

JMBAG	
Ime i prezime	

Programiranje i programsko inženjerstvo

Završni ispit

26. siječnja 2017.

Rješenja 1, 2. i 3. zadatka napisati na vlastitim papirima i predati ih u košuljici.

Rješenje 4. zadatka napisati na za to predviđeno mjesto uz zadatak.

1. (12 bodova)

Zapis postojeće direktne binarne datoteke `osoba.bin` sadrži šifru osobe (*int*), prezime (*niz znakova duljine 20+1 znak*) i koeficijent (*float*). Redni broj zapisa odgovara šifri osobe. U datoteci nema "praznih" zapisa (zapisa u kojima nema podataka o osobi). Napisati program koji će posljednjoj osobi u datoteci čije je prezime jednako prezimenu prve osobe iz datoteke, koeficijent postaviti na prosječni koeficijent svih osoba, te na zaslone ispisati odgovarajuću poruku (za moguće oblike poruka pogledati primjere).

Primjer 1:

1	Horvat	3.5
2	Ban	2.5
3	Kolar	2.0
4	Horvat	7.0
5	Kolar	2.0
6	Horvat	1.5
7	Novak	2.5

Posljednji zapis s prezimenom Horvat je 6. zapis. Program će koeficijent u 6. zapisu postaviti na 3.0 i ispisati poruku:
Postavljen rbr. 6.

Primjer 2:

1	Turk	3.5
2	Ban	2.5
3	Kolar	2.0
4	Horvat	7.0
5	Kolar	2.0
6	Horvat	1.5
7	Novak	2.5

Posljednji zapis s prezimenom Turk je 1. zapis. Program će koeficijent u 1. zapisu postaviti na 3.0 i ispisati poruku:
Postavljen rbr. 1.

Primjer 3:

1	Novak	1.5
---	-------	-----

Posljednji zapis s prezimenom Novak je 1. zapis. Program će koeficijent u 1. zapisu postaviti na 1.5 i ispisati poruku:
Postavljen rbr. 1.

Primjer 4:

--	--	--

U datoteci nema niti jednog zapisa. Program će ispisati poruku:
Nema zapisa.

2. (12 bodova)

Napisati definiciju funkcije `gen` koja će u zadanom proizvoljno velikom dvodimenzijском polju cijelih brojeva za zadanih *m* redaka i *n* stupaca članove polja popuniti nulama i jedinicama, pri čemu vrijednost pojedinog člana, nulu ili jedinicu, odabire na slučajan način.

Napisati glavni program. U glavnom programu definirati cjelobrojno polje od 50 redaka i 100 stupaca. Pozivom funkcije generirati polje veličine 3 retka i 5 stupaca, te sadržaj polja ispisati na zaslon u obliku tablice. Postupak generiranja i ispisivanja polja ponoviti uzastopno deset puta. Ispis mora biti usklađen s oblikom ispisa prikazanim u primjeru.

Primjer:

```
1 0 0 1 0
0 0 1 1 0
1 1 1 1 0

0 0 0 0 1
0 0 0 1 1
0 1 1 0 0
```

... na ovom mjestu se ispisuje još 7 sličnih tablica

```
0 0 0 1 0
1 0 0 0 0
1 1 1 1 0
```

U modulu `funkcije.c` napisati funkciju `kazaljka` koja vraća pokazivač na prvo veliko slovo u zadanom znakovnom nizu (*string*). Ako u nizu nema niti jednog velikog slova, funkcija vraća NULL pokazivač.

U modulu `glavni.c` napisati glavni program koji s tipkovnice učitava jedan znakovni niz koji sigurno neće biti dulji od 50 znakova. Pozvati funkciju `kazaljka`, te na temelju dobivenog rezultata ispisati odgovarajuću poruku. Za moguće oblike poruka pogledati primjere izvršavanja programa.

Primjer 1:

Upisite niz: ac1gh!mHgF 1c
Prvo veliko slovo je: H
Ostatak niza je: gF 1c

Primjer 2:

Upisite niz: a2 l9M
Prvo veliko slovo je: M
Ostatak niza je:

Primjer 3:

Upisite niz: ab1 h!m
Nema velikog slova

4. (4 boda)

Prikazan je sadržaj tekstualne datoteke `ulaz.txt`.

