Marie Biernacki

COMP-SCI 303 Data Structures

Professor Shah

Assignment 01 Report

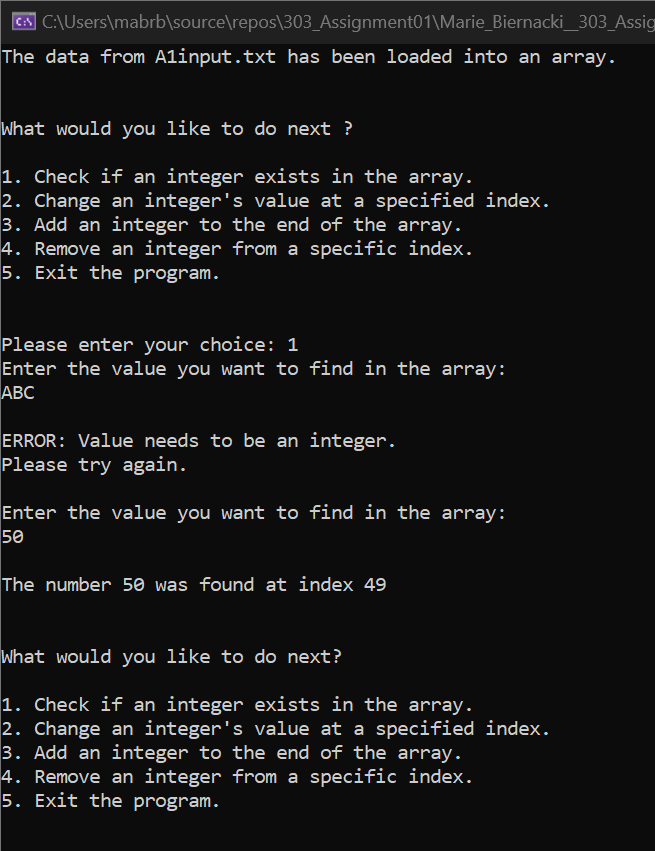
**PROGRAM EXECUTION:**

* **IN MAIN.CPP**
  + create temporary dynamic array to store data from input file and a temporary size
  + call the readFile function, passing tempArray and tempSize as parameters
  + call the operations function, passing tempArray and tempSize as parameters
* **IN HEADER.CPP**
  + **declare the following functions:**
    - bool isDigits(string str1);
    - void readFile(int\*& arr, int& arrSize);
    - void operations(int arr[], int arrSize);
    - bool doesExist(int arr[], int arrSize);
    - void modifyValue(int arr[], int arrSize);
    - void addInteger(int\*& arr, int& arrSize);
    - void removeIdx(int\*& arr, int& arrSize);
* **IN FUNCTIONS.CPP**
  + **define the following functions:**
    - **bool isDigits(string str1);**
      * return true or false based on if the string contains only integers
    - **void readFile(int\*& arr, int& arrSize);**
      * open the input file A1input.text and verify
      * use stringstream to read all of the values into the array
      * output a message stating the file was read into an array
    - **void operations(int arr[], int arrSize);**
      * output the menu options to the user and prompt for input
      * based on user input, call the appropriate function or exit the program
      * if the user input is invalid, continue to loop until the user enters the correct input
    - **bool doesExist(int arr[], int arrSize);**
      * prompt the user for a value to find in the array
      * loop through the array until the value is found or not found
      * if the user input is invalid, continue to loop until the user enters the correct input
    - **void modifyValue(int arr[], int arrSize);**
      * prompt user for index value to modify
        + if the user input for index is invalid, loop until the user enters correct input
      * prompt user for value to add to the specified index
        + if the user input for value is invalid, loop until the user enters correct input
      * otherwise, output the old value at the specified index, update the index to the new value, and output the new value at the specified index
    - **void addInteger(int\*& arr, int& arrSize);**
      * prompt user for a number to add to the end of the array
        + if the user input for value is invalid, loop until the user enters correct input
      * otherwise, update the array size and create a new array using the new size
      * copy all elements in the current array to the new array and add the new element to the end
      * output a statement that the number was successfully added to the end of the array
    - **void removeIdx(int\*& arr, int& arrSize);**
      * prompt user for an index value to remove
        + if the user input for index is invalid, loop until the user enters correct input
      * otherwise, update the array size and create a new array using the new size (smaller)
      * copy all elements in the current array to the smaller array, except for the element at the specified index
      * output a statement that the index was successfully removed

