To run the code, you must put the cs303\_assignment1.h, cs303\_assignment1.cpp, and cs303\_assignment1\_main.cpp files into a directory with the integers.txt file.

When you run the code, the following assumptions are made:

The .txt file must be ‘integers.txt’, I did not ask the user to enter the filename of the .txt file.

You can use another file full of integers, but it must be named ‘integers.txt’.  
The integers.txt file has 100 integers in it, 10 lines with 10 integers per line.

The integers.txt file only contains integers and they are separated by spaces.

The integers.txt file does not contain any integers with the value 0.

The user cannot add to the array more than 50 times without removing values or the array will be full (If you do this it will print an error message that the array is full).

The input is only validated for the two functions that are specified as needing input validation in the assignment (add/modify), so if you try to break the program, you can.

Because I created a menu and commands for the functions to be called multiple times, there were a lot of instances where I needed to validate input, but because this aspect of the assignment was never required, I restricted myself to only validating input using try/catch blocks for the two functions that were required as stated on the assignment.

There is some additional input validation in places otherwise but once again if you try to break the program I am sure it can be done quite easily.

When you run the program, it will bring up a menu of commands. You must type in a capital letter referring to the command you are trying to use (I didn’t use toupper()), otherwise the program will say your input is invalid and prompt you to enter a command again.

There’s a menu, and a while loop to perform the functions on the array multiple times.

After finishing using the functions, you can print the array with “P” and then use command “X” to exit the program.

For my remove function, I remove the value at the index and slide all of the values down one index and then make the last index 0. I did this instead of making values 0 because my add function relies on finding the first 0 position to find the “end” of the array.

Screenshot of the program running below:

A screenshot of a computer program

Description automatically generated with medium confidence