MTRN2500 C++ Exercise 1

Objectives

- Understand basic C++ program structure

- Understand and practice basic input and output

- Use std::array instead of C-array to improve reliability

- Use range-based for loop instead of traditional for loop to improve readability

- Use pass-by-reference instead of pass-by-value/pass-by-pointer to improve efficiency and safety

- Use auto to improve easiness of coding

- Use const to improve reliability

- Understand and practice function overloading

Tasks

This week's exercise is to write a program that will read in five integers, print them out, give the maximum and minimum of the five integers, and then repeat the same task for five floating-point numbers.

You need to use the code template provided at the bottom of this document (or copy from the link).

Basic process of your program should be

Stage 1. Process five integer numbers

Step a. Read five integer numbers input by user into a std::array<int, 5> intArray (you can assume the numbers are entered correctly)

example: your terminal should look like:

```
1 - Please input five integers:
4 7 2 15 -8
```

Step b. Print the elements of the array out

example: your terminal now should look like:

```
1 - Please input five integers:
4 7 2 15 -8
The five integers you have input were:
4 7 2 15 -8
```

Step c. Find the maximum and minimum values and print them out

example: your terminal now should look like:

```
1 - Please input five integers:
4 7 2 15 -8
The five integers you have input were:
4 7 2 15 -8
The maximum integer was: 15
The minimum integer was: -8
```

Stage 2. Process five decimal numbers

Step a. Read five decimal numbers input by user into a std::array<double, 5> doubleArray (you can assume the numbers are entered correctly)

example: your terminal now should look like:

```
1 - Please input five integers:
4 7 2 15 -8
The five integers you have input were:
4 7 2 15 -8
The maximum integer was: 15
The minimum integer was: -8
2 - Please input five decimals:
3.5 4.8 -2.59 9.76 13.1
```

Step b. Print the elements of the array out

example: your terminal now should look like:

```
1 - Please input five integers:
4 7 2 15 -8
The five integers you have input were:
4 7 2 15 -8
The maximum integer was: 15
The minimum integer was: -8
2 - Please input five decimals:
3.5 4.8 -2.59 9.76 13.1
The five decimals you have input were:
3.5 4.8 -2.59 9.76 13.1
```

Step c. Find the maximum and minimum values and print them out

example: your terminal now should look like:

```
1 - Please input five integers:
4 7 2 15 -8
The five integers you have input were:
4 7 2 15 -8
The maximum integer was: 15
The minimum integer was: -8
2 - Please input five decimals:
3.5 4.8 -2.59 9.76 13.1
The five decimals you have input were:
3.5 4.8 -2.59 9.76 13.1
The maximum decimal was: 13.1
The minimum decimal was: -2.59
```

Specification

1. The template that you should work on can be found via https://onlinegdb.com/SkD9-dhVw (you can fork it to edit and download the file or copy the code to a local file)

2. You should not make any changes to the main function

3. You should implement the other functions to make the program work

4. Try using a range-based for loop (rather than a traditional for loop) when reading numbers into the array and examining the elements stored in the array

5. Try using const, auto and reference (&) wherever possible and appropriate, and think about why if some of them did not work in places you have tried

6. Try to understand how the function overloading works

7. Follow the MTRN2500 style guide as much as possible

8. You can assume the inputs are always in correct format (the same as in the example) - no need to check the correctness of the inputs in the code

9. Its ok to have a trailing space character when printing the array

Hints

1. You can comment out the stage 2 part and work on stage 1 first

2. When working on a stage, you can comment out the dependent functions out and work on the independent ones first

3. If you are not familiar with range-based for loops, use traditional for loops first and modify them later

4. If you are not familiar with const/auto/reference, skip using them at the beginning and try adding them later

Submission

1. Submit a cpp file named Exercise1_z1234567.cpp to Moodle where z1234567 is your zID

2. Submission deadline: 17:00 21st Sep 2020

3. No late submission is accepted

Assessment

1. A tutor will mark your submission by running your submitted program without making any modifications (no debugging from the tutor should be assumed)

2. If the program runs correctly, 1 mark will be given (you must use range-based for loop, const, auto, and reference at least once each to get full marks)

3. If the program runs partially correctly, 0.5 marks will be given (e.g., results were all correct but range-based for loop, const, auto, and reference were not all used at least once, or results were not fully correct)

4. If the program runs completely wrong (e.g., no clear effort in addressing the problem or irrelevant submission), 0 marks will be given

