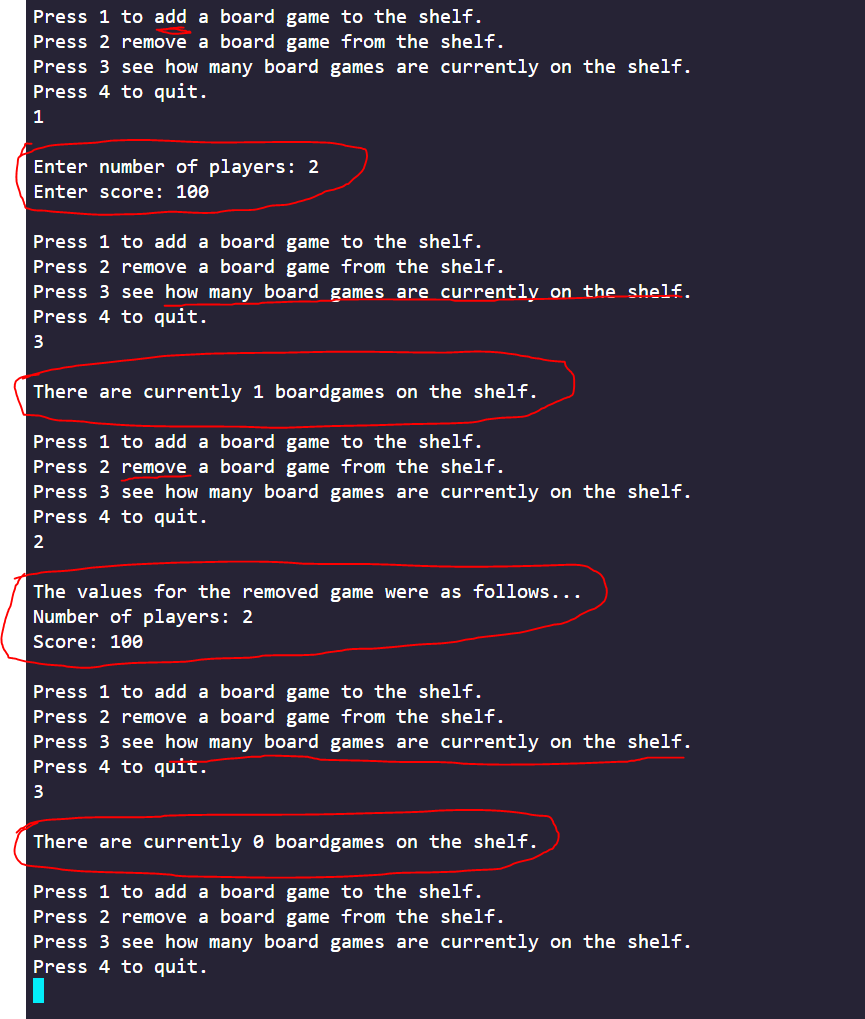
Run instructions: We’re submitting 2 folders containing task 1-3 and task 4, zip them both into one zip file. Both includes all the header files needed to run on its own.

Task 1:

To design the add method, we first pass in a game into the initial function. This game is added to the private array of games which makes up a shelf, it is added specifically at the index of the number of games stored on the shelf. This is so that whenever a game is added to the array it is added at the first available space that does not contain another game. The numStored variable is then incremented by one to show that the current space is now filled.

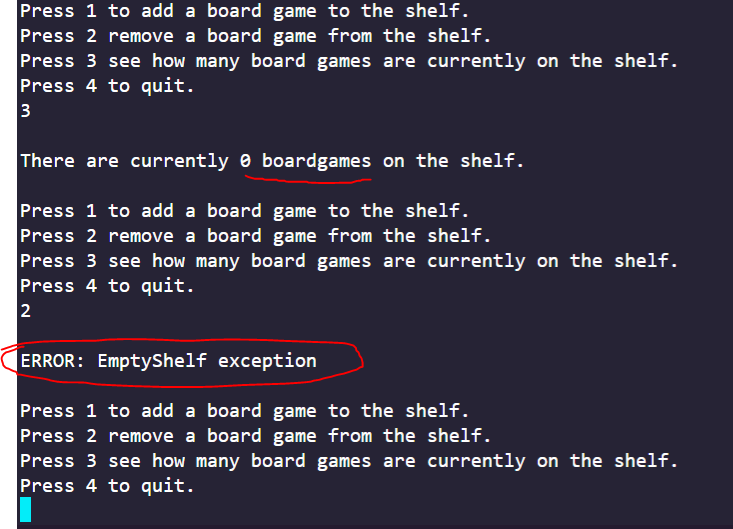
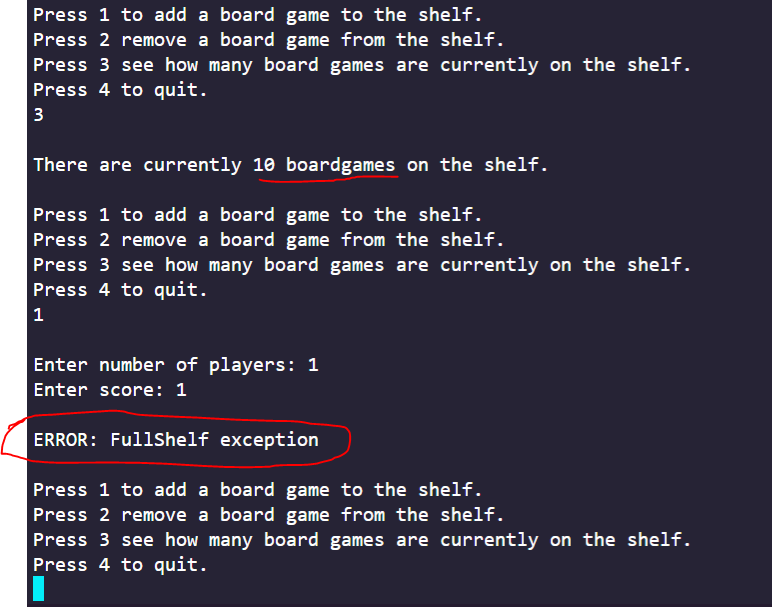
For the remove method, we first created a temporary variable that holds the last game stored in the array of games. Then we set the last game in the array to an empty value. After we decremented the variable that counts the number of games in the array by 1 if that value is greater than zero. Finally, we Outputted the attributes for the temporary game that was stored in a variable using cout. Then we returned the temporary game.

Task 2:



Task 3:

For taks 3, we created two separate exception classes. One for the FullShelf exception and one for the EmptyShelf which are used to catch the errors within the class.

Some of the advantages of trapping an error in the class versus the calling function include: having separate code from the main function which allows for easier understanding and modification, you can easily figure out what type of error occured by looking back at the exception code because the errors are easily differentiated, and it is much cleaner because you could try to test for an error within a large specific block of code much more easily. 

Task 4:

Using template instead of specific types will save time and effort and write code, and also will make the structure of the code really easy to understand. You don’t have to repeat your code, writing specific cases, when the class or function can just be ordered to morphed into whatever type you need later on. Templates can also be overloaded, which are equally convenient compared to normal functions and classes. Also, function templates use no memory.

It also saves memory in that some classes uses floating point numbers in their variables, while others are simpler and only need integers. Having the same template for both objects would save the latter object from having to use doubles.

Disclaimer: I’m not sure what the exercise is asking of us. We created a collection/shelf of games and input some video-games, but we could also have created a collection of video-games and input some video-games. Both work.

