

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
- na temelju primjera u zadatku ne smije se zaključiti kako u datoteci "ionako ima malo podataka", pa se zbog toga cijela datoteka može učitati u neko golemo polje i nakon toga nešto raditi s podacima koji se nalaze u tom polju. Datoteke općenito mogu biti vrlo velike, stoga se uvijek čitaju i obrađuju pojedinačni podaci ili pojedinačni zapisi u datoteci. Možete uočiti da niti u jednom rješenju u zadacima za vježbu nije u primarnu memoriju odjednom pročitani sadržaj cijele datoteke.

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 13. 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.

4. Što će biti sadržaj datoteke `izlaz.txt` nakon obavljanja sljedećeg programa ako se u tekstnoj datoteci `podaci.txt` nalazi sadržaj prikazan u desnom pravokutniku? Riješiti "na papiru" i rezultat provjeriti izvršavanjem na računalu.

```
#include <stdio.h>

#define MAXLIN 7

int main(void) {
    char redak[MAXLIN];

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

    return 0;
}
```

```
abcde↵
fghijk↵
lmnoprs↵
ABCDEFGH↵
```

5. Napisati program koji čita realne brojeve iz tekstne 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 odgovarajuću poruku.

Primjeri izvršavanja programa

Za sadržaj datoteke

```
22 13.5    -2  1↵
11.0 -3.5%2.2↵
```

Na zaslon ispisati

```
7.000000
```

Za sadržaj datoteke

```
a11 4.5 22↵
4.2 11↵
```

Na zaslon ispisati

```
Nije ucitan niti jedan broj
```

Za sadržaj datoteke

```
1 2 3↵
1 2 3↵
```

Na zaslon ispisati

```
2.000000
```

6. U tekstnoj datoteci `osobe.txt` upisani su podaci o osobama: matični broj (cijeli broj), ime (ne sadrži razmake, nije dulje od 20 znakova), prezime (ne sadrži razmake, nije dulje od 20 znakova) i datum rođenja u obliku prikazanom u primjeru.

Primjer sadržaja datoteke `osobe.txt`

```
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 8.6.1989.
```

Napisati program koji će na zaslon ispisati podatke o osobama iz datoteke `osobe.txt` čije ime sadrži niz znakova **na** ili su rođene 1987. godine. Za datoteku iz primjera, program treba na zaslon ispisati sljedeće:

```
952 Nikolina Medvedec.
101 Iva Vurnek.
551 Anatolij Ozimec.
471 Ivana Suhina.
```

7. U tekstnoj datoteci `mjerenja.txt` upisani su podaci 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. zapis

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

znači da su u danu s rednim brojem 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.

U jednom danu nikad se ne obavlja više od 24 mjerenja.

Napisati program koji će za svaki zapis iz datoteke `mjerenja.txt` na zaslon ispisati redni broj dana, broj mjerenja obavljenih u tom danu i srednju vrijednost u tom danu izmjerenih temperatura.

Primjer izvršavanja programa

Za sadržaj datoteke

```
#15#4#3#-11.5#7#2.0#19#5.0#23#-2.5.
#42#8#1#-5.5#4#2.0#7#5.0#11#12.5#14#11.5#16#2.0#19#0.5#23#-2.5.
#219#2#3#12.5#14#18.2.
#92#7#2#4.5#4#2.0#7#5.0#11#11.5#14#21.5#18#9.5#21#5.0.
```

Na zaslon ispisati

```
Dan= 15, Broj mjerenja= 4, Prosjek= -1.75.
Dan= 42, Broj mjerenja= 8, Prosjek= 3.19.
Dan=219, Broj mjerenja= 2, Prosjek= 15.35.
Dan= 92, Broj mjerenja= 7, Prosjek= 8.43.
```

8. Napisati program koji će u binarnu datoteku `fibonacci.bin` redom upisati prvih 40 članova Fibonaccijevog niza (1, 1, 2, 3, 5, ..., 102334155). Svaki član niza u datoteku upisati kao podatak tipa `int`.
9. Napisati program koji će s tipkovnice učitati cijeli broj `n`, a zatim na zaslon ispisati prvih `n` članova Fibonaccijevog niza, čitajući ih iz datoteke `fibonacci.bin` koja je nastala kao rezultat izvršavanja programa iz prethodnog zadatka.

Primjer izvršavanja programa

```
Upisite n > 12.↵
1.↵
1.↵
2.↵
3.↵
5.↵
8.↵
13.↵
21.↵
34.↵
55.↵
89.↵
144.↵
```

10. U tekstnoj datoteci `filmovi.txt` upisani su podaci o filmovima. Film je po žanru klasificiran u dramu, komediju ili SF. Naziv filma sigurno nije dulji od 60 znakova.

