CS319 Class Test: 23 Feb 2024

Instructions:

- **Answer all three questions**.
- Your solution of each question should be in the form of a C++ program.
  Email it to niall.madden@universityofgalway.ie You can send a single file, or one file per question (as you prefer). Each of your files should include comments with your name, ID number, and email address.
- This is an "open book" test: you can use your lecture notes, and any other resource at
  `https://www.niallmadden.ie/2324-CS319`
- You may not communicate with anyone during the test.

.....

Q1 Here is part of a C++ program that

(a) assigns random **int**eger values to variables **a**, **b**, and **c**,

(b) outputs their values,

(c) sorts them into increasing order

(d) outputs the sorted values values.

The code provides does Step (a), but not steps (b), (c) and (d).

```
#include <iostream>
#include <cstdlib> //needed for the rand()

int main(void)
{
  int a=rand()%12, b=rand()%10, c=rand()%8;
  // (b) Your code for outputting a, b, and c goes here.

  // (c) Your code for sorting a, b, and c goes here.

  // (d) Your code to output sorted a, b, and c goes here.
  return(0);
}
```

You can also download it from
`niallmadden.ie/2324-CS319/ClassTest/Q1-part.cpp`.

Task: Add the missing code that completes steps (b), (c) and (d).

The goal of Q1 is to verify that you can output data, and work with **if** *statements.*

.....

Q2 For this question, it helps to know that

- **float a[10];**
  creates an array of 10 floats called $a[0]$, $a[1]$, ..., $a[9]$.
- **rand()** returns an "random" **int** between 0 and RAND_MAX;

Write a program that works as follows.

(a) the program has a function with header
float SumArray(float list[], int len); that returns the sum of the entries in the array **list[]** which is of length **len**

(b) the program has a function with header
float SumArray(float list[], int len); that scales the entries in the array **list[]**, which is of length **len**, so that they sum to 1.

(c) The **main()** part of the program:

i. uses a **for** loop to create an array of ten **float**s, and sets the entries to be a random number between 0 and 1. (Tip: you can divide the value returned by **rand()** by RAND_MAX. Remember to recast both as **float**s.

ii. It then outputs the entries in this array.

iii. It next uses the **Normalise()** function to re-scale the entires in the array so that they sum to 1.

iv. It outputs the entries in this normalised array.

The goal of Q2 is to verify that you are competent writing **for***-loops and functions.*

.....

Q3 (a) Write a program that has a function with header
int RandInterval(int a, int b);
that takes a two integer arguments, **a** and **b**, and returns a random number (generated by **rand()**) that is between a and b, inclusive. Important: either a or b can be positive or negative, and the function must be capable of returning either of these.

(b) In your **main()** function, verify that **RandInterval()** works by

- Prompting the user to enter values of a and b;
- Check that $a < b$. If not, prompt the user for new values.
- Outputs a random number between a and b.

The purpose of Q3 is to verify that you can read input, write functions, and use **if** *(or* **while***) statements.*