

23. Vježbe uz predavanja

1. Napisati program koji će sadržaj formatirane datoteke `ulaz.txt` prepisati na zaslon, ali tako da se umjesto malih slova ispisuju velika. Za čitanje znakova iz datoteke treba koristiti funkciju `fgetc`. Pomoću editora (npr. `notepad`) prirediti formatiranu datoteku `ulaz.txt`, pohraniti ju unutar nekog kazala (direktorija), npr. `c:\tmp`, te testirati program.
2. Napisati funkciju `broji` koja kao argument prima niz znakova koji predstavlja **ime formatirane datoteke** (tip podatka `char *`). Funkcija treba otvoriti tok podataka za čitanje iz datoteke sa zadanim imenom (tj. "otvoriti datoteku"), prebrojati samoglasnike (broje se i "mali" i "veliki" samoglasnici), zatvoriti tok podataka (tj. "zatvoriti datoteku"), te u pozivajući program vratiti broj samoglasnika. Ukoliko funkcija ne uspije otvoriti datoteku, kao broj samoglasnika vraća -1. Za čitanje znakova iz datoteke treba koristiti funkciju `fgetc`. Pomoću editora (npr. `notepad`) prirediti jednu formatiranu datoteku, te napisati glavni program kojim će se testirati rad funkcije.
3. Slično kao u prethodnom zadatku. Funkcija `broji` kao argument prima **tok podataka** (tip podatka `*FILE`) koji je već otvoren za formatiranu datoteku. Glavni program otvara tok podataka za postojeću formatiranu datoteku, predaje ga funkciji, a kad funkcija obavi brojanje, glavni program ispisuje rezultat i zatvara tok podataka.
4. Prepraviti rješenje zadatka 6. (ispis tablice množenja) iz prethodnih vježbi uz predavanja. Tablicu množenja, umjesto na zaslon, treba zapisati u formatiranu datoteku čije ime treba učitati s tipkovnice. Editorom (npr. `notepad`) provjeriti sadržaj dobivene datoteke.
5. Napisati program koji pomoću funkcije `fscanf` čita realne brojeve iz formatirane datoteke `brojevi.txt`, te po završetku čitanja (kada se dođe do kraja datoteke ili se pri čitanju dogodi pogreška) na zaslon ispisuje ili aritmetičku sredinu učitanih brojeva ili poruku "Nije procitan niti jedan broj". Npr. za sadržaj datoteke:

22	13.5	-2	1
11.0	-3.5		

na zaslon treba ispisati: *7.000000*

Npr. za sadržaj datoteke

a11	4.5	22
4.2	11	

na zaslon treba ispisati: *Nije procitan niti jedan broj.*

6. U formatiranu datoteku `osobe.txt` editorom upisati podatke o osobama (matični broj, ime, prezime i datum rođenja). Primjer sadržaja datoteke prikazan je ovdje:

```
952 Nikolina Medvedec 15.5.1989
101 Iva Vurnek 17.6.1987
412 Natalija Voras 11.6.1988
551 Anatolij Ozimec 24.2.1989
115 Franjo Zdilar 22.11.1986
471 Ivana Suhina 8.6.1988
```

Napisati program koji će na zaslon ispisati sve podatke o osobama iz datoteke `osobe.txt` čije ime sadrži niz znakova `na`. Za datoteku iz primjera, program treba na zaslon ispisati sljedeće:

```
952 Nikolina Medvedec 15.5.1989
551 Anatolij Ozimec 24.2.1989
471 Ivana Suhina 8.6.1988
```

7. U formatiranoj datoteci `mjerenja.txt` nalaze se zapisi o mjerenjima temperature u obliku:

```
#ddd#NN#hh#x.x#hh#xx.x#hh#xx.x#...hh#xx.x#
```

U zapisu se nalazi redni broj dana u godini (ddd), broj mjerenja obavljenih tog dana (NN), sat u kojem je obavljeno mjerenje (hh), izmjerena temperatura (x.x). Npr.

```
#175#4#3#-11.5#12#2.0#16#5.0#22#-2.5#
```

Prethodni zapis znači da su u danu s rednim broje 175 obavljena 4 mjerenja: u 3 sata temperatura je bila -11.5 stupnjeva, u 12 sati 2 stupnja, u 16 sati 5 stupnjeva i u 22 sata -2.5 stupnja.

Za svaki zapis iz datoteke `mjerenja.txt` ispisati na zaslon redni broj dana i srednju vrijednost izmjerenih temperatura za taj dan.

8. Ovaj zadatak je važan za razumijevanje funkcije `fgets`. U predavanjima prvo pažljivo pročitati što točno radi funkcija `fgets`.