Primjer sadržaja datoteke `filmovi.txt`

```
Taxi Driver|drama|1976.↵
Modern Times|komedija|1936.↵
Citizen Kane|drama|1941.↵
2001: A Space Odyssey|SF|1968.↵
The Godfather|drama|1972.↵
Alien|SF|1979.↵
Chinatown|drama|1974.↵
Monty Python and the Holy Grail|komedija|1975.↵
```

Napisati program koji će stvoriti tri nove tekstne datoteke. Datoteka `drama.txt` treba sadržavati popis filmova žanra drame, datoteka `SF.txt` popis SF filmova, a datoteka `komedija.txt` popis komedija. Za datoteku iz primjera, program treba stvoriti sljedeće datoteke:

```
Taxi Driver, 1976.↵
Citizen Kane, 1941.↵
The Godfather, 1972.↵
Chinatown, 1974.↵
```

```
2001: A Space Odyssey, 1968.↵
Alien, 1979.↵
```

```
Modern Times, 1936.↵
Monty Python and the Holy Grail, 1975.↵
```

11. Stvoriti novu binarnu datoteku `slucajni.bin` i u nju upisati 50 zapisa. Svaki od tih zapisa treba sadržavati slučajno odabrani cijeli broj n (tipa `int`) koji može poprimiti vrijednosti iz intervala $[2, 8]$, te n slučajno odabranih cijelih brojeva (tipa `int`) koji mogu poprimiti vrijednosti iz intervala $[150, 160]$.
12. Napisati program koji će na zaslon ispisati sve zapise datoteke `slucajni.bin` koja je nastala kao rezultat obavljanja programa iz prethodnog zadatka. 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.↵
itd.
```

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 <stdio.h>

```
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;
}
```

4. -

5. #include <stdio.h>

```
int main(void) {
    FILE *ulTok = fopen("brojevi.txt", "r");

    float x, suma = 0.f;
    int brojac = 0;

    while (fscanf(ulTok, "%f", &x) == 1) {
        suma += x;
        ++brojac;
    }

    fclose(ulTok);

    if (brojac > 0) {
        printf("%f\n", suma / brojac);
    } else {
        printf("Nije ucitan niti jedan broj\n");
    }

    return 0;
}
```

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

```
int main(void) {
    int mbr;
    char ime[20 + 1], prez[20 + 1];
    int godina;

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

    while (fscanf(ulTok, "%d%s%s%d.%d.%d",
                  &mbr, ime, prez, &godina) == 4) {
        if (godina == 1987 || strstr(ime, "na") != NULL) {
            printf("%d %s %s\n", mbr, ime, prez);
        }
    }

    fclose(ulTok);

    return 0;
}
```

7. #include <stdio.h>

```
int main(void) {
    FILE *ulTok = fopen("mjerjenja.txt", "r");

    int dan, brojMj;
    while (fscanf(ulTok, "%d%d", &dan, &brojMj) == 2) {
        float temperatura, suma = 0.f;
        for (int i = 0; i < brojMj; ++i) {
            fscanf(ulTok, "%d%f ", &temperatura);
            suma += temperatura;
        }

        printf("Dan=%3d, Broj mjerjenja=%2d, Prosjek=%7.2f\n",
            dan, brojMj, suma / brojMj);
    }
    fclose(ulTok);

    return 0;
}
```

8. #include <stdio.h>

```
#define N 40

int main(void) {
    int a_i_minus2, a_i_minus1 = 1, a_i = 1;

    FILE *tok = fopen("fibonacci.bin", "wb");

    for (int i = 1; i <= N; ++i) {
        if (i > 2) {
            a_i_minus2 = a_i_minus1;
            a_i_minus1 = a_i;
            a_i = a_i_minus1 + a_i_minus2;
        }
        fwrite(&a_i, sizeof(a_i), 1, tok);
    }

    fclose(tok);

    return 0;
}
```

9. #include <stdio.h>

```
int main(void) {
    int n, fibClan;

    printf("Upisite n > ");
    scanf("%d", &n);

    FILE *tok = fopen("fibonacci.bin", "rb");

    for (int i = 1; i <= n; ++i) {
        fread(&fibClan, sizeof(fibClan), 1, tok);
        printf("%d\n", fibClan);
    }

    fclose(tok);

    return 0;
}
```

10. #include <stdio.h>

```
#include <string.h>

int main(void) {
    char naslov[60 + 1], zanr[8 + 1];
    int godina;

    FILE *ulTok = fopen("filmovi.txt", "r");
    FILE *dramaTok = fopen("drama.txt", "w");
    FILE *sfTok = fopen("SF.txt", "w");
    FILE *komedijsaTok = fopen("komedijsa.txt", "w");
    FILE *izlTok = NULL;

    while (fscanf(ulTok, " %[^|]|%[^|]|%d",
                  naslov, zanr, &godina) == 3) {
        if (strcmp(zanr, "drama") == 0) {
            izlTok = dramaTok;
        } else if (strcmp(zanr, "SF") == 0) {
            izlTok = sfTok;
        } else {
            izlTok = komedijsaTok;
        }
        fprintf(izlTok, "%s, %d\n", naslov, godina);
    }

    fclose(ulTok);
    fclose(dramaTok);
    fclose(sfTok);
    fclose(komedijsaTok);

    return 0;
}
```

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

#define MIN_N 2
#define MAX_N 8
#define DG 150
#define GG 160
#define BROJ_ZAPISA 50

int main(void) {
    FILE *izTok = fopen("slucajni.bin", "wb");
    srand ((unsigned)time(NULL));

    int n, polje[MAX_N];
    for (int i = 1; i <= BROJ_ZAPISA; ++i) {
        // odaberi i zapisi slucajan n
        n = rand() % (MAX_N - MIN_N + 1) + MIN_N;
        fwrite(&n, sizeof(n), 1, izTok);

        // odaberi i zapisi n slucajnih brojeva
        for (int j = 0; j < n; ++j) {
            polje[j] = rand() % (GG - DG + 1) + DG;
        }
        fwrite(polje, sizeof(int), n, izTok);
    }

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

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

#define MAX_N 8

int main(void) {
    FILE *ulTok = fopen("slucajni.bin", "rb");

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

    fclose (ulTok);

    return 0;
}
```