

**Napomene:**

- Savjetuje se navedene zadatke riješiti ubrzo nakon predavanja
- Savjetuje se ne gledati rješenja prije nego se pokuša samostalno riješiti zadatke

## 25. vježbe uz predavanja

1. U tekstualnu datoteku **ulaz.txt** editorom su upisani podaci o osobama (matični broj i prezime). Primjer sadržaja datoteke prikazan je ovdje:

```
952 Medvedec
101 Vurnek
205 Habajec
412 Voras
551 Ozimec
115 Zdilar
104 Pugelnik
471 Suhina
```

Napisati program koji će sadržaj datoteke **ulaz.txt** prepisati u novu binarnu datoteku **izlaz.bin**. Jedan zapis datoteke **izlaz.bin** sadrži: matični broj (long) i prezime (15+1 znak).

2. Napisati program koji će na zaslon ispisati matične brojeve i prezimena svih osoba u čijim se prezimenima nalazi malo slovo a. Podaci o osobama se nalaze u postojećoj binarnoj datoteci **izlaz.bin** koja je dobivena kao rezultat obavljanja programa iz prethodnog zadatka.

**Napomena:** kod rješavanja ovog, a također i svih sličnih zadataka s datotekama, nije dopušteno sve zapise "učitati u memoriju", npr. u jedno veliko polje, i onda po tom polju tražiti zapise koji zadovoljavaju neki uvjet. Umjesto toga, u memoriju učitati prvi zapis, provjeriti treba li na zaslon ispisati podatke iz tog zapisa, zatim učitati drugi zapis, itd.

3. Svaki zapis datoteke **slucajni.bin** treba sadržavati neki cijeli broj  $n$  (tipa int) koji može poprimiti vrijednosti iz intervala  $[2, 8]$ , te  $n$  cijelih brojeva (tipa int) koji poprimaju vrijednosti iz intervala  $[150, 160]$ . Napisati program koji će u novu binarnu datoteku **slucajni.bin** upisati 20 zapisa koji se formiraju na slučajan način. Za svaki zapis prvo treba na slučajan način odrediti  $n$ , a zatim na slučajan način odrediti svaku od  $n$  vrijednosti koje čine ostatak zapisa.
4. Napisati program koji će na zaslon ispisati sve zapise datoteke **slucajni.bin** koja je nastala kao rezultat obavljanja programa iz prethodnog primjera. Npr. ispis može izgledati ovako:

```
6 153 160 155 159 152 152
5 160 152 150 153 160
2 160 159
4 152 152 155 151
5 155 155 158 159 153
3 151 158 151
2 150 153
2 158 156
4 150 151 153 152
3 156 153 154
8 150 154 157 155 155 155 158 156
4 153 159 155 157
8 155 150 159 152 152 154 152 160
3 153 156 158
7 157 158 160 160 155 158 157
2 151 159
3 156 152 159
7 157 157 152 153 152 155 151
8 160 158 159 157 154 156 159 152
5 157 159 160 153 157
```

5. Napisati program kojim će se u novu binarnu datoteku `slova.bin` redom upisati sva velika, a nakon njih sva mala slova abecede.

6. Jednim pozivom funkcije `scanf` u varijablu `s` učitati s tipkovnice vrijednosti svih njenih članova:

```
struct str1 {
    int stanje[2];
};
struct str2 {
    struct str1 x;
    float k;
} s;
```

7. Jednim pozivom funkcije `scanf` u varijablu `w` učitati s tipkovnice vrijednosti svih njenih članova:

```
struct str1 {
    float stanje[2];
};
struct str2 {
    struct str1 y;
    int k;
    char ime[10+1];
} w;
```

8. Jednim pozivom funkcije `scanf` u varijablu `b` učitati s tipkovnice vrijednosti svih njenih članova:

```
struct str1 {
    float x;
    int k;
};
struct str2 {
    struct str1 z1;
    struct str1 z2;
} b;
```

9. U `student.h` opisati strukturu `student_s` koja je namijenjena pohrani podataka o jednom studentu i njegovim ocjenama u dva semestra. Za studenta se evidentira jmbag (niz od 10 znakova), imePrezime (niz znakova ne dulji od 40 znakova), broj ocjena u prvom semestru (cijeli broj ne veći od 6), ocjene u prvom semestru (polje cijelih brojeva ne dulje od 6), broj ocjena u drugom semestru (cijeli broj ne veći od 8) i ocjene u drugom semestru (polje cijelih brojeva ne dulje od 8).

