PROGRAMMING TECHNIQUES, A.A. 2023/2024

Laboratory 1

Objectives

- Solving basic I/O problems

Technical contents

- Definition and usage of integer, float, char variables
- I/O functions: *(f)scanf* and *(f)printf*, *(f)gets* and *(f)puts*, *(f)getc/getchar* and *(f)putc*
- Directive *#define*
- Format specifications (*%d*, *%f*, *%c*, *%s*)
- Read data from file/keyboard – Print data on file/video
- Use of cast operators

Solve the following during the laboratory

Exercise 1. Using the IDE create a new project and write/build the following program in C language. Check that there are no errors when you run the program.

```c
#include <stdio.h>
int main(void)
{
    int x, y;
    float z;

    printf("Insert an integer number:");
    scanf("%d", &x);
    y = 3;
    z = (float)(x)/y;

    printf("%d/%d=%.3f\n", x, y, z);
    return 0;
}
```

Try executing the program with different input values from keyboard: *0*, *9*, *15*, *20*. Using the debugger, check the content of the variables *x*, *y* e *z*.

Exercise 2. Using the IDE create a new project and create a text file by the name of "Guide.txt" inside the project folder. The content of the file may be the following:

"Quarantadue!" urlo' Loonquawl. "Questo e' tutto cio' che sai dire dopo un lavoro di sette milioni e mezzo di anni?"
"Ho controllato molto approfonditamente," disse il computer, "e questa e' sicuramente la risposta. Ad essere sinceri, penso che il problema sia che voi non abbiate mai saputo veramente qual e' la domanda."

Write/build the following program in C language. Check that there are no errors when you run the program.

```c
#include <stdio.h>

int main() {
    FILE *fp_read, *fp_write;
    char file_char, choice;

    if ((fp_read = fopen("../Guide.txt", "r")) == NULL) {
        printf("Error opening file\n");
        return 1;
    }
    if ((fp_write = fopen("../Output.txt", "w")) == NULL) {
        printf("Error opening file\n");
        return 2;
    }

    printf("Print on console (C) or on file (F):");
    choice = getchar();

    while (!feof(fp_read)) {
        file_char = fgetc(fp_read);
        if (!feof(fp_read)){
            switch (choice) {
                case 'C':
                    printf("\nChar printed on the console: %c", file_char);
                    break;
                case 'F':
                    fputc(file_char, fp_write);
                    printf("\nChar saved on file: ");
                    putchar(file_char);
                    break;
                default:
                    printf("Wrong choice\n");
                    return 3;
            }
        }
    }

    fclose(fp_read);
    fclose(fp_write);

    return 0;
}
```

After building the program, try to run it and test the different cases.
In depth: What happens if you omit the line *if (!feof(fp_read))*? Why?

Exercise 3. Using the IDE create a new project and create a text file by the name of "Bronte.txt" inside the project folder. The content of the file may be the following:

Ho sognato nella mia vita,
sogni che son rimasti sempre con me,
e che hanno cambiato le mie idee;
son passati attraverso il tempo e attraverso di me,
come il vino attraverso l'acqua,
ed hanno alterato il colore della mia mente.

Write/build the following program in C language. Check that there are no errors when you run the program.

```c
#include <stdio.h>

int main() {
    FILE *fp_read, *fp_write_odd, *fp_write_even;
    char file_string[100], name[20];
    int counter = 0;

    if ((fp_read = fopen("../Bronte.txt", "r")) == NULL) {
        printf("Error opening file\n");
        return 1;
    }
    if ((fp_write_odd = fopen("../Output_odd.txt", "w")) == NULL) {
        printf("Error opening file\n");
        return 2;
    }

    if ((fp_write_even = fopen("../Output_even.txt", "w")) == NULL) {
        printf("Error opening file\n");
        return 3;
    }

    printf("What's your name?");
    gets(name);

    while (!feof(fp_read)) {
        counter++;
        if (counter%2==0) {
            fscanf(fp_read, "%s", file_string);
            if (!feof(fp_read)) {
                printf("%s\nI am reading:\n%s\n\n", name, file_string);
                fprintf(fp_write_even, "%s", file_string);
            }
        }
        else {
            fgets(file_string, 100, fp_read);
            if (!feof(fp_read)) {
                puts(name);
                puts("I am reading:");
                puts(file_string);
                fputs(file_string, fp_write_odd);
            }
        }
    }

    fclose(fp_read);
    fclose(fp_write_even);
    fclose(fp_write_odd);

    return 0;
}
```

What happens when the counter is odd? What happens when the counter is even? Which is the difference between *fgets* and *fscanf*?

Exercise 4. Write a C program to compute either **the area** of a circle or of a square, based on the choice and **inputs of the user**. The user is requested to type a set of characters and numbers from keyboard, in a specific order. The first character should be **Q** for square or **I** for circle. In case it is a square, the user can decide whether the area should be computed given the value of the diagonal or of the side of the square, by specifying its respective length after the character D or S. In case it is a circle, the user can decide whether the area should be computed given the value of the diameter or of the radius by specifying its respective length after the character D or R, respectively. All the values inserted by the user should be integer, the computed area should be a float.

For example, if the user types ***Q D10***, the program should compute the area of a square with diagonal equal to 10 and print the following on the screen:

The area of the square with diagonal 10 is 50.0

Suggestions:

a) Define a constant PI for the value 3.14 with #define. Remember: with #define there are no '=' and no ';'
b) Acquire the input of the user. **Careful:** you need to acquire single characters and numeric values from keyboard...
c) Compute the value of the area (float) based on the acquired input.
d) Print the result on screen.

Remember:
Square: *Area = S*S = D*D/2*, Circle*: Area = pi*R*R = pi*D*D/4*

Exercise 5. Write a "calculator" in C, that performs basic arithmetic operations (addition, subtraction, division and multiplication) on two given inputs ***op1*** and ***op2***.
Write a C program that:

a) Acquires a character from keyboard using *getchar* (either '+', '-', '*' or '/') to decide which should be the arithmetic operation to perform;
b) Acquires two float operands (ex. *21.0* and *2.0)* from keyboard, using *scanf*.
c) Prints the selected operator on the screen, followed by the result of the operation (for example: */ 10.50*).

Careful: What happens if ***op2*** is 0? How do you handle the problem?

Exercise 6. Starting from the code of Exercise 5, write a C program that reads a set of operations from a text file, reported one per line. The program should create a second file, reporting in each line the operator and the result of the corresponding operation in the input file. The result should be written with two digits after the decimal point, as in the example below.

Example

Input file: Operations.txt:

```
+ 15.225 30.51
- 42.1 10.01
* 0.62 2.4
/ 5.0 2.5
```

Output file: Results.txt:

```
+ 45.74
- 32.09
* 1.49
/ 2.00
```