**OUTPUT SCREENSHOTS:**

1. **doesExist**

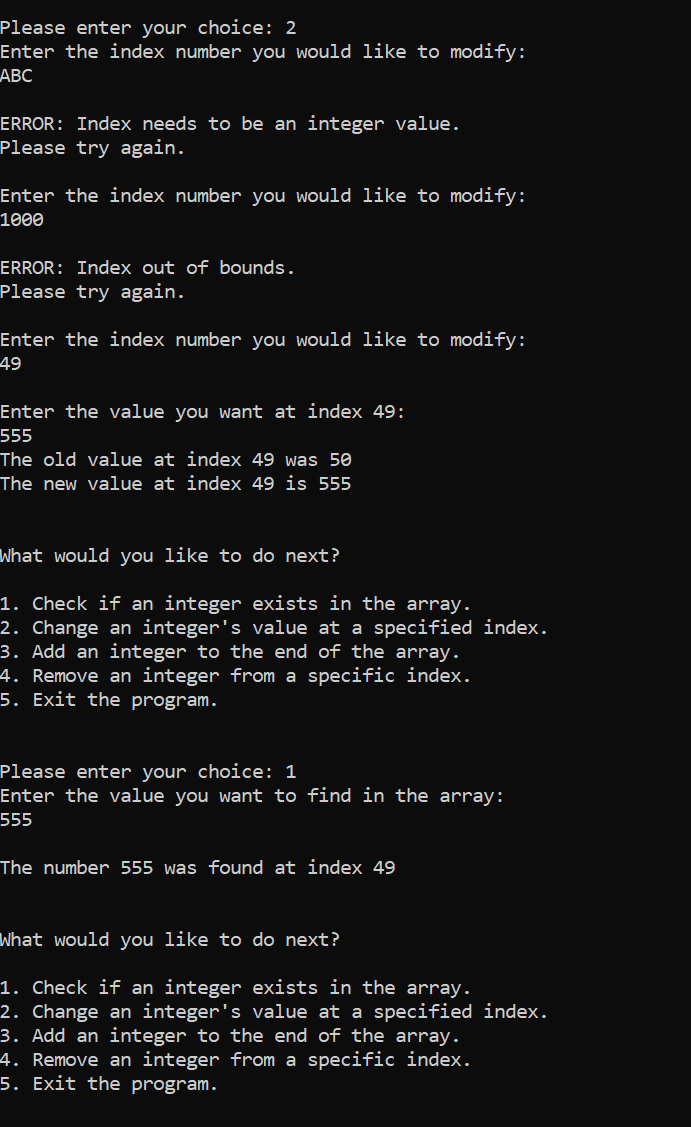
handles errors appropriately, finds the value 50 in the array



1. **modifyValue**

handles errors appropriately, replaces the value at index 49 with 555

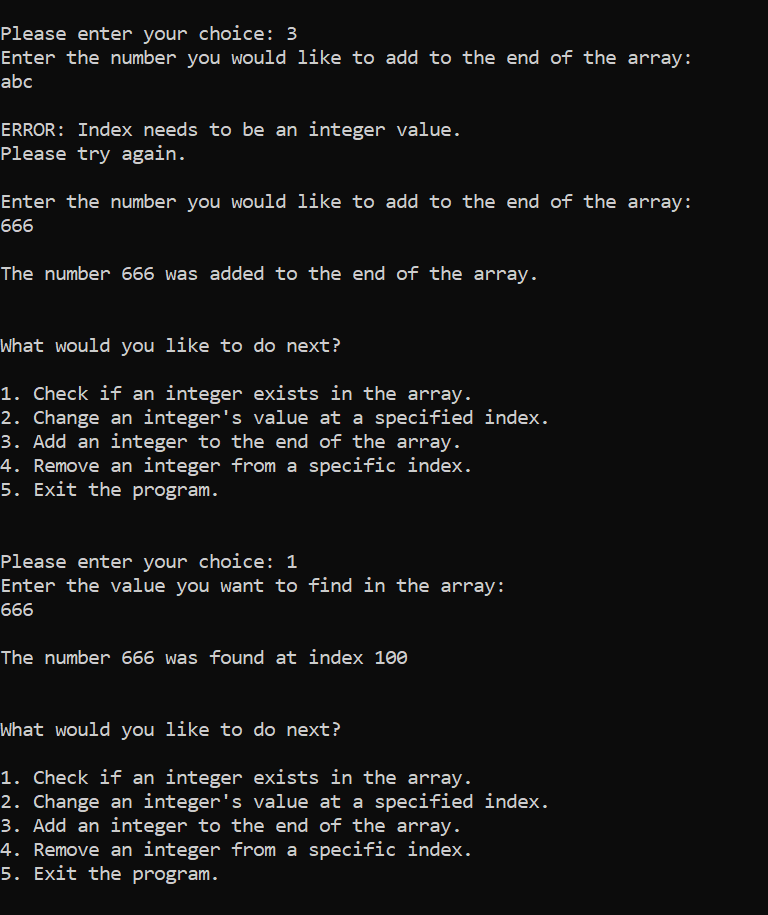
(verified by checking that 555 exists at index 49)