U nastavku `student.h` deklarirati funkciju `racunajProsjeke` koja za zadanog studenta (jedan od formalnih argumenata je dakle struktura koja sadrži sve podatke o studentu) izračunava prosjek ocjena za prvi semestar (realni broj) i prosjek ocjena za drugi semestar (realni broj).

U nastavku `student.h` deklarirati funkciju `ispisPodataka` koja za zadanog studenta pozivom funkcije `racunajProsjeke` izračunava prosjek ocjena za prvi i drugi semestar, te na zaslon ispisuje podatke o studentu u obliku:

```
0012345678 Ivan Horvat
1. semestar: 3 4 4 3, prosjek 3.50
2. semestar: 3 4 4 3 5 5 3 2, prosjek 3.63
```

U glavnom programu **inicijalizatorom uz definiciju varijable** inicijalizirati varijablu `jedanStudent` koja odgovara strukturi `student_h` i pozivom funkcije `ispisPodataka` ispisati podatke o studentu.

10. U student.h iz prethodnog zadatka dodati deklaraciju funkcije generirajStudenta. Funkcija stvara i u pozivajući program vraća na slučajan način generirane podatke o studentu u strukturi koja odgovara strukturi student\_h. ImePrezime se generira kao slučajna kombinacija 10 znakova između 'O' i '9'. ImePrezime se generira kao slučajna kombinacija 20 znakova između 'A' i 'Z'. Broj ocjena (između 1 i 6) i ocjene (1 do 5) u prvom semestru, te broj ocjena (između 1 i 8) i ocjene (1 do 5) u drugom semestru također se generiraju generatorom pseudoslučajnih brojeva. U student.c dopisati definiciju funkcije generirajStudenta.

U glavnom programu definirati polje koje sadrži podatke za 10 studenata. Uzastopnim pozivanjem funkcije generirajStudenta generirati sve elemente polja (u svakom elementu polja će biti podaci o jednom studentu), a zatim uzastopnim pozivanjem funkcije ispisPodataka (iz prethodnog zadatka) ispisati podatke o svim studentima. Npr.

```
3238588347 VYWXZBDZUCBCEUUMPZTE
1. semestar: 3 5, prosjek 4.00
2. semestar: 2 1 1 3 1 1 5 4, prosjek 2.25
1228471688 OPOYBCIWGORJUPTYREIP
1. semestar: 4 1 3, prosjek 2.67
2. semestar: 2, prosjek 2.00
5399851236 LQGOHGSWFOZAGARMDUYV
1. semestar: 4, prosjek 4.00
2. semestar: 1 3 3 2 1 1 4 1, prosjek 2.00
2418024532 AMLGQPXGHLNCPMHDBQFJ
1. semestar: 3 4 3, prosjek 3.33
2. semestar: 5 3 2 4 1, prosjek 3.00
6578313733 PLUSTZOGDINVJTQSNLDU
1. semestar: 1 2 4 1 1 3, prosjek 2.00
2. semestar: 5 2 4, prosjek 3.67
5699228480 OBAHSJMEPWNUOBJQZWBY
1. semestar: 4 4, prosjek 4.00
2. semestar: 4 3 3 4 4, prosjek 3.60
7370116510 KQLULGLRTMMNILRQWENE
1. semestar: 4 3 2, prosjek 3.00
2. semestar: 1 4 5, prosjek 3.33
4210533926 HZBDKUWAUNDAYABNOKLE
1. semestar: 4, prosjek 4.00
2. semestar: 1 2 5 5 1 2 3, prosjek 2.71
2785697461 DLQWQLPDVETOGGPQOGPW
1. semestar: 5 3 1 2, prosjek 2.75
2. semestar: 4 1 5 1 3 5 3 4, prosjek 3.25
8217092120 VIPJNMIYCBTIWSQBZDVW
1. semestar: 1 2, prosjek 1.50
2. semestar: 5 4, prosjek 4.50
```

