

**Napomena**

U svim zadacima, na svim provjerama znanja i vježbama, vrijedi sljedeće

- uspješnost obavljanja operacija otvaranja i zatvaranja toka (datoteke) te čitanja i pisanja podataka u/iz toka (datoteke) treba provjeravati isključivo ako se to u zadatku eksplicitno traži
- ako u tekstu zadatka nije navedena mapa u kojoj se nalaze datoteke s kojima program treba baratati, tada se podrazumijeva da se nalaze u radnoj mapi

1. Napisati program koji će sadržaj datoteke `ulaz.txt` prepisati na zaslone, ali tako da se umjesto malih slova ispisuju velika (ostali znakovi na zaslon se ispisuju nepromijenjeni). Pomoću editora (npr. Notepad) prirediti datoteku `ulaz.txt` i pohraniti ju u mapu koja nije ista kao radna mapa. Npr. ako je radna mapa `c:/upro`, tada se datoteka može pohraniti u mapu `c:/tmp`. Testirati program.
2. Napisati funkciju `brojiSamoglasnike` s parametrom `imeDatoteke` koji je tipa niz znakova i koji predstavlja put (*path*) do datoteke. Funkcija treba otvoriti ulazni tok iz dotične datoteke ("otvoriti datoteku za čitanje"), prebrojati samoglasnike (broje se i "mali" i "veliki" samoglasnici), zatvoriti tok ("zatvoriti datoteku"), te u pozivajući program vratiti broj samoglasnika. Ako funkcija ne uspije otvoriti tok (datoteku), kao broj samoglasnika vraća -1.

Pomoću editora (npr. Notepad) kreirati jednu datoteku i pohraniti ju u neku mapu (npr. u mapi `c:/tmp` kreirati datoteku `test.txt`).

Napisati glavni program koji će učitati put do datoteke (sigurno neće biti dulji od 100 znakova i sigurno neće sadržavati bjeline), pozvati funkciju i na zaslon ispisati broj samoglasnika ili poruku u skladu s primjerom izvršavanja programa.

Primjeri izvršavanja programa

```
Upisite kompletan put do datoteke > C:/tmp/test.txt.↵  
Broj samoglasnika u datoteci C:/tmp/test.txt je 47
```

```
Upisite kompletan put do datoteke > C:/tmp/nemaTakveDatoteke.txt.↵  
Nije uspjelo otvaranje datoteke C:/tmp/nemaTakveDatoteke.txt
```

3. Prepraviti rješenje 4. zadatka (ispis tablice množenja) iz prethodnih vježbi uz predavanja. Tablicu množenja, umjesto na zaslon, treba zapisati u datoteku `tablica.txt`. Izvršiti program i editorom (npr. Notepad) provjeriti sadržaj dobivene datoteke.

**Rješenja:**

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

   int main(void) {
       FILE *tokPod = fopen("c:/tmp/ulaz.txt", "r");
       // nije se trazilo da provjerimo je li otvaranje upjelo
       // stoga ne provjeravamo.
       // ali zato moramo racunati da ce se mozda dogoditi
       // runtime pogreska

       int c;
       while ((c = fgetc(tokPod)) != EOF) {
           putchar(toupper(c));
       }

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

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

int brojiSamoglasnike(char *ime) {
    FILE *ulTok = fopen(ime, "r");

    int brojac = 0;
    if (ulTok != NULL) {
        int c;
        while ((c = fgetc(ulTok)) != EOF) {
            c = toupper(c);
            if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') {
                ++brojac;
            }
        }
        fclose(ulTok);
    } else {
        // otvaranje ulaznog toka podataka nije uspjelo
        brojac = -1;
    }

    return brojac;
}

#define MAXPATH 100

int main(void) {
    char kompletanPut[MAXPATH + 1];
    printf("Upisite kompletan put do datoteke > ");
    scanf("%s", kompletanPut);

    int brojSamogl = brojiSamoglasnike(kompletanPut);

    if (brojSamogl == -1) {
        printf("Nije uspjelo otvaranje datoteke %s", kompletanPut);
    } else {
        printf("Broj samoglasnika u datoteci %s je %d",
            kompletanPut, brojSamogl);
    }

    return 0;
}
```

## 3. #include &lt;stdio.h&gt;

```
int main(void) {
    int n;
    printf("Upisati n > ");
    scanf("%d", &n);

    FILE *tok = fopen("tablica.txt", "w");

    fprintf(tok, "Tablica mnozenja %d x %d\n", n, n);

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

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

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

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

    return 0;
}
```