****

1. **addInteger**

handles errors appropriately, adds the value 666 to the end of the array

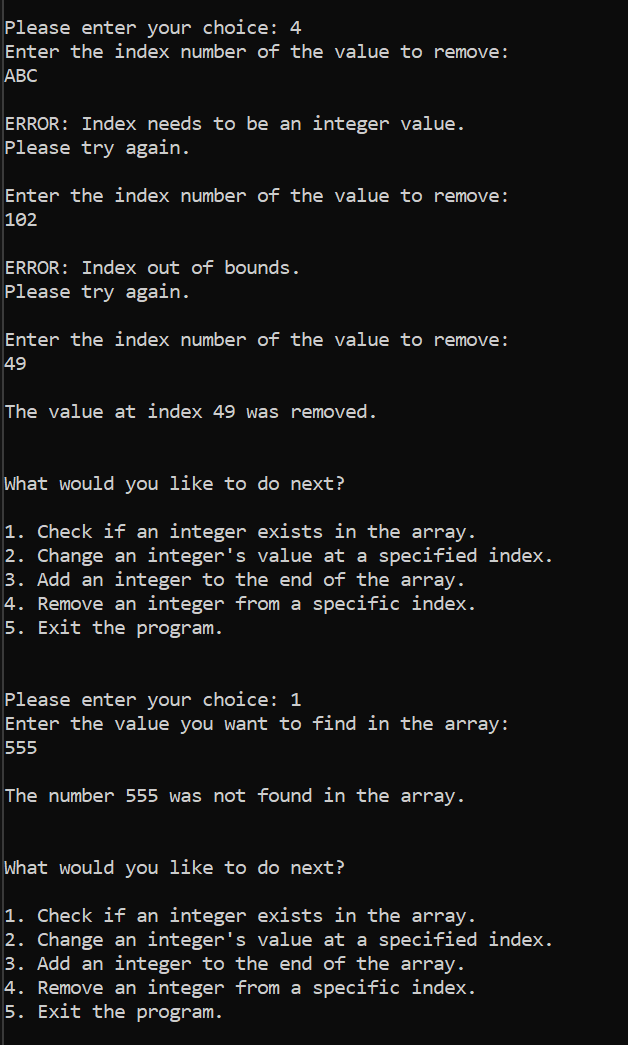
(verified by checking that 666 exists at index 100)

****

1. **removeIdx**

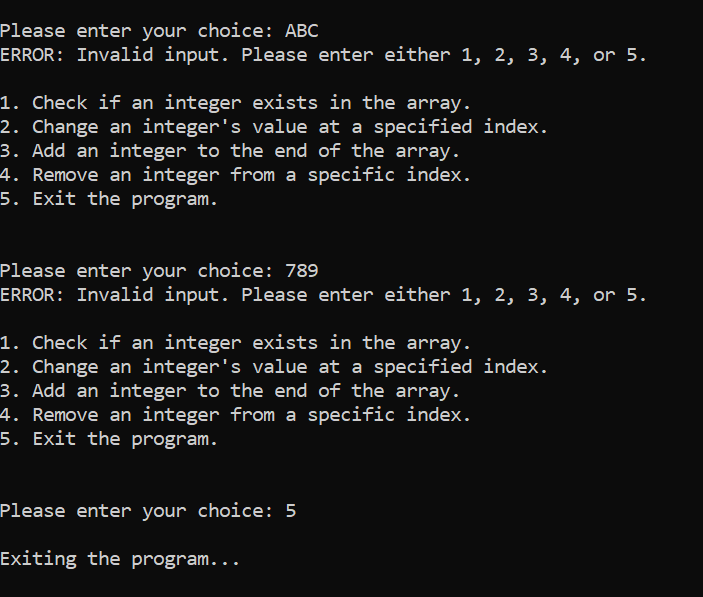
handles errors appropriately, removes the value at index 49

(verified by checking if 555 exists in the array — it does not)

****

1. **operations**

handles errors appropriately, successfully exits the program when asked

****