Uz pretpostavku da je sadržaj datoteke `podaci.txt`:

```
abcde
fghijk
lmnoprs
ABCDEFGH
```

što će biti sadržaj datoteke `izlaz.txt` nakon obavljanja sljedećeg programa:

```
#include <stdio.h>
#include <stdlib.h>
#define MAXLIN 7

int main () {
    FILE *du, *di;
    char linija[MAXLIN];

    du = fopen ("podaci.txt", "r");
    di = fopen ("izlaz.txt", "w");
    while (fgets(linija, MAXLIN, du) != NULL ) {
        fputs (linija, di);
    }
    fclose (du);
    fclose (di);

    return 0;
}
```

Rješenja svih zadataka provjerite prevođenjem i testiranjem vlastitih programa!

Rješenja:

Rješenje 1. zadatka

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

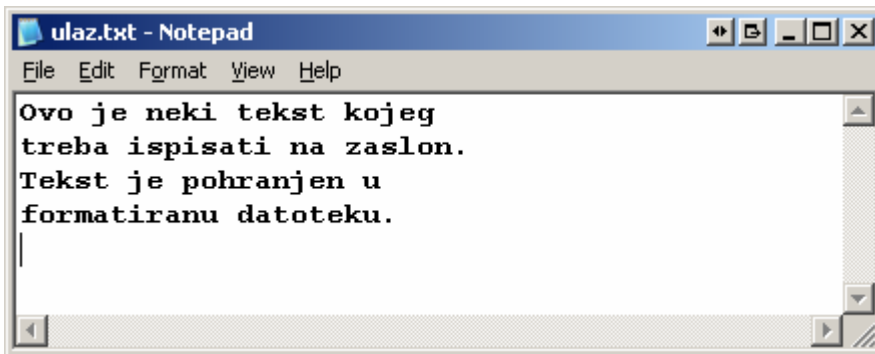
int main () {
    int c;
    FILE *tokPod;
    char *imeDat = "ulaz.txt";
    tokPod = fopen(imeDat, "r");

    if (tokPod == NULL) {
        printf("Ne mogu otvoriti %s\n", imeDat);
        exit(-99);
    }

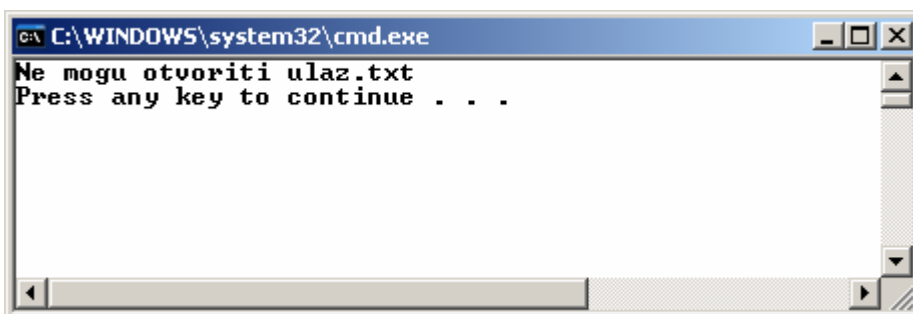
    while ((c = fgetc(tokPod)) != EOF)
        putchar(toupper(c)); /* ili fputc(toupper(c), stdout); */

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

Datoteka `ulaz.txt` napisana je pomoću editora (npr. notepad), te pohranjena u direktorij `c:\tmp`



Rezultat izvršavanja programa će biti ispis poruke



Što se dogodilo? Program ne može "otvoriti datoteku" jer datoteku `ulaz.txt` pokušava pronaći u radnom kazalu (*working directory*). U Visual Studio razvojnom okruženju je *working directory* obično podešen tako da odgovara kazalu u kojem je pohranjen projekt.

Moguće rješenje 1:

U C programu navesti apsolutni put do datoteke (*absolute path*) kojim će se u cijelosti opisati lokacija datoteke `ulaz.txt`, npr. tako da se umjesto imena datoteke "`ulaz.txt`" koristi "`c:\\tmp\\ulaz.txt`" ili "`c:/tmp/ulaz.txt`"

Moguće rješenje 2:

Promijeniti podešenje za *working directory* u Visual Studio razvojnom okruženju na sljedeći način: otvoriti prozor *properties* za svoj projekt, u *Configuration Properties* odabrati *Debugging*, za svojstvo *Working directory* upisati `c:/tmp` (odnosno neko drugo kazalo, ovisno o tome gdje se nalazi datoteka koju treba "otvoriti").

