

## 25. Vježbe uz predavanja

1. Napisati program kojim će se u novu neformatiranu datoteku `slova.bin` redom upisati sva velika, a nakon njih sva mala slova abecede.
2. Funkcijama `fseek` i `ftell` odrediti i ispisati na zaslon broj znakova u datoteci `slova.bin` iz prethodnog zadatka. Zatim na slučajan način (funkcijom `rand`) odabirati pozicije u datoteci, sa svake odabrane pozicije pročitati jedan znak i ispisati ga na zaslon. Postupak završiti nakon što se na zaslon ispiše 40 znakova ili veliki znak 'X'.
3. Jednim pozivom `scanf` funkcije u varijablu `s` učitati vrijednosti svih njenih članova:

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

4. Jednim pozivom `scanf` funkcije u varijablu `w` učitati vrijednosti svih njenih članova:

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

5. Jednim pozivom `scanf` funkcije u varijablu `b` učitati vrijednosti svih njenih članova:

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

6. U formatiranu datoteku `ulaz.txt` editorom su upisani podaci o osobama (matični broj, ime, prezime i datum rođenja). Primjer sadržaja datoteke prikazan je ovdje:

```
952 Nikolina Medvedec 15.05.1989
101 Davor Vurnek 17.06.1987
205 Zoran Habajec 09.12.1987
412 Davor Voras 11.06.1988
551 Zvonimir Ozimec 24.02.1989
115 Franjo Zdilar 22.01.1986
104 Stipe Pugelnik 03.05.1987
471 Nina Suhina 08.06.1988
```

Napisati program koji će sadržaj datoteke `ulaz.txt` prepisati u novu neformatiranu datoteku `izlaz.bin`. Jedan zapis datoteke `izlaz.bin` sadrži: matični broj (long), ime (15+1 znak), prezime (15+1 znak), dan rođenja (short), mjesec rođenja (short), godinu rođenja (short). Koristiti strukture!

7. Napisati program koji će na zaslon ispisati matične brojeve, imena i prezimena svih osoba koje rođendan imaju u mjesecu lipnju. Podaci o osobama se nalaze u postojećoj neformatiranoj datoteci **izlaz.bin** koja je dobivena kao rezultat obavljanja programa iz prethodnog zadatka. Koristiti strukture!
- 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.
8. Koristi se slijedna formatirana datoteka **ulaz.txt** opisana u 6. zadatku. Napisati program koji će na zaslon ispisati prezime i ime za svaki treći zapis iz datoteke, počevši od prvog zapisa (tj. za prvi zapis, za četvrti zapis, za sedmi zapis, itd).
9. U formatiranoj datoteci **kupljeno.txt** upisani su podaci o kupljenim artiklima. Zapis datoteke sadrži šifru artikla (3 znamenke) i broj kupljenih komada tog artikla (2 znamenke):

101	12
115	2

Zapis direktne neformatirane datoteke **artikli** sadrži šifru artikla (short), naziv artikla (20+1 znak) i cijenu jednog komada artikla (float). Redni broj zapisa u datoteci odgovara šifri artikla. Napisati program koji će na zaslon ispisati račun u sljedećem obliku:

	1	2	3	4
	1234567890	1234567890	1234567890	12345678901234
Telefon	Kanasonic	12	10.00	120.00 kn
CD Player	Suny	2	1100.10	2200.20 kn
UKUPNO:				2320.20 kn

10. U direktnoj neformatiranoj datoteci **bodovi** nalaze se podaci o 10 studenata i bodovima koje su dobili za zalaganje na nastavi. Svaki zapis sadrži matični broj (int), prezime i ime (20+1 znak) i broj bodova (int). Matični brojevi su u rasponu od 1-10, a redni broj zapisa odgovara matičnom broju. Napisati program kojim će se za jednog **slučajno** odabranog studenta za 10% povećati dotadašnju vrijednost njegovih bodova. Ograničiti uvećani broj bodova na maksimalnih 500 bodova.
11. Napisati program koji na kraj datoteke **mjerenja.txt** koja je opisana u 7. zadatku iz 23. vježbi uz predavanja, dodaje jedan novi zapis o mjerenjima temperature. Preko tipkovnice se učitava redni broj dana, broj mjerenja (ne više od 24) i parovi (sat, temperatura). Ukoliko u datoteci već postoji zapis o učitanoj rednom broju dana, dojavljuje se odgovarajuća poruka.
12. Uz pretpostavku da se dnevno mogu obaviti najviše 24 mjerenja temperature, prepisite datoteku **mjerenje.txt** u odgovarajuću **direktnu** neformatiranu datoteku **mjerenja.bin**. Redni broj zapisa u datoteci **mjerenja.bin** odgovara rednom broju dana.
13. Napišite program koji za redni broj dana učitava preko tipkovnice, čita zapis iz datoteke **mjerenja.bin** i ispisuje vremena mjerenja i temperature izmjerene tog dana. Ukoliko mjerenje za zadani dan ne postoji, ispisati odgovarajuću poruku.

