaz, int *brSlova, int *brBroj){
    *brSlova = *brBroj = 0;

    while (*ulaz != _____){
        if ( _____){
            (*brSlova)++;
        }
        else if ( _____){
            (*brBroj)++;
        }
        _____;
    }
}
```

Zadatak 5.

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

int main(void) {
    int brojac[12] = {0};
    int i, n;

    srand ((unsigned) time(NULL));

    n = rand() % 12 + 1;
    for (i = 0; i < n; ++i) {
        ++brojac[baciKocku()-1];
    }

    return 0;
}

int bacikocku(void) {
    int rez;
    rez = rand() % 101 + 100;
    return rez;
}
```

ILI: $n = (\text{float}) \text{rand}() / (\text{RAND_MAX} + 1) * 12 + 1;$

ILI: $\text{rez} = (\text{float}) \text{rand}() / (\text{RAND_MAX} + 1) * 101 + 100;$

Zadatak 6.

13000-1