



Svaku datoteku otvaramo funkcijom `fopen`, a zatvaramo funkcijom `fclose`. Opći oblik naredbe za otvaranje datoteke je: `fopen ("ime_datoteke","nacin_otvaranja");`, a za zatvaranje: `fclose(identifikator);`

Ime_datoteke je identifikator koji je dodijeljen datoteci pri pohrani na neki medij (npr. `proba.txt` ili slično). Ako datoteka nije u trenutačnoj mapi, potrebno je navesti i putanju do nje (npr. `C:/Jezici/Programi/popis.txt`).

Primjer:

```
#include <stdio.h>

int main()
{
    FILE *p;
    p=fopen ("C:/Jezici/Programi/popis.txt", "a+");
    if (p==NULL)
    {
        printf ("\nGreska pri otvaranju.");
        return 1;
    }
    fputs ("\ndodatak\n", p);
    fprintf (p, "\ndodatak\n");
    fclose (p);
    printf ("\nPozdrav.\n");
    return 0;
}
```

Zadatak 1. Napravite C program koji će omogućiti kreiranje datoteke `razred.dat`.

Program treba omogućiti upisivanje u datoteku ime, prezime i adresu za `n` učenika jednog razreda.

```
#include <stdio.h>

int main()
{
    int i, n;
    char ime[20], prez[20], adr[20];
    FILE *f;
    f=fopen ("razred.dat", "w+");
    if (f==NULL)
    {
        printf ("\nGreska pri otvaranju.");
        return 1;
    }
    puts ("\nKoliko imate učenika?");
    scanf ("%d", &n);
}
```

```

for (i=0; i<n; i++)
{
    printf ("\nPodaci %d. učenika:", i+1);
    printf ("\nIme:");
    scanf ("%s", ime);
    printf ("\nPrezime:");
    scanf ("%s", prez);
    printf ("\nAdresa:");
    scanf ("%s", adr);
    fprintf (f, "%s %s %s \n", prez, ime, adr);
}
fclose (f);
return 0;
}

```

Zadatak 2. Napravite C program koji omogućuje kreiranje datoteke odlikasi.dat. Program treba omogućiti upisivanje u datoteku ime razreda i broj učenika u tom razredu (za n razreda). Na zaslonu treba ispisati ukupan broj odličnih učenika.

```

#include <stdio.h>

int main()
{
    struct podaci
    {
        char ime[4];
        int br;
    }razr;
    int i, n, s=0;
    FILE *f;
    f=fopen ("odlikasi.dat", "w");
    if (f==NULL)
    {
        printf ("\nGreska pri otvaranju.");
        return 1;
    }
    puts ("\nKoliko imate razreda?");
    scanf ("%d", &n);
    for (i=1; i<=n; i++)
    {
        printf ("\nUpisite podatke za %d. razred:", i);
        printf ("\nRazred:");
        scanf ("%s", razr.ime);
        printf ("\nBr. odlikasa:");
        scanf ("%d", &razr.br);
        fprintf (f, "%s %d \n", razr.ime, razr.br);
    }
    fclose (f);
    f=fopen ("odlikasi.dat", "r");
    if (f==NULL)
    {
        printf ("\nGreska pri otvaranju.");
        return 1;
    }
    for (i=1; i<=n; i++)
    {
        fscanf (f, "%s %d", razr.ime, &razr.br);
        s+=razr.br;
    }
    fclose (f);
    printf ("\nUkupan broj odlikasa je: %d.\n", s);
    return 0;
}

```

Prilikom pisanja prethodna dva programa najviše posla je bilo unutar funkcija `fscanf()` i `fprintf()`. U takvim slučajevima je bolje koristiti se funkcijama **fwrite()** i **fread()** koje su namijenjene upisivanju i čitanju nizova podataka različitih duljina kao što su polja, strukture, polja struktura...

Opći oblik tih funkcija: `fwrite(blok_podataka, velicina_bloka, broj_blokova, f);`
`fread(blok_podataka, velicina_bloka, broj_blokova, f);`

gdje je:

- `blok_podataka` - pokazivač na blok podataka koji treba upisati,
- `velicina_bloka` - je veličina bloka podataka u bajtovima,
- `broj_blokova` - je broj blokova koje treba upisati ili ispisati,
- `f` - pokazivač na datoteku.

Zadatak 2, riješen upotrebom ovih funkcija:

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

```
int main()
{
    struct podaci
    {
        char ime[4];
        int br;
    }razr;
    int i, n, s=0;
    FILE *f;
    f=fopen ("odlikasi3.dat","w");
    if (f==NULL)
    {
        printf ("\nGreska pri otvaranju.");
        return 1;
    }
    puts ("\nKoliko imate razreda?");
    scanf ("%d",&n);
    for (i=1; i<=n; i++)
    {
        printf ("\nUpisite podatke za %d. razred:",i);
        printf ("\nRazred:");
        scanf ("%s", razr.ime);
        printf ("\nBr. odlikasa:");
        scanf ("%d", &razr.br);
        // umjesto: fprintf (f,"%s %d \n", razr.ime, razr.br);
        fwrite(&razr,sizeof(struct podaci),1,f);
    }
    fclose (f);
    f=fopen ("odlikasi3.dat","r");
    if (f==NULL)
    {
        printf ("\nGreska pri otvaranju.");
        return 1;
    }
    for (i=1; i<=n; i++)
    {
        // umjesto: fscanf (f,"%s %d", razr.ime, &razr.br);
        fread(&razr,sizeof(struct podaci),1,f);
        s+=razr.br;
    }
    fclose (f);
    printf ("\nUkupan broj odlikasa je: %d.\n", s);
    return 0;
}
```

Direktan pristup podacima u datoteci:

Ako u datoteci treba pronaći određeni podatak, moramo je pretražiti, a to radimo funkcijama `fseek()` i `ftell()`.

Funkcija `ftell()` nam daje trenutno mjesto pokazivača u datoteci (prvi znak ima mjesto 0).

Opći oblik funkcije: `ftell(f);`

gdje je: `f` - pokazivač na datoteku.

Funkcija `fseek()` omogućava postavljanje pokazivača položaja na točno određeno mjesto.

Opći oblik funkcije: `fseek(f,broj_bajtova,pocetno_mjesto);`

gdje je:

- `f` - pokazivač na datoteku,
- `broj_bajtova` - označava za koliko bajtova će se ostvariti pomak (to je sljedeće mjesto pokazivača - ofset),
- `pocetno_mjesto` - je mjesto od kojeg se obavlja pomak.

Argument `pocetno_mjesto` može poprimiti ove vrijednosti:

- `SEEK_SET` početak datoteke
- `SEEK_CUR` trenutno mjesto u datoteci
- `SEEK_END` kraj datoteke.

Direktan pristup datoteci je bolje upotrebljavati kod binarnih datoteka.

Zadaci

- 1) Napisati C program koji će kreirati datoteku `kupci.dat`. Program treba omogućiti upisivanje sljedećih podataka: ime, prezime, adresa i broj telefona svih kupaca neke tvrtke za uvoz.
- 2) Napisati C program kojim ćete omogućiti vođenje evidencije članova jedne videoteke. Program treba omogućiti korisniku kreiranje, čitanje ili dodavanje novih članova u datoteku `clanovi.dat` koja sadrži sljedeće podatke o članovima: prezime, ime, šifru, adresu i broj telefona. Kreiranje, čitanje ili dodavanje se odabire iz izbornika na početku programa.