1. What was the most useful technique you used to find the bugs? Why was it more useful than other techniques you tried?

There are many techniques that I used for this code but some of them I found very useful to find bugs in this code and out of which I like the most is the unit testing because Unit tests help detect bugs early in the development process, which makes it easier and less time-consuming to fix them. Also Writing unit tests helps ensure that the code works as intended, and that it meets the desired specifications. This leads to higher code quality and more reliable software.

1. Look up answers to the following questions and report your findings:
2. What is the largest integer and double values you can store?
3. Why is there a limit on the maximum value you can store in a variable?
4. If you exceed the maximum value an integer can hold, what happens? Explain why the format causes this to happen.
5. What is the format for the storage of a floating-point variable? How does this differ from the way an integer is stored?
6. The data type that is used to store the integer determines the largest integer value that can be stored in a variable. On most systems, a "int" data type may store a value up to 2,147,483,647. 9,223,372,036,854,775,807 is the maximum value that can be stored in a "long long int" data type. On most computers, the highest value that can be stored in a "double" data type, which is used to store floating-point values, is roughly 1.8 x 10308. However, it's important to note that the exact maximum value may vary depending on the programming language and system architecture.
7. There is a limit on the max value to be stored in a variable because of the finite space and memory constraints. In a computer’s memory there is a fixed amount of space for different data types. The limit on the maximum value is determined by the number of bits allocated to represent the data type.
8. An overflow error happens when the value of an integer is greater than it can handle. An integer variable's format dictates that a set number of bits are used to hold the value. An overflow occurs and the value wraps around to the minimum value that can be represented with the specified number of bits if the value stored in the variable is greater than the highest value that can be expressed with that number of bits. F or example, in a 32-bit signed integer, if you add 1 to the maximum value (2,147,483,647), it will wrap around to the minimum value (-2,147,483,648).
9. Floating-point variables, such as "double" in many programming languages, are typically stored in a format defined by the IEEE 754 standard. The decimal point is represented by a binary exponent in the fixed-point format that floating-point numbers are stored in. This format consists of three parts: the sign bit, the exponent, and the mantissa (also called the fraction). Unlike integers, floating-point numbers use a binary scientific notation, where the value is represented as a fraction times a power of 2. For the mantissa and the exponent, floating-point numbers have a variable amount of bits, unlike integers, which are kept in a binary representation. The precision is sacrificed in order to store a considerably wider range of values.
10. What is the default amount of stack memory that is given to a program when Visual Studio starts a C or C++ program? What is the default heap size? Did you hit any of the limits? If so, which one(s)? If you hit a limit, would increasing the amount of memory allocated to the program fix the problem? Justify your answer. Why do they limit the stack and heap size for a program?

The default heap and stack memory sizes might vary depending on the operating system and language. Although the problem might not always be resolved by simply increasing memory allocation, both the stack and the heap have memory limits that can be reached. Memory limitations prevent programs from consuming excessive amounts of memory, which could harm other programs or cause system crashes. The restrictions allow the system to function efficiently and dependably. The default stack size for a program started in Visual Studio can be around 1 MB per thread.