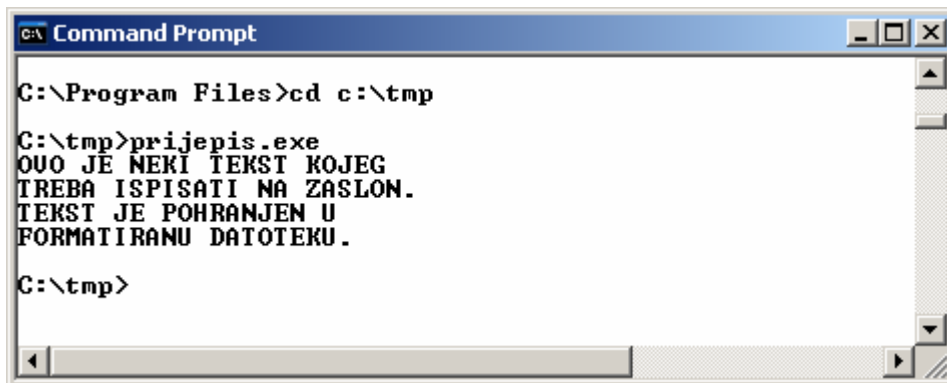
Moguće rješenje 3:

U razvojnom okruženju Visual Studio se obično kao rezultat prevođenja dobiva datoteka s izvršnim kodom čije ime odgovara imenu projekta, s ekstenzijom (nastavkom) `.exe`. Npr. ako je ima projekta `prijepis`, tada se u kazalu *debug* unutar kazala u kojem je pohranjen projekt, nakon prevođenja može pronaći datoteka `prijepis.exe`. Kopirati dotičnu datoteku u kazalo u kojem se nalazi datoteka `ulaz.txt`, pokrenuti *command-line interpreter* (tzv. *command prompt*), naredbom `cd` (change directory) pozicionirati se u kazalo u kojem se nalazi datoteka `ulaz.txt`, te pokrenuti izvršavanje programa. Radno kazalo će u tom slučaju biti kazalo iz kojeg je pokrenuto izvršavanje programa.

Npr. nakon otvaranja prozora *command prompt*, izvršiti naredbu

```
cd c:\tmp
```

i pokrenuti program:



```
C:\Program Files>cd c:\tmp
C:\tmp>prijepis.exe
OVO JE NEKI TEKST KOJEG
TREBA ISPISATI NA ZASLON.
TEKST JE POHRANJEN U
FORMATIRANU DATOTEKU.
C:\tmp>
```

Ovdje opisan problem (i jedno od rješenja) treba uzeti u obzir kod rješavanja i testiranja rješenja svih zadataka s datotekama. Međutim, **u rješenjima zadataka na završnom ispitu nije potrebno voditi računa o stvarnoj lokaciji datoteke, dovoljno je navesti samo ime datoteke (bez navođenja punog puta do datoteke).**

Rješenje 2. zadatka

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

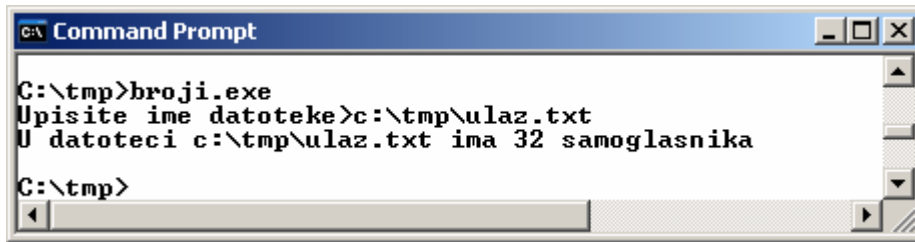
#define MAXIMEDAT 64

int broji (char *ime) {
    int broj;
    char c;
    FILE *ulTok;

    ulTok = fopen (ime, "r");
    if (ulTok != NULL) {
        broj = 0;
        while ((c = fgetc(ulTok)) != EOF) {
            c = toupper(c);
            if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
                broj++;
        }
        fclose(ulTok);
        return broj;
    }
    else {
        /* otvaranje ulaznog toka podataka nije uspjelo */
        broj = -1;
    }
    return broj;
}

int main () {
    char imedat[MAXIMEDAT+1];
    int broj;
    printf("Upisite ime datoteke>");
    scanf("%s", imedat);
    broj = broji(imedat);
    if (broj == -1)
        printf("Problemi kod otvaranja datoteke %s\n", imedat);
    else
        printf("U datoteci %s ima %d samoglasnika\n", imedat, broj);
    return 0;
}
```

Izvođenje programa (brojanje samoglasnika u datoteci c:\tmp\ulaz.txt iz prethodnog zadatka):



