University of Nottingham
UK | CHINA | MALAYSIA

EEEE1042 - **Coursework 1**

Total marks: 100

In this coursework assessment you are to create `.c` programs that accomplish the tasks described in the problem statement below. This coursework assessment is made available to the students from Moodle on Thursday Nov 18th and is due back uploaded on Moodle by 5pm, Wednesday Nov 24th. Please submit your programs on Moodle by the due date. This assessment is worth 10% of your final grade.

Marking Rubric

- 0-25%: Codes developed but did not compile properly.
- 26-50%: Codes developed and compiled, but not achieving expected result, programs are buggy or inefficient.
- 51-75%: Codes developed and compiled with correct bug-free and efficient results but modularity/clarity/commenting practices are not well followed.
- 76-100%: Codes developed and compiled using taught best practice standards, with clear, well-commented code and modularity rules being demonstrated/followed.

The codes you submit must be developed by yourself. Any codes which are **significantly** similar to each other, or to code found on the internet will be penalized as plagiarism. The comments in your code will be contributing evidence that the codes submitted are your own.

Problem statement

```
1.0     12.0
2.0     15.2
3.0     17.8
4.0     19.1
5.0     15.3
6.0     13.1
7.0     14.3
```

Figure 1: An example of the structure of the file `input.txt` that your program is to read in.

1. You are to develop a C program that reads a filename supplied by the user. Inside the file are two columns of numbers, an example of which is depicted in Figure 1. The name of the input file is a command-line parameter that your program must read in. Your program is to complete the following tasks:

   (a) Read the name of the input file from the command line. You may choose any appropriate name that you wish for the input file.

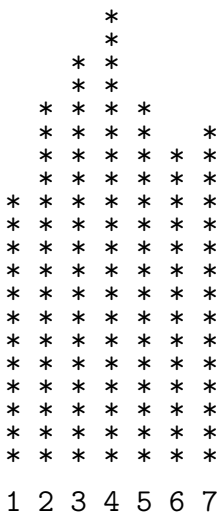
   (b) Open the input file for reading (checking for its existence, and exiting gracefully if it doesn't exist) and read the file's contents into memory. Use the data in Figure 1 as your test input data.

   (c) The two numbers from the input file represent a histogram where the first number is the bin-label, which could, for example, represent the day of the week, and the second number is a measure of some quantity we are interested in. Create a function that uses the data read from the input file to plot a histogram with labels. The bars of the histogram extend horizontally with the length of the bars proportionate to the number associated with each label. Such as:

```
1.0/12.00  ************************
2.0/15.20  ******************************
3.0/17.80  ************************************
4.0/19.10  ****************************************
5.0/15.30  ******************************
6.0/13.10  *************************
7.0/14.30  ******************************
```

Save your program to the file: lastName_firstName_assessment1_EEEE1042.1.c and submit it to Moodle by the due date. For full marks be sure to adequately comment your code. You may also submit your input file with which your program was tested and a make file showing how you call the program, reading in the input file. In the absence of a submitted input file I will be testing your program with my own input file.

2. Histograms usually run vertically rather than horizontally. Create a subfunction that allows you to plot the histogram vertically such that the output looks more like this:



Save your program to the file: lastName_firstName_assessment1_EEEE1042.2.c and submit it to Moodle by the due date. For full marks be sure to adequately comment your code.

3. In the third part of this assignment, you are to modify your work from the previous part as follows: Instead of reading the histogram directly from the input file, you are to **calculate** the histogram from a text document.

   (a) Modify your program to take in the name of a text file as input,

   (b) Open the text file and read in each word from the input file **computing/counting** the histogram based on the length of each word.

   - The histogram is computed/counted by measuring the length of each read word in the text file and incrementing a frequency bin for the word of that particular length.

   (c) Pass the **computed** histogram data to your previously written subroutine for plotting. Ensure your histogram can work with the subroutines for both horizontal and vertical printing.

   Save your program to the file: `lastName_firstName_assessment1_EEEE1042.3.c` and submit it to Moodle by the due date. For full marks be sure to adequately comment your code. You may also submit your input file with which your program was tested and a make file showing how you call the program, reading in the input file. In the absence of a submitted input file I will be testing your program with my own input file.