5. Feedback on the coding style will also be given but not reflected in the marks

Code Template

Can also be found at: https://onlinegdb.com/SkD9-dhVw

```
// File:        Exercise1_z1234567.cpp
// Description: Exercise 1, MTRN2500 20T3
// Autor:       XX XX
// zID:         z1234567
// Date:        XX/XX/2020


// ------------------------general instructions--------------------------------
// implement all the functions except the main() to make this program work;
// you can modify both the parameters and the body of the functions;
// try using range_based for loops;
// try using const, auto and reference (&) wherever possible and appropriate
// and think about why if some of them did not work in places you have tried
// ----------------------------------------------------------------------------

#include <iostream>
#include <array>

// ----------------------------stage 1-----------------------------------------
// TODO: modify readFiveIntegersFromCin function using range-based for loop
// description: read five integers from std::cin;
// return:      a std::array storing five integers user entered
std::array<int, 5> readFiveIntegersFromCin (void) {
    // create an int array with five elements using std::array
    // and initialise the elements with zeros
    std::array<int, 5> inputs {};
```

```cpp
    // TODO: replace the following for loop with a range-based for loop
    for (unsigned int i = 0; i < 5; ++i) {
        std::cin >> inputs[i];
    }
    return inputs;
}

// TODO: implement printArrayToCout function for intArray;
// description: print five integers to std::cout
//              printing extra spaces is fine
void printArrayToCout (std::array<int, 5> intArray) { // can be modified
    // your code here...


}

// TODO: implement findMaximumValue function for intArray;
// description: find the maximum value in the array;
// return:      the found value
int findMaximumValue (std::array<int, 5> intArray) { // can be modified
    // your code here...
}

// TODO: implement findMinimumValue function for intArray;
// description: find the minimum value in the array;
// return:      the found value
int findMinimumValue (std::array<int, 5> intArray) { // can be modified
    // your code here...
}

// TODO: implement printMaximumToCout function for intArray;
// description: print the maximum number in the array to std::cout
void printMaximumToCout (std::array<int, 5> intArray) { // can be modified
    int maximum = findMaximumValue (intArray); // can be modified
    // your code here...
}

// TODO: implement printMinimumToCout function for intArray;
// description: print the minimum number in the array to std::cout
void printMinimumToCout (std::array<int, 5> intArray) {
    int minimum = findMinimumValue (intArray); // can be modified
    // your code here...
}

// ----------------------------stage 2----------------------------------------
// TODO: implement readFiveDoublesFromCin function;
// description: read five doubles from std::cin;
// return:      a std::array storing five doubles user entered
std::array<double, 5> readFiveDoublesFromCin (void) {
    // your code here...
}

// TODO: implement printArrayToCout function for doubleArray;
// description: print five doubles to std::cout
//              printing extra spaces is fine
```

```cpp
void printArrayToCout (...) { // can be modified
    // your code here...
}

// TODO: implement findMaximumValue function for doubleArray;
// description: find the maximum value in the array;
// return:      the found value
double findMaximumValue (...) { // can be modified
    // your code here...
}

// TODO: implement findMinimumValue function for doubleArray;
// description: find the minimum value in the array;
// return:      the found value
double findMinimumValue (...) { // can be modified
    // your code here...
}

// TODO: implement printMaximumToCout function for doubleArray;
// description: print the maximum number in the array to std::cout
void printMaximumToCout (...) { // can be modified
    // your code here...
}

// TODO: implement printMinimumToCout function for doubleArray;
// description: print the minimum number in the array to std::cout
void printMinimumToCout (...) { // can be modified
    // your code here...
}

// ---------------------------main function---------------------------------
// you should not make changes to the main function;
// instead, implement the other functions to make this program work
int main() {

    // Stage 1
    // ask user to input five integer numbers and store them in an int array
    std::cout << "1 - Please input five integers:\n";
    std::array<int, 5> intArray = readFiveIntegersFromCin();

    // print all the elements of the int array to std::cout
    printArrayToCout(intArray);

    // print the maximum value to std::cout
    printMaximumToCout(intArray);
    // print the minimum value to std::cout
    printMinimumToCout(intArray);

    // Stage 2
    // ask user to input five decimal numbers and store them in a double array
    std::cout << "2 - Please input five decimals:\n";
    std::array<double, 5> doubleArray = readFiveDoublesFromCin();

    // print all the elements of the double array to std::cout
```

```
    printArrayToCout(doubleArray);

    // print the maximum value to std::cout
    printMaximumToCout (doubleArray);
    // print the minimum value to std::cout
    printMinimumToCout(doubleArray);

    return 0;
}
```