



Универзитет „Св. Кирил и Методиј“ - Скопје
**ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ
И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО**

Аудиторски вежби 10

Датотеки

Структурно програмирање

Потсетување од предавања

Отворање датотеки за читање/запишување

- Обработка на датотеки се состои од запишување, вчитување или менување содржина од на датотека од некој стандарден медиум како диск.
- Обработката на датотеките во програмскиот јазик C се прави со помош на структурата FILE дефинирана во `stdio.h`
- За да се започне со обработка на датотеката прво мора да се отвори со помош на функцијата `fopen` која како резултат враќа покажувач кон структура (FILE*)

Функција за отворање датотека

```
FILE *fopen(const char *filename, const char *mode);
```

`filename` - целосна патека заедно со името на датотеката која сакаме да ја отвориме

`mode` - начин на отворање на датотеката

Потсетување од предавања

Начини на отворање на датотеки

Начин	Значење
r	Отвора постоечка датотека само за читање
w	Отвора датотека за запишување (датотеката не мора да постои)
a	Отвора датотека за додавање содржина (датотеката не мора да постои)
r+	Отвора датотека за читање и запишување од почетокот на датотеката
w+	Отвора датотека за читање и запишување (ја пребришува содржината на датотеката)
a+	Отвора датотека за читање и запишување (додава на крајот од датотеката ако постои)

Потсетување од предавања

Отворање датотеки за читање/запишување

Пример отворање на датотека

```
FILE *fp = fopen("C:\\test.txt", "r");
```

За да се отвори датотеката во бинарен мод се додава b крајот на аргументот за начинот на отворање пр. ("rb")

По завршување со работа со датотеката таа треба да се затвори со помош на функцијата `fclose`

Пример затворање на датотека

```
fclose(fp);
```

Потсетување од предавања

Читање и запишување од/во датотека

Функции за читање од датотека

```
int fscanf(FILE* fp, "kontrolna niza", arguments_list)
int fgetc(FILE* fp)
```

Функции за запишување во датотека

```
int fprintf(FILE* fp, "kontrolna niza", arguments_list)
int fputc(int c, FILE* fp)
```

Задача 1

Решение 1/2

Да се напише програма која за дадена текстуална датотека "text.txt" ќе го одреди односот на самогласките и согласките.

```
#include <stdio.h>

int is_letter(char c) {
    return (c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z');
}

int is_vowel(char c) {
    c = tolower(c);
    switch (c) {
        case 'a':
        case 'e':
        case 'i':
        case 'o':
        case 'u':
            return 1;
        default:
            return 0;
    }
}
```

Задача 1

Решение 2/2

```
int main() {
    char c;
    int count_consonats = 0, count_vowels = 0;
    FILE *dat;
    // Opening file
    if ((dat = fopen("text.txt", "r")) == NULL) {
        printf("The file %s cannot be open.\n", argv[1]);
        return -1;
    }
    // Reading char by char until the EndOfFile (EOF)
    while ((c = fgetc(dat)) != EOF) {
        if (is_letter(c)) {
            if (is_vowel(c))
                count_vowels++;
            else
                count_consonats++;
        }
    }
    fclose(dat);
    printf("Ratio vowel/consonats: %d/%d = %5.2f\n", count_vowels,
        count_consonats,
        (float) count_vowels / count_consonats);
    return 0;
}
```

Задача 2

Да се напише програма која секој ред од дадена текстуална датотека "input.txt" ќе го копира во излезна датотека "output.txt" така што пред секој ред ќе додаде уште еден ред во кој ќе стои бројот на знаци што ги содржи прочитаниот ред од влезната датотека. Во секој ред може да има најмногу 80 знаци.

Задача 2

Решение 1/2

```
#include <stdio.h>
#define MAX 81

int main(int argc, char *argv[]) {
    char line[MAX], *c;
    FILE *input, *output;
    if ((input = fopen("input.txt", "r")) == NULL) {
        printf("File %s could not be open.\n", argv[1]);
        return -1;
    }
    if ((output = fopen("output.txt", "w")) == NULL) {
        printf("File %s could not be open\n", argv[2]);
        return -1;
    }

    while ((fgets(line, MAX, input)) != NULL) {
        int count = strlen(line);
        fprintf(output, "%d %s", count, line);
    }
    fclose(input);
}
```

Задача 2

Решение 2/2

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

Задача 3

Да се напише програма која ќе ги прочита елементите од една матрица сместени во текстуалната датотека “matrica1.txt”. Во првиот ред од датотеката се запишани бројот на редици и колони на матрицата. Секој елемент од матрицата е реален број запишан во посебен ред во датотеката. Потоа матрицата треба да се транспонира и да се запише во нова датотека “matrica2.txt” на истиот начин.

