"Ss. Cyril and Methodius" University in Skopje
FACULTY OF COMPUTER SCIENCE AND ENGINEERING

Lession 10

Files

Structured Programming

Opening files for reading/writing

Remainders from lectures

- Processing files includes writing, reading and changing contents of files to some standard media as disk.
- Processing files in C is done using the struct `FILE` defined in `stdio.h`
- To start processing the file first it must be open using the function `fopen` that returns pointer to the struct (`FILE*`)

Function for opening file

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

`filename` - full path of the file we want to open

`mode` - opening mode

Modes of opening files

Remainders from lectures

Mode	Meaning
`r`	Opens existing file only for reading
`w`	Opens existing file for writing (the file must exist)
`a`	Opens existing file for appending (the file must exist)
`r+`	Opens file for reading and writing at the beginning of the file
`w+`	Opens file for reading and writing (deletes the contents of the file)
`a+`	Opens file for reading and writing (appends at the end of the file if it exists)

Opening files for reading/writing

Remainders from lectures

Example opening file

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

To open in binary mode `b` should be appended at the mode of opening ex. (“`rb`”)

After the end of processing, the file should be closed using the function `fclose`

Example of closing file

```
fclose(fp);
```

Reading and writing from/to file

Remainders from lectures

Functions for reading from file

```
int fscanf(FILE* fp, ‘‘control array’’, arguments_list)
int fgetc(FILE* fp)
```

Functions for writing to file

```
int fprtinf(FILE* fp, ‘‘control array’’, arguments_list)
int fputc(int c, FILE* fp)
```

Problem 1
Solution 1/2

Write a program that for given textual file will find the ratio of vowel/consonants. The name of the file is passed as argument.

```c
#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;
    }
}
```

Problem 1
Solution 2/2

```c
int main(int argc, char *argv[]) {
    char c;
    int count_consonats = 0, count_vowels = 0;
    FILE *dat;
    // We need at least 2 arguments
    if (argc < 2) {
        printf("Usage: %s \"file path\"\n", argv[0]);
        return -1;
    }
    // Opening file
    if ((dat = fopen(argv[1], "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;
}
```

Problem 2

Write a program that will copy each line from given text file in to new output file, but before the line will print the length of the line. The name of the input and output file are given as command line arguments, and if they are missing write correct usage. Each line can have max of 80 characters.

Problem 2
Solution 1/2

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

int main(int argc, char *argv[]) {
    char line[MAX], *c;
    FILE *input, *output;
    if (argc < 3) {
        printf("Usage: %s [input_filename] [output_filename]\n",
               argv[0]);
        return -1;
    }
    if ((input = fopen(argv[1], "r")) == NULL) {
        printf("File %s could not be open.\n", argv[1]);
        return -1;
    }
    if ((output = fopen(argv[2], "w")) == NULL) {
        printf("File %s could not be open\n", argv[2]);
        return -1;
    }
```

Problem 2
Solution 2/2

```
    while ((fgets(line, MAX, input)) != NULL) {
        int count = strlen(line);
        fprintf(output, "%d %s", count, line);
    }
    fclose(input);
    fclose(output);
    return 0;
}
```

Problem 3

Write a program that will read elements of a matrix written in text file with name “matrica.txt”. In the first line of the file are written the number of rows and columns of the matrix. Each element of the matrix is floating point number written in separate line. The transposed matrix write in a new output file “matrica2.txt” using the same format.

Problem 3
Solution 1/2

```c
#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);
```

Problem 3
Solution 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, "%7.2f\n", b[i][j]);
    fclose(output);
    return (0);
}
```

Problem 4

Given a text file "KRSPrimer.txt" write a program that will print the count of lines that have at least 10 vowels, and the total vowels in the file. Each line has max of 80 characters.

Problem 4
Solution

```c
#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;
}
```

Problem 5

Write a program that from a given text file will print the words that have multiple occurance of the same letter (the letter occurs two or more times). Ignore the case of the letters in the comparison. Each word is written in a separate line in the file and the max length of the words is 20 characters. The name of the file is a command line argument.

Example words

banana, text, different, Copacabana, etc. . .

Problem 5
Solution 1/2

```c
#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;
}
```

Problem 5
Solution 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;
}
```

Problem 6

Write a program that will print the count of occurances of a given word composed only from digits in a given textual file. The name of the file and the word to be found are given as a command line arguments. The program should check if all needed arguments are provided, and if not so should print apropriate message.

Problem 6
Solution 1/2

```c
#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
int main(int argc, char **argv) {
    char c;
    int brPojavuvanja = 0;
    FILE *dat;
    if (argc != 3) {
        printf("Nevalidni argumenti na komandna linija\n");
        exit(-1);
    } else {
        if ((dat = fopen(argv[1], "r")) == NULL) {
            printf("Datotekata %s ne se otvora!\n", argv[1]);
            exit(-1);
        }
    }
```

Problem 6
Solution 2/2

```
    char *zbor = argv[2];
    int i = 0, count = 0;
    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;
}
```

Materials and Questions

Lectures, exsercises and announcements
courses.finki.ukim.mk

Source code of all examples and problems
https://github.com/tdelev/SP/tree/master/latex/src

Questions and discussion
forum.finki.ukim.mk