**Rješenja svih zadataka provjerite prevođenjem i testiranjem vlastitih programa!**

## Rješenja: NE GLEDATI prije nego sami pokušate riješiti zadatke

### Rješenje 1. zadatka

```
#include <stdio.h>

int main () {
    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 2. zadatka

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

int main () {
    FILE *ulTok;
    char c;
    int brojZnakova, slucajnaPozicija, brojIspisanih = 0;

    ulTok = fopen ("slova.bin", "rb");
    fseek(ulTok, 0L, SEEK_END);
    /* znak je velicine jednog bajta, stoga je broj znakova jednak broju bajtova */
    brojZnakova = ftell(ulTok);
    printf("Broj znakova u datoteci jest: %d\n", brojZnakova);

    srand((unsigned)time(NULL));
    do {
        /* slucajna pozicija u datoteci: [0, brojZnakova - 1] */
        slucajnaPozicija = rand() % brojZnakova;
        fseek(ulTok, slucajnaPozicija, SEEK_SET);
        fread(&c, sizeof(c), 1, ulTok);
        printf("%c", c);
        brojIspisanih++;
    } while (c != 'X' && brojIspisanih < 40);
    printf("\n");
    fclose (ulTok);
    return 0;
}
```

### Rješenje 3. 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 4. 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 5. 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 6. zadatka

```
#include <stdio.h>
int main () {
    FILE *ulTok, *izTok;
    char tocka;

    struct {
        long int mbr;
        char ime[15+1];
        char prez[15+1];
        short int dan;
        short int mjesec;
        short int godina;
    } osoba;

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

    while (fscanf(ulTok, "%ld%s%s%hd%c%hd%c%hd",
                &osoba.mbr,
                osoba.ime,
                osoba.prez,
                &osoba.dan,
                &tocka,
                &osoba.mjesec,
                &tocka,
                &osoba.godina) == 8) {
        fwrite(&osoba, sizeof(osoba), 1, izTok);
    }
    fclose (ulTok);
    fclose (izTok);
    return 0;
}
```

## Rješenje 7. zadatka

```
#include <stdio.h>

int main () {
    FILE *ulTok;
    struct {
        long int mbr;
        char ime[15+1];
        char prez[15+1];
        short int dan;
        short int mjesec;
        short int godina;
    } osoba;

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

    while (fread(&osoba, sizeof(osoba), 1, ulTok) == 1) {
        if (osoba.mjesec == 6)
            printf("%d %s %s\n",
                osoba.mbr,
                osoba.ime,
                osoba.prez);
    }
    fclose (ulTok);
    return 0;
}
```

## Rješenje 8. zadatka

```
#include <stdio.h>

int main () {
    FILE *ulTok;
    long int mbr;
    char ime[15+1];
    char prez[15+1];
    char datum[2+1+2+1+4+1];
    int rbr = 0;

    ulTok = fopen ("ulaz.txt", "r");

    while (fscanf(ulTok, "%ld%s%s%s",
        &mbr,
        ime,
        prez,
        datum) == 4) {
        if (rbr % 3 == 0)
            printf("%s %s\n", ime, prez);
        rbr++;
    }
    fclose (ulTok);
    return 0;
}
```