```
C:\tmp>broji.exe
Upisite ime datoteke>c:\tmp\ulaz.txt
U datoteci c:\tmp\ulaz.txt ima 32 samoglasnika
C:\tmp>
```

Rješenje 3. zadatka

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

#define MAXIMEDAT 64

int broji (FILE *ulTok) {
    int broj = 0;
    char c;
    while ((c = fgetc(ulTok)) != EOF) {
        c = toupper(c);
        if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
            broj++;
    }
    return broj;
}

int main () {
    char imedat[MAXIMEDAT+1];
    int broj;
    FILE *ulaz;
    printf("Upisite ime datoteke>");
    scanf("%s", imedat);

    ulaz = fopen (imedat, "r");
    if (ulaz != NULL) {
        broj = broji(ulaz);
        printf("U datoteci %s ima %d samoglasnika\n", imedat, broj);
        fclose(ulaz);
    }
    else {
        printf("Problemi kod otvaranja datoteke %s\n", imedat);
    }
    return 0;
}
```

Rješenje 4. zadatka

```
#include <stdio.h>

int main() {
    int n, i, j;
    char imeDat[64+1];
    FILE *tok;

    printf("Upisite ime datoteke: ");
    scanf("%s", imeDat);
    do {
        printf("Upisite vrijednost za n: ");
        scanf ("%d",&n);
    } while (n < 1 || n > 15 );

    tok = fopen(imeDat, "w");

    fprintf(tok, "Tablica mnozenja %dx%d\n", n, n);
    fprintf(tok, "      ");
    for (i = 1; i <= n; i++) {
        fprintf(tok, "%4d", i);
    }
    fprintf(tok, "\n");

    fprintf(tok, "----+", i);
    for (i = 1; i <= n; i++) {
        fprintf(tok, "----", i);
    }
    fprintf(tok, "\n");

    for (i = 1; i <= n; i++) {
        fprintf(tok, "%4d!", i);
        for (j = 1; j <= n; j++) {
            fprintf(tok, "%4d", i*j);
        }
        fprintf(tok, "\n");
    }

    fprintf(tok, "----+", i);
    for (i = 1; i <= n; i++) {
        fprintf(tok, "----", i);
    }
    fprintf(tok, "\n");

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

Rješenje 5. zadatka

```
#include <stdio.h>

int main () {
    float x, suma = 0.0f;
    int brojac = 0;
    FILE *ulTok;

    ulTok = fopen ("brojevi.txt", "r");
    while (fscanf(ulTok, "%f", &x) == 1 ) {
        suma += x;
        brojac++;
    }
    fclose (ulTok);
    if (brojac > 0)
        printf("%f\n", suma/brojac);
    else
        printf("Nije procitan niti jedan broj\n", suma/brojac);

    return 0;
}
```

Rješenje 6. zadatka

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

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

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

    while (fscanf(ulTok, "%d%s%s%hd%c%hd%c%hd",
                &mbr,
                ime,
                prez,
                &dan,
                &tocka,
                &mjesec,
                &tocka,
                &godina) == 8)
        if (strstr(ime, "na") != NULL)
            printf("%d %s %s %d.%d.%d\n", mbr, ime, prez, dan, mjesec, godina);
    fclose (ulTok);
    return 0;
}
```

Rješenje 7. zadatka

```
#include <stdio.h>

int main () {
    FILE *du;

    int dan, brojMj, sat, i;
    float temp;
    float suma;
    char c;

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

    while (fscanf(du, "%c%d%c%d%c", &c, &dan, &c, &brojMj, &c) == 5) {
        suma = 0.0f;
        for (i = 0; i < brojMj; i++) {
            fscanf(du, "%d%c%f%c", &sat, &c, &temp, &c);
            suma += temp;
        }
        printf("%3d: %f\n", dan, suma/brojMj);
        /* sad procitati \n koji je ostao neprocitan */
        fscanf(du, "%c", &c);
    }

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

Rješenje 8. zadatka

Nakon 1. čitanja sadržaj polja linija je: **abcde\n\0**

Nakon 2. čitanja sadržaj polja linija je: **fghijk\0**

Nakon 3. čitanja sadržaj polja linija je: **\n\0**

Nakon 4. čitanja sadržaj polja linija je: **lmnopr\0**

Nakon 5. čitanja sadržaj polja linija je: **s\n\0**

Nakon 6. čitanja sadržaj polja linija je: **ABCDEFGF\0**

Nakon 7. čitanja sadržaj polja linija je: **GH\0**

8. čitanje će vratiti NULL

Sadržaj datoteke `izlaz.txt` je:

abcde
fghijk
lmnoprs
ABCDEFGFH