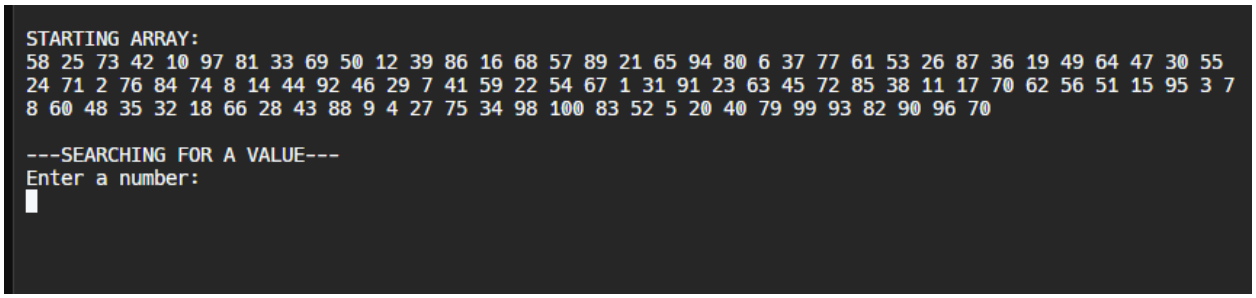
Haley Nilsen
CS303 Data Structures
Assignment 1
9/11/23

