Exercise 15 – Input and Output

Objective

The major objective is to practice C input and output.

Reference Material

This is based entirely on the *Input and Output* chapter. The practical session is located in the following directory:

Windows Directory:	**c:\qacprg\inout**
Windows Solution directory:	**c:\qacprg\inout\solution**
Linux Directory:	**/home/user1/qacprg/INOUT**
Linux Solution directory:	**/home/user1/qacprg/INOUT/Solution**

Overview

The first two questions are file-input exercises. Question 3 asks you to accept command-line arguments in your program. Questions 4 and 5 ask you to write a file-copy program, using `fread` and `fwrite` instead of `fgetc` and `fputc` (c.f. **mycopy.c** in the notes).

Practical Outline

1. Open the Visual Studio Solution **lines.sln**. Write a program that asks for the name of a file, reads this file and using `fgetc()` (or `getc()`) reports how many lines there are.

   Hint: count the number of `'\n'` characters in the file.

   When you are satisfied with your program, re-implement a solution which uses `fgets()`.

   A solution for this question is available in the Visual Studio Solution **solution\lines1.sln**.

2. Working in the same Visual Studio Solution, modify the program (choose either technique – using `fgetc()`/`fgets()` as appropriate) so that it displays the first 23 lines on the screen, and then waits for the user to press a key before displaying the next 23, etc. (very similar to the `more` command.)

Hint:

Use the library function `getch` (*beware - it is not standard!*) to read an unbuffered character from the keyboard. You will have to include the `conio.h` header.

A solution for this question is available in the Visual Studio Solution **solution\lines2.sln**.

3. Still in the same Visual Studio Solution, change your program written so that it can be given the name of the file to process as a command-line argument.

   A solution for this question is available in the Visual Studio Solution **solution\lines3.sln**.

Optional:

4. Open the Visual Studio Solution **filecpy.sln**, and take a look at the code template provided in **filecpy.sln**. The program asks the user for two file names; extend the program so that it copies the contents from one file to the other. Use the `fread` and `fwrite` functions to do the file reading and writing. For guidence you can refer to the example of **mycopy.c** or **myfcopy.c** in the notes.

   A solution for this question is available in the Visual Studio Solution **solution\filecpy1.sln**.

5. In the same Visual Studio Solution, change the `main` function so that both filenames are taken in as command-line arguments.

   A solution for this question is available in the Visual Studio Solution **solution\filecpy2.sln**.