## Rješenje 9. zadatka

Zadatak ilustrira kako slijedno čitati **sve** zapise iz formatirane datoteke, te kako se, na temelju ključa potrage, pronalazi i čita zapis iz direktne neformatirane datoteke. Također, pokazuje kako se upotrebom formata podaci ispisuju na zaslون u točno traženom obliku.

```
#include <stdio.h>

int main () {
    FILE *kup, *art;

    struct {
        short sifArt;
        char nazArt[20+1];
        float cijena;
    } artZapis;

    short sifArt, kolicina;
    float suma = 0;

    kup = fopen("kupljeno.txt", "r");
    art = fopen("artikli", "rb");

    while (fscanf (kup, "%3hd%2hd", &sifArt, &kolicina) == 2) {
        /* Za svaki zapis iz kupljeno.txt treba pronaci cijenu u datoteci artikli. */
        /* Prvo, postavi kazaljku neposredno ISPRED odgovarajućeg zapisa */
        fseek(art, (long)sizeof(artZapis)*(sifArt-1), SEEK_SET);
        /* Procitaj cijeli zapis u strukturu */
        fread(&artZapis, sizeof(artZapis), 1, art);
        /* Pretpostavljamo da za svaki sifArt sigurno postoji zapis u artikli */
        printf("%s %2d %8.2f %8.2f kn\n",
            artZapis.nazArt,
            kolicina,
            artZapis.cijena,
            artZapis.cijena * kolicina);
        suma += artZapis.cijena * kolicina;
    }
    printf("UKUPNO:%34.2f kn", suma);
    fclose(kup);
    fclose(art);
    return 0;
}
```

## Rješenje 10. zadatka

Zadatak ilustrira kako se na slučajan način odabire redni broj zapisa, kako se čita odabrani zapis iz direktne formatirane datoteke, te kako se sadržaj zapisa direktne formatirane datoteke može promijeniti. Također ilustrira kako se ista datoteka može koristiti i za čitanje i za pisanje.

**NAPOMENA:** između operacija čitanja i pisanja u **isti tok podataka** mora se nalaziti barem jedan poziv funkcije `fseek` nad tim tokom podataka (čak i u slučaju kad nam pomak značke nije potreban jer se značka već nalazi na željenoj poziciji). Između operacija pisanja i čitanja **iz istog toka podataka**, mora se nalaziti barem jedan poziv funkcije `fseek` ili `fflush`. Funkcija `fflush(tokPodataka)` prazni međuspremnik toka podataka. Ne pridržavamo li se tih pravila, mogli bismo izvedeći svoje programe za vježbu dobiti posve pogrešne rezultate. **Na završnom ispitu se neće inzistirati na tom detalju (neće se "oduzimati bodovi" u slučaju nepoštovanja tih pravila)**

Dakle, ako u programu za vježbu imamo pisanje pa čitanje iz istog toka podataka

```
fwrite(..., tokPodataka);
/* ovdje negdje između se mora nalaziti ili fflush(tokPodataka) ili fseek(tokPodataka, ...) */
fread(..., tokPodataka);
```

Ako u programu za vježbu imate čitanje pa pisanje u isti tok podataka

```
fread(..., tokPodataka)
/* ovdje negdje između se mora nalaziti fseek(tokPodataka, ...) */
fwrite(..., tokPodataka)
```

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

int main () {
    FILE *dU1Iz1;
    struct {
        int mbr;
        char prezIme[20+1];
        int brBod;
    } zapis;
    int mbr;
    dU1Iz1 = fopen("bodovi", "r+b");

    /* inicijaliziraj generator slučajnih brojeva */
    srand((unsigned) time(NULL));
    /* odaberi slučajni broj iz intervala [1,10] */
    mbr = rand() % 10 + 1;

    printf("(Kontrolni ispis)povecati bodove za mbr: %d\n", mbr);
    /* Postavi kazaljku neposredno ISPRED odgovarajućeg zapisa */
    fseek(dU1Iz1, (long)sizeof(zapis)*(mbr-1), SEEK_SET);
    /* Procitaj cijeli zapis u strukturu */
    fread(&zapis, sizeof(zapis), 1, dU1Iz1);
    /* Povecaj broj bodova */
    zapis.brBod *= 1.1;
    if (zapis.brBod > 500)
        zapis.brBod = 500;
    /* VAZNO: ne zaboraviti zapisati promijenjene podatke natrag u datoteku!!! */
    /* Postavi kazaljku neposredno ISPRED odgovarajućeg zapisa jer nakon
       prethodnog citanja, kazaljka je bila neposredno IZA odgovarajućeg zapisa */
    fseek(dU1Iz1, -1L*(long)sizeof(zapis), SEEK_CUR);
    /* Zapisi sadržaj cijele strukture u datoteku */
    fwrite(&zapis, sizeof(zapis), 1, dU1Iz1);