Задача 3

Решение 1/2

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int main() {
    int i, j, m, n;
    float a[MAX][MAX], b[MAX][MAX];
    FILE *input, *output;
    if ((input = fopen("matrica1.txt", "r")) == NULL) {
        printf("Datotekata matrica1.txt ne se otvora!\n");
        exit(1);
    }
    if (!feof(input))
        fscanf(input, "%d %d", &m, &n);

    if ((m > MAX) || (n > MAX)) {
        printf("Mnogu golema matrica!");
        return (-1);
    }
    for (i = 0; i < m && !feof(input); i++)
        for (j = 0; j < n && !feof(input); j++)
            fscanf(input, "%f", &a[i][j]);
    fclose(input);
}
```

Задача 3

Решение 2/2

```
if (i != m || j != n) {
    printf("Nema dovolno podatoci vo datotekata!");
    return (-1);
}
for (i = 0; i < m; i++)
    for (j = 0; j < n; j++)
        b[j][i] = a[i][j];
if ((output = fopen("matrica2.txt", "w")) == NULL) {
    printf("Datotekata matrica2.txt ne se otvora!\n");
    exit(1);
}
fprintf(output, "%d %d\n", n, m); /* obratno */

for (i = 0; i < n; i++)
    for (j = 0; j < m; j++)
        fprintf(output, "%.2f\n", b[i][j]);
fclose(output);
return (0);
}
```

Задача 4

Дадена е текстуална датотека “KRSPrimer.txt”. Да се напише програма која ќе го отпечати бројот на редови во кои има по најмногу десет самогласки, како и вкупниот број на самогласки во датотеката. Во секој ред има најмногу 80 знаци.

Задача 4

Решение

```
#include <stdio.h>
#include <stdlib.h>
int e_samoglaska(char c) {
    return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
}
int main() {
    int red = 0, vkupno = 0;
    FILE *dat; char c;
    if ((dat = fopen("KRSPrimer.txt", "r")) == NULL) {
        printf("Datotekata KRSPrimer.txt ne se otvora");
        exit(-1);
    }
    int samoglaski = 0;
    while ((c = fgetc(dat)) != EOF) {
        if (e_samoglaska(c)) {
            ++samoglaski;
            ++vkupno;
        }
        if (c == '\n') {
            if (samoglaski > 10) {
                red++;
            }
            samoglaski = 0;
        }
    }
    printf("Vkupno %d reda imaat povekje od 10 samoglaski\n", red);
    printf("Vo datotekata ima vkupno %d samoglaski.\n", vkupno);
    return 0;
}
```

Задача 5

Да се напише програма која за дадена текстуална датотека "zborovi.txt" ќе ги испечати сите зборови во кои се јавуваат повеќе од две исти букви (буквата се јавува три или повеќе пати). Да не се прави разлика меѓу големи и мали букви. На крајот треба да се испечати бројот на зборови што го задоволуваат условот. Сите зборови во датотеката се разделени со нов ред. Максималната должина на зборовите е 20 знаци.

Пример зборови

банана, текстот, Обратното, бинарната, дефинирани,
Копакабана

Задача 5

Решение 1/2

```
#include <stdio.h>
#include <ctype.h>
#include <stdlib.h>
#define WORDLEN 21
#define LINELEN 81
int ima_poveke_od2isti(char *w) {
    char *c;
    int isti;
    while (*w) {
        c = w + 1;
        isti = 1;
        while (*c) {
            if (tolower(*w) == tolower(*c))
                isti++;
            c++;
        }
        if (isti > 2)
            return 1;
        w++;
    }
    return 0;
}
```

Задача 5

Решение 2/2

```
int main() {
    char zbor[WORDLEN];
    FILE *f;
    int brzb = 0;
    if ((f = fopen("zborovi.txt", "r")) == NULL) {
        printf("Datotekata %s ne se otvora.\n", argv[1]);
        return -1;
    }
    while (fgets(zbor, WORDLEN, f)) != NULL) {
        if (ima_poveke_od2isti(zbor)) {
            puts(zbor);
            brzb++;
        }
    }
    printf("\nVkupno %d zborovi.\n", brzb);
    fclose(f);
    return 0;
}
```

Задача 6

Да се напише програма која ќе го испечати бројот на појавувања на збор составен само од цифри (се чита од СВ) во текстуална датотека со име "dat.txt".

Задача 6

Решение 1/2

```
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main() {
    char c;
    int brPojavuvanja = 0;
    FILE *dat;

    if ((dat = fopen("dat.txt", "r")) == NULL) {
        printf("Datotekata %s ne se otvora!\n", argv[1]);
        exit(-1);
    }

    char zbor[50];
    gets(zbor);
    int i = 0, count = 0;
```

Задача 6

Решение 2/2

```
while ((c = fgetc(dat)) != EOF) {
    if (isdigit(c)) {
        if (c != zbor[i++]) {
            if (count == strlen(zbor)) {
                brPojavuvanja++;
            }
            count = 0;
        } else {
            count++;
        }
    } else {
        if (count == strlen(zbor)) {
            brPojavuvanja++;
        }
        count = 0;
    }
}
printf("Zborot %s se pojavuva %d pati vo datotekata\n", zbor,
        brPojavuvanja);
return 0;
}
```

Материјали и прашања

Предавања, аудиториски вежби, соопштенија
courses.finki.ukim.mk

Изворен код на сите примери и задачи
<https://github.com/tdelev/SP/tree/master/latex/src>

Прашања и дискусија
forum.finki.ukim.mk