## Rješenja

### Rješenje 1. zadatka

```
#include <stdio.h>

int main (void) {
    FILE *ulTok, *izTok;

    long int mbr;
    char prez[15+1];

    ulTok = fopen ("ulaz.txt", "r");
    izTok = fopen ("izlaz.bin", "wb");

    while (fscanf(ulTok, "%ld%s",
                  &mbr,
                  prez) == 2) {
        fwrite(&mbr, sizeof(mbr), 1, izTok);
        fwrite(prez, sizeof(prez), 1, izTok);
    }
    fclose (ulTok);
    fclose (izTok);

    return 0;
}
```

### Rješenje 2. zadatka

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

int main (void) {
    FILE *ulTok;
    long int mbr;
    char prez[15+1];

    ulTok = fopen ("izlaz.bin", "rb");

    while (fread(&mbr, sizeof(mbr), 1, ulTok) == 1 &&
           fread(prez, sizeof(prez), 1, ulTok) == 1)
        if (strchr(prez, 'a') != NULL)
            printf("%d %s\n", mbr, prez);
    fclose (ulTok);
    return 0;
}
```

### Rješenje 3. zadatka

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

#define MIN_N 2
#define MAX_N 8
#define DG 150
#define GG 160
#define BROJ_ZAPISA 20

int main (void) {
    FILE *izTok;
    int i, j, n, polje[MAX_N];

    izTok = fopen ("slucajni.bin", "wb");
    /* inicijalizacija generatora sluc. brojeva */
    srand ((unsigned) time(NULL));

    for (i = 1; i <= BROJ_ZAPISA; ++i) {
        /* odaberi slucajan n */
        n = rand() % (MAX_N - MIN_N + 1) + MIN_N;
        /* odaberi n slucajnih brojeva */
        for (j = 0; j < n; ++j)
            polje[j] = rand() % (GG - DG + 1) + DG;
        fwrite(&n, sizeof(n), 1, izTok);

        fwrite(polje, sizeof(int), n, izTok);
        /* ili fwrite(polje, sizeof(int)*n, 1, izTok); */
    }
    fclose (izTok);
    return 0;
}
```

### Rješenje 4. zadatka

```
#include <stdio.h>

#define MAX_N 8

int main (void) {
    FILE *ulTok;
    int i, n, polje[MAX_N];

    ulTok = fopen ("slucajni.bin", "rb");

    while (fread (&n, sizeof(n), 1, ulTok) == 1) {
        fread (polje, sizeof(int), n, ulTok);
        /* ili fread (polje, sizeof(int)*n, 1, ulTok); */
        printf("%d ", n);
        for (i = 0; i < n; ++i)
            printf("%d ", polje[i]);
        printf("\n");
    }
    fclose (ulTok);
    return 0;
}
```

## Rješenje 5. zadatka

```
#include <stdio.h>

int main (void) {
    FILE *izTok;
    char c;
    izTok = fopen ("slova.bin", "wb");
    for (c = 'A'; c <= 'Z'; ++c) {
        fwrite(&c, sizeof(c), 1, izTok);
    }
    for (c = 'a'; c <= 'z'; ++c) {
        fwrite(&c, sizeof(c), 1, izTok);
    }
    fclose (izTok);
    return 0;
}
```

## Rješenje 6. zadatka

```
scanf("%d%d%f", &s.x.stanje[0], &s.x.stanje[1], &s.k);

/* ispis radi provjere */
printf("%d\n%d\n", s.x.stanje[0], s.x.stanje[1]);
printf("\n");
printf("%f\n", s.k);
```

## Rješenje 7. zadatka

```
scanf("%f%f%d%10s", &w.y.stanje[0], &w.y.stanje[1], &w.k, w.ime);
/* ispis radi provjere */
printf("%f\n%f\n", w.y.stanje[0], w.y.stanje[1]);
printf("\n");
printf("%d\n", w.k);
printf("\n");
printf("%s\n", w.ime);
```

