

## EE101 C programming and SW engineering 1

### Lab Practice 10 – Files

Use your preferred compiler to investigate the programming exercises below. This laboratory concerns the use of files and standard input/output functions.

<b>Exercise 1</b>
-------------------

Write a C program to read data from the keyboard, write it to a file called INPUT, again read the same data from the INPUT file, and display it on the screen.

**Hints:** The end of the input data is indicated by entering an EOF character, which is control-Z/control-D in the Windows7/UNIX system. The file is closed at this signal.

**Example: Data Input**

Programming in C is fun (Enter)

^Z (Enter)

**Data Output**

Programming in C is fun

## Exercise 2

Write a C program to double space the new lines in a file.

**Hints:** Find a newline and duplicate it. Example:

### Input File

Peter: Hello, my name is Peter.

Peter: How are you?

Mary: I am fine. I am Mary. Thanks.

Peter: Do you like C programming?

Mary: Yes, I love it.

### Output File

Peter: Hello, my name is Peter.

Peter: How are you?

Mary: I am fine. I am Mary. Thanks.

Peter: Do you like C programming?

Mary: Yes, I love it.

### Exercise 3

Analyse and explain in detail the following C program.

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

int main(){
char fname[MAX];
int c;
FILE *ifp;

printf("Please enter a file name: ");
scanf("%s", fname);
ifp=fopen(fname,"rb");
fseek(ifp, 0, SEEK_END);
fseek(ifp, -1, SEEK_END);
while(ftell(ifp)!=0){
    c= fgetc(ifp);
    putchar(c);
    fseek(ifp,-2,SEEK_CUR);
}
printf("\n");
return 0;
}
```

*/\*open a file in binary mode for reading\*/*  
*/\*move to the end of the file\*/*  
*/\*move back one character\*/*  
*/\*fgetc moves forward one character\*/*  
*/\*back up two characters\*/*

Does the program go wrong? If yes, please explain it.

### Exercise 4

Write a C program first to ask user to create a file named DATA which consists of a list of integers (e.g. 1 2 3 5 78 56 ...). Then the program reads the numbers from the created file DATA and displays all odd numbers together (e.g. displays them in the same row) followed by all even numbers together.

**Hints:** You might try to use the functions **getw** and **putw** which are integer-oriented functions. They are similar to the **fgetc** and **fputc** functions that are used to read and write character values. These functions would be useful for dealing with only integer data (also for this exercise). The syntax of **fgetc** and **fputc** is as follows: **putw(integer,filepointer)** and **getw(filepointer)**.