

## 22. Dodatne vježbe

1. Testirajte sve primjere s predavanja u vezi funkcije printf.
2. Što će se ispisati sljedećim programom:

```
#include <stdio.h>

int main() {
    printf("|%d|\n", 10);
    printf("|% d|\n", 10);
    printf("|%+d|\n", 10);
    printf("|%d|\n", -10);
    printf("|% d|\n", -10);
    printf("|%+d|\n", -10);

    printf("|%5d|\n", 10);
    printf("|%-5d|\n", 10);

    printf("|%s|\n", "Ana");
    printf("|%5s|\n", "Ana");
    printf("|%-5s|\n", "Ana");

    printf("|%0d|\n", 10);
    printf("|%05d|\n", 10);

    printf("|%x|\n", 27);
    printf("|%#x|\n", 27);

    printf("|%o|\n", 9);
    printf("|%#o|\n", 9);

    return 0;
}
```

3. Napisati program koji će s tipkovnice pročitati cijeli broj broj n ( $1 \leq n \leq 15$ ). Program treba na zaslon ispisati "tablicu množenja". Npr. za zadani n=12, ispis na zaslon treba izgledati ovako:

**Tablica množenja 12x12**

|     | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10  | 11  | 12  |
|-----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 1!  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10  | 11  | 12  |
| 2!  | 2  | 4  | 6  | 8  | 10 | 12 | 14 | 16 | 18  | 20  | 22  | 24  |
| 3!  | 3  | 6  | 9  | 12 | 15 | 18 | 21 | 24 | 27  | 30  | 33  | 36  |
| 4!  | 4  | 8  | 12 | 16 | 20 | 24 | 28 | 32 | 36  | 40  | 44  | 48  |
| 5!  | 5  | 10 | 15 | 20 | 25 | 30 | 35 | 40 | 45  | 50  | 55  | 60  |
| 6!  | 6  | 12 | 18 | 24 | 30 | 36 | 42 | 48 | 54  | 60  | 66  | 72  |
| 7!  | 7  | 14 | 21 | 28 | 35 | 42 | 49 | 56 | 63  | 70  | 77  | 84  |
| 8!  | 8  | 16 | 24 | 32 | 40 | 48 | 56 | 64 | 72  | 80  | 88  | 96  |
| 9!  | 9  | 18 | 27 | 36 | 45 | 54 | 63 | 72 | 81  | 90  | 99  | 108 |
| 10! | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90  | 100 | 110 | 120 |
| 11! | 11 | 22 | 33 | 44 | 55 | 66 | 77 | 88 | 99  | 110 | 121 | 132 |
| 12! | 12 | 24 | 36 | 48 | 60 | 72 | 84 | 96 | 108 | 120 | 132 | 144 |

Npr. za zadani n=1, ispis na zaslon treba izgledati ovako:

**Tablica množenja 1x1**

| 1  |
|----|
| 1! |
| 1  |

4. Preko tipkovnice se unose matični brojevi i datumi rođenja osoba. Datum rođenja sadrži redni broj dana, redni broj mjeseca i redni broj godine. Npr. sadržaj ulaza može izgledati ovako:

```
101 15.5.1989
1526 17.6.1987
17 9.12.1987
0
```

Za svaku osobu pomoću funkcije `scanf` učitati matični broj, redni broj dana, redni broj mjeseca i redni broj godine. Učitavanje se prekida kada se za matični broj upiše 0. Nakon završenog učitavanja, ispisati matične brojeve i godine rođenja osoba. Za prikazani primjer ulaza, izlaz treba izgledati ovako:

```
101. 1989.
1526. 1987.
17. 1987.
```

Može se pretpostaviti da će korisnik utipkati podatke za najviše 100 osoba.

5. Napisati program koji će sadržaj formatirane datoteke `ulaz.txt`, koja se nalazi u direktoriju `c:\tmp`, 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 jednu formatiranu datoteku, te testirati program.
6. Napisati funkciju `broji` koja kao argument prima niz znakova koji predstavlja **ime** formatirane datoteke. 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 ćete testirati rad funkcije.
7. Slično kao u prethodnom zadatku. Funkcija `broji` kao argument prima tok podataka (tip podatka `*FILE`). Glavni program otvara tok podataka, predaje ga funkciji, a kad funkcija obavi brojanje glavni program ispisuje rezultat i zatvara tok podataka.