    fclose(dU1Iz1);
    return 0;
}
```

U ovom zadatku je `fseek` između `fread` i `fwrite` ionako bio potreban, pa se nije moglo pogriješiti u odnosu na prethodnu napomenu.

## Rješenje 11. zadatka

Zadatak ilustrira kako se na kraj slijedne formatirane datoteke može dodati zapis.

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

int main () {
    FILE *di;
    int dan, noviDan, i;
    int sati[24];
    float temp[24];
    int brojMjerenja;
    char ostatakRetka[512];
    char c;

    di = fopen("mjerenja.txt", "a+");

    /* učitati redni broj novog dana */
    printf("Upisite redni broj novog dana>");
    scanf("%d", &noviDan);

    /* ako redni broj dana već postoji, završiti program */
    while (fscanf(di, "%c%d%", &c, &dan) == 2) {
        if (dan == noviDan) {
            printf("Dan već postoji\n");
            exit(-1);
        }
        /* preskociti ostatak retka */
        fgets(ostatakRetka, 512, di);
    }

    /* učitati broj mjerenja */
    printf("Upisite broj mjerenja>");
    scanf("%d", &brojMjerenja);

    /* učitati parove u polja sati i temp */
    for (i=0; i < brojMjerenja; i++) {
        printf("Upisite %d. par sata i temperature>", i+1);
        scanf("%d %f", &sati[i], &temp[i]);
    }
    /* ovdje bi se moglo provjeriti jesu li unesene "duple"
    vrijednosti sati, ali to se nije tražilo */

    /* zapisati redni broj dana i broj mjerenja */
    /* fseek na kraj datoteke nije potreban jer je datoteka
    otvorena u modu "a+" */
    fprintf(di, "%d#%d#", noviDan, brojMjerenja);

    /* zapisati parove sat-temp */
    for (i = 0; i < brojMjerenja; i++) {
        fprintf(di, "%d#%.1f#", sati[i], temp[i]);
    }

    /* zapisati "novi red" \n */
    fprintf(di, "\n");

    fclose(di);
    return 0;
}
```

## Rješenje 12. zadatka

Zadatak ilustrira pisanje u direktnu formatiranu datoteku čiji zapis, između ostalog, sadrži polje.

```
#include <stdio.h>

int main () {
    FILE *du, *di;
    struct {
        int rbrDan;
        int brMjer;
        int sati[24];
        float temp[24];
    } zapis;
    int i;
    char c;
    float suma = 0.0f;

    du = fopen("mjerjenja.txt", "r");
    di = fopen("mjerjenja.bin", "wb");

    while (fscanf(du, "%d%c%d%c", &c, &zapis.rbrDan, &c, &zapis.brMjer, &c) == 5) {
        for (i = 0; i < zapis.brMjer; i++) {
            fscanf(du, "%d%c%f%c", &zapis.sati[i], &c, &zapis.temp[i], &c);
        }
        /* procitaj \n s kraja retka */
        fscanf(du, "%c", &c);
        /* zapisati u dir. neformatiranu datoteku */
        fseek(di, (long)sizeof(zapis)*(zapis.rbrDan-1), SEEK_SET);
        fwrite(&zapis, sizeof(zapis), 1, di);
    }
    fclose(du);
    fclose(di);

    return 0;
}
```

## Rješenje 13. zadatka

Zadatak ilustrira čitanje iz direktne neformatirane datoteke čiji zapis, između ostalog, sadrži polje. U rješenju zadatka treba uočiti da zapis direktne neformatirane datoteke može biti "prazan". Uočiti na koji se način utvrđuje da je zapis "prazan".

```
#include <stdio.h>

int main () {
    FILE *du;
    struct {
        int rbrDan;
        int brMjer;
        int sati[24];
        float temp[24];
    } zapis;
    int i, rbrDan;
    du = fopen("mjerjenja.bin", "rb");

    printf("Upisite redni broj dana>");
    scanf("%d", &rbrDan);

    fseek(du, (long)(rbrDan-1)*sizeof(zapis), SEEK_SET);
    fread(&zapis, sizeof(zapis), 1, du);

    if (zapis.rbrDan == rbrDan) {
        printf("mjerjenja za dan: %d\n", zapis.rbrDan);
        for (i=0; i < zapis.brMjer; i++) {
            printf("    Sat: %2d    Temperatura: %5.1f\n", zapis.sati[i], zapis.temp[i]);
        }
    }
    else {
        printf("Nema podataka o mjerjenju za zadani dan\n");
    }
    fclose(du);

    return 0;
}
```