

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

```
C:\Users\mabrb\source\repos\303_Assignment01\Marie_Biernacki_303_Assig
The data from A1input.txt has been loaded into an array.

What would you like to do next ?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.

Please enter your choice: 1
Enter the value you want to find in the array:
ABC

ERROR: Value needs to be an integer.
Please try again.

Enter the value you want to find in the array:
50

The number 50 was found at index 49

What would you like to do next?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.
```

2. modifyValue

handles errors appropriately, replaces the value at index 49 with 555 (verified by checking that 555 exists at index 49)

```
Please enter your choice: 2
Enter the index number you would like to modify:
ABC

ERROR: Index needs to be an integer value.
Please try again.

Enter the index number you would like to modify:
1000

ERROR: Index out of bounds.
Please try again.

Enter the index number you would like to modify:
49

Enter the value you want at index 49:
555
The old value at index 49 was 50
The new value at index 49 is 555

What would you like to do next?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.

Please enter your choice: 1
Enter the value you want to find in the array:
555

The number 555 was found at index 49

What would you like to do next?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.
```

3. addInteger

handles errors appropriately, adds the value 666 to the end of the array
(verified by checking that 666 exists at index 100)

```
Please enter your choice: 3
Enter the number you would like to add to the end of the array:
abc

ERROR: Index needs to be an integer value.
Please try again.

Enter the number you would like to add to the end of the array:
666

The number 666 was added to the end of the array.

What would you like to do next?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.

Please enter your choice: 1
Enter the value you want to find in the array:
666

The number 666 was found at index 100

What would you like to do next?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.
```

4. removeIdx

handles errors appropriately, removes the value at index 49
(verified by checking if 555 exists in the array — it does not)

```
Please enter your choice: 4
Enter the index number of the value to remove:
ABC

ERROR: Index needs to be an integer value.
Please try again.

Enter the index number of the value to remove:
102

ERROR: Index out of bounds.
Please try again.

Enter the index number of the value to remove:
49

The value at index 49 was removed.

What would you like to do next?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.

Please enter your choice: 1
Enter the value you want to find in the array:
555

The number 555 was not found in the array.

What would you like to do next?

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.
```

5. operations

handles errors appropriately, successfully exits the program when asked

```
Please enter your choice: ABC  
ERROR: Invalid input. Please enter either 1, 2, 3, 4, or 5.
```

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.

```
Please enter your choice: 789  
ERROR: Invalid input. Please enter either 1, 2, 3, 4, or 5.
```

1. Check if an integer exists in the array.
2. Change an integer's value at a specified index.
3. Add an integer to the end of the array.
4. Remove an integer from a specific index.
5. Exit the program.

```
Please enter your choice: 5
```

```
Exiting the program...
```