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

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

### Rješenje 3. zadatka

```
#include <stdio.h>

int main() {
    int n, i, j;
    do {
        printf("Upisite vrijednost za n: ");
        scanf ("%d",&n);
    } while (n < 1 || n > 15 );

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

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

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

    printf("----+", i);
    for (i = 1; i <= n; i++) {
        printf("----", i);
    }
    printf("\n");
    return 0;
}
```

### Rješenje 4. zadatka

```
#include <stdio.h>
#define MAX 100

int main() {
    int dan, mjesec;
    int matBroj, matBrojevi[MAX], godine[MAX];
    int brojUcitanih = 0, i;
    char tocka;
    do {
        scanf("%d", &matBroj);
        if (matBroj != 0) {
            matBrojevi[brojUcitanih] = matBroj;
            scanf("%d%c%d%c%d", &dan, &tocka, &mjesec, &tocka, &godine[brojUcitanih]);
            brojUcitanih++;
        }
        else
            break;
    } while (1);
    for (i = 0; i < brojUcitanih; i++)
        printf("%4d. %4d.\n", matBrojevi[i], godine[i]);
    printf("\n");
    return 0;
}
```

## Rješenje 5. zadatka

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

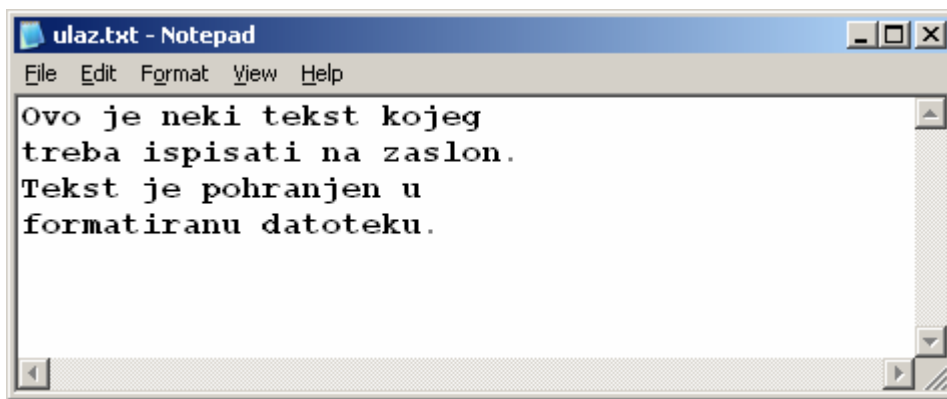
int main () {
    int c;
    FILE *tokPod;
    /* uocite \\ znakove */
    char *imeDat = "c:\\tmp\\ulaz.txt";
    tokPod = fopen(imeDat, "r");

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

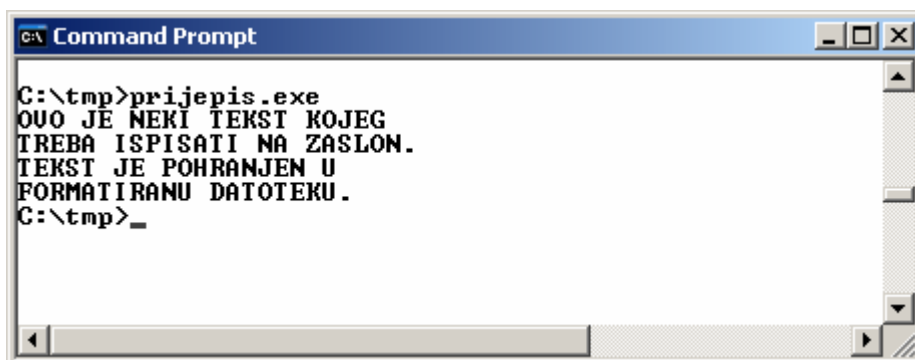
    while ((c = fgetc(tokPod)) != EOF)
        putchar(toupper(c)); /* ili fputc(toupper(c), stdout); */

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

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



Izvođenjem programa dobije se:



## Rješenje 6. zadatka

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

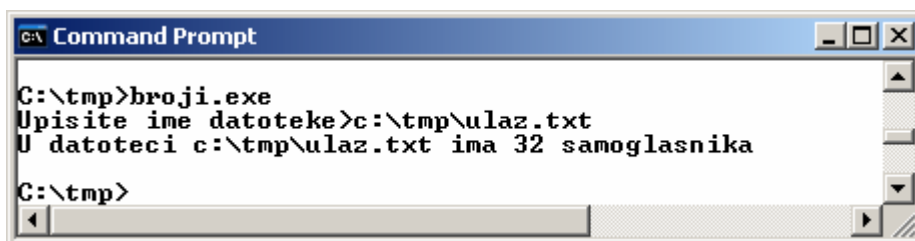
#define MAXIMEDAT 64

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:\> Command Prompt

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

C:\tmp>
```

## Rješenje 7. 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;
}
```