## Rješenje 8. zadatka

```
scanf("%f%d%f%d", &b.z1.x, &b.z1.k, &b.z2.x, &b.z2.k);
/* ispis radi provjere */
printf("%f\n%d\n", b.z1.x, b.z1.k);
printf("\n");
printf("%f\n%d\n", b.z2.x, b.z2.k);
```

## Rješenje 9. zadatka

### student.h

```
struct student_s {
    char jmbag[10+1];
    char imePrezime[40+1];
    int brOcjenalsem;
    int ocjene1sem[6];
    int brOcjena2sem;
    int ocjene2sem[8];
};

void racunajProsjeke(struct student_s student
                    , float *prosjeck1sem
                    , float *prosjeck2sem);

void ispisPodataka(struct student_s student);
```

### student.c

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

void racunajProsjeke(struct student_s student
                    , float *prosjeck1sem
                    , float *prosjeck2sem) {
    int i, suma;
    suma = 0;
    for (i = 0; i < student.brOcjenalsem; ++i) {
        suma += student.ocjene1sem[i];
    }
    *prosjeck1sem = (float)suma / student.brOcjenalsem;
    suma = 0;
    for (i = 0; i < student.brOcjena2sem; ++i) {
        suma += student.ocjene2sem[i];
    }
    *prosjeck2sem = (float)suma / student.brOcjena2sem;
    return;
}

void ispisPodataka(struct student_s student) {
    float prosjek1, prosjek2;
    int i;
    racunajProsjeke(student, &prosjeck1, &prosjeck2);
    printf("%s %s\n", student.jmbag, student.imePrezime);
    printf("1. semestar:");
    for (i = 0; i < student.brOcjenalsem; ++i) {
        printf(" %d", student.ocjene1sem[i]);
    }
    printf(", prosjek %4.2f\n", prosjek1);
    printf("2. semestar:");
    for (i = 0; i < student.brOcjena2sem; ++i) {
        printf(" %d", student.ocjene2sem[i]);
    }
    printf(", prosjek %4.2f\n", prosjek2);
    return;
}
```

### glavni.c

```
#include "student.h"
int main (void) {
    struct student_s jedanStudent = {
        "0012345678"
        , "Ivan Horvat"
        , 4
        , {3, 4, 4, 3}
        , 8
        , {3, 4, 4, 3, 5, 5, 3, 2}
    };
    ispisPodataka(jedanStudent);

    return 0;
}
```

## Rješenje 10. zadatka

**student.h (samo dopuna zaglavlne datoteke iz prethodnog zadatka)**

```
struct student_s generirajStudenta(void);
```

**student.c (samo dopuna modula iz prethodnog zadatka)**

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

```
struct student_s generirajStudenta(void) {  
    /* racuna se da je srand vec pozvan u pozivajucem programu */  
    struct student_s noviStud;  
    int i;  
    for (i = 0; i < 10; ++i) {  
        noviStud.jmbag[i] = rand()%('9'-'0'+1) + '0';  
    }  
    noviStud.jmbag[10] = '\\0';  
  
    for (i = 0; i < 20; ++i) {  
        noviStud.imePrezime[i] = rand()%('Z'-'A'+1) + 'A';  
    }  
    noviStud.imePrezime[20] = '\\0';  
  
    noviStud.brOcjenalsem = rand()%6 + 1;  
    for (i = 0; i < noviStud.brOcjenalsem; ++i) {  
        noviStud.ocjenalsem[i] = rand()%5 + 1;  
    }  
  
    noviStud.brOcjena2sem = rand()%8 + 1;  
    for (i = 0; i < noviStud.brOcjena2sem; ++i) {  
        noviStud.ocjene2sem[i] = rand()%5 + 1;  
    }  
  
    return noviStud;  
}
```

**glavni.c (novi glavni program)**

```
#include <stdlib.h>  
#include <time.h>  
#include "student.h"  
int main (void) {  
    struct student_s studenti[10];  
    int i;  
    srand((unsigned)time(NULL));  
    for (i = 0; i < 10; ++i) {  
        studenti[i] = generirajStudenta();  
    }  
    for (i = 0; i < 10; ++i) {  
        ispisPodataka(studenti[i]);  
    }  
    return 0;  
}
```