Znakovi ↵ predstavljaju znakovne *newline* (\n), a znakovi □ predstavljaju praznine (*blank*).

ulaz.txt:

5-24↵
12341.55.11↵
A B C D E F G 0123456

Prikazan je odsječak izvornog koda programa `prog.c`. Izvršna verzija programa nalazi se u datoteci `prog.exe`. Program je pokrenut naredbom

```
prog.exe < ulaz.txt
```

Prikazati što će se tijekom izvršavanja programa ispisati na zaslonu, vodeći računa o točnoj poziciji ispisanih znakova: svaki "kvadratić" na papiru predstavlja mjesto na zaslonu na koje se ispisuje po jedan znak; svaki "redak kvadratića" predstavlja jedan redak na zaslonu.

prog.c (odsječak)

```
int i1, i2, i3, i4;

float x1, x2;

char s1[10], s2[10], c1, c2, c3;

scanf("%d %d %d%d", &i1, &i2, &i3, &i4);

printf("%3d %d %d %d\n", i1, i2, i3, i4);

fscanf(stdin, "%f%f", &x1, &x2);

fprintf(stderr, "%5.1fEEE%f\n", x1, x2);

scanf("%5s %s", s1, s2);

fprintf(stdout, "%s%s\n", s1, s2);

scanf("%c %c %c", &c1, &c2, &c3);

printf("%c%c%c", c1, c2, c3);
```

A large rectangular area filled with a uniform grid of small squares, typical of graph paper. The grid consists of 20 columns and 10 rows of squares, totaling 200 squares. The grid is used for drawing or plotting.

Rješenja

```
1.  #include <stdio.h>
    #include <string.h>
    int main (void) {
        FILE *tok;
        struct osoba_s {
            int sif;
            char prez[20+1];
            float koef;
        } oPrvi, oTekuci, oZadnji;

        int brojac = 0;
        float suma = 0.f;
        tok = fopen("osoba.bin", "r+b");

        while (fread(&oTekuci, sizeof(struct osoba_s), 1, tok) == 1) {
            ++brojac;
            suma += oTekuci.koef;
            if (brojac == 1) {
                oZadnji = oPrvi = oTekuci ;
            } else if (strcmp(oTekuci.prez, oPrvi.prez) == 0) {
                oZadnji = oTekuci;
            }
        }
        if (brojac == 0) {
            printf("Nema zapisa.");
        }
        else {
            oZadnji.koef = suma / brojac;
            fseek(tok, sizeof(struct osoba_s) * (oZadnji.sif - 1), SEEK_SET);
            fwrite(&oZadnji, sizeof(struct osoba_s), 1, tok);
            printf("Postavljen rbr. %d.\n", oZadnji.sif);
        }
        fclose(tok);
        return 0;
    }
```

alternativno

```
1. #include <stdio.h>
#include <string.h>
int main (void) {
    FILE *tok;
    struct osoba_s {
        int sif;
        char prez[20+1];
        float koef;
    } oPrvi, oTekuci, oZadnji;

    int brojac = 0;
    float suma = 0.f;
    tok = fopen("osoba.bin", "r+b");

    if (fread(&oPrvi, sizeof(struct osoba_s), 1, tok) == 1) {
        oZadnji = oPrvi;
        ++brojac;
        suma += oPrvi.koef;
        while (fread(&oTekuci, sizeof(struct osoba_s), 1, tok) == 1) {
            if (strcmp(oTekuci.prez, oPrvi.prez) == 0) {
                oZadnji = oTekuci;
            }
            ++brojac;
            suma += oTekuci.koef;
        }
        oZadnji.koef = suma / brojac;
        fseek(tok, sizeof(struct osoba_s) * (oZadnji.sif - 1), SEEK_SET);
        fwrite(&oZadnji, sizeof(struct osoba_s), 1, tok);
        printf("Postavljen rbr. %d.\n", oZadnji.sif);
    }
    else {
        printf("Nema zapisa.");
    }
    fclose(tok);
    return 0;
}
```

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

void gen (int *mat, int m, int n, int maxstup) {
    int i, j;
    for (i = 0; i < m; ++i)
        for (j = 0; j < n; ++j)
            *(mat + i * maxstup + j) = rand() % 2;
}

int main(void) {
    int i, j, k;
    int mat[50][100];

    srand((unsigned)time(NULL));
    for (k = 0; k < 10; ++k) {
        gen(&mat[0][0], 3, 5, 100);
        for (i = 0; i < 3; ++i) {
            for (j = 0; j < 5; ++j)
                printf("%d ", mat[i][j]);
            printf("\n");
        }
        printf("\n");
    }
    return 0;
}
```

3.

```
#include <ctype.h>
#include <stdlib.h>
#include "funkcije.h"

char *kazaljka (char *niz) {
    while (*niz != '\0' && ! isupper(*niz)) {
        ++niz;
    }
    return *niz == '\0' ? NULL : niz;
}
```

```
char *kazaljka (char *niz) {
    while (*niz != '\0') {
        if (isupper(*niz)) {
            return niz;
        }
        ++niz;
    }
    return NULL;
}
```

glavni.c

```
#include <stdio.h>
#include "funkcije.h"

int main (void) {
    char *pok, niz[50+1];
    printf("Upisite niz: ");
    gets(niz);
    pok = kazaljka(niz);
    if (pok == NULL) {
        printf("Nema velikog slova");
    }
    else {
        printf("Prvo veliko slovo je: %c\n", *pok);
        printf("Ostatak niza je: %s", pok+1);
    }
    return 0;
}
```

funkcije.h

```
char *kazaljka (char *niz);
```

4.

[illegible]