

## COMP1711 Worksheet 6 – Input / Output

### General points

Resources can be downloaded as a zip file with this worksheet.

Exercise 3 can be submitted to Gradescope to test the autograding functionality.

### 1. Command line arguments

Write a program `echoes.c` that prints out (echoes) all command line arguments as a list, line by line.

For example if I executed the program as

```
./echoes 1 hello -3.0 class
```

the program output would be

```
./echoes
1
hello
-3.0
class
```

### 2. File I/O

Write a program, `fileswap.c`, to read in a text file and write it out to a second file in transposed form.

The input data is integer and in a fixed format of rows and columns. You can assume the file is correctly formatted.

The program accepts 4 command line arguments:

- Input file name (string)
- Number of rows in input (nrow) (integer)
- Number of columns in input (ncol) (integer)
- Output file name (string)

Dynamically allocate an array to store the data that is read in.

The output should have ncol rows and nrow columns such that the  $k^{\text{th}}$  row of the input is the  $k^{\text{th}}$  column of the input.

For example an input file:

```
1 2 3
4 5 6
```

would produce an output file:

```
1 4
2 5
3 6
```

### Tests

A file of data [mat.in](#) is provided with integer numerical data with 10 rows and 4 columns

You can execute your program with

```
./fileswap mat.in 10 4 mat.out
```

### 3. Roots of a quadratic equation (Gradescope assessment)

In worksheet 3 you wrote a program that computed the roots of a given quadratic.

The coefficients (a,b,c) were defined in your code, so that to change the equation you had to recompile the code.

Now we will extend the code to a new version, [quadratic3.c](#), and accept values for a,b,c as command line arguments.

Coefficients are expected to be floating point values.

You can submit your solution to Gradescope for autograding to test that functionality.

### Hints

Begin from the skeleton C program, [quadratic3.c](#).

The majority of your code can be reused from the [quadratic2.c](#) solution.

The print statements should not be altered – the autograder expects them to be used exactly as is.

Remember to test compilation

```
gcc quadratic3.c -o quadratic3 -lm
```

and execution of your code before submission. For example you would now type

```
./quadratic3 2.0 3.0 4.0
```

to test the first case below.

### Tests

Test your program for the cases from the previous sheet and the following cases:

- $2x^2 + 3x + 4 = 0$  (no roots)
- $x^2 + 1.5x - 2.5 = 0$  (roots 1, -2.5)
- $x^2 - 3x + 2.25 = 0$  (root 1.5)

**Grading**

Submit your work to Gradescope using from the Minerva module site at

Assessment and Feedback -> Submit My Work -> Submit your programming assignments

This will open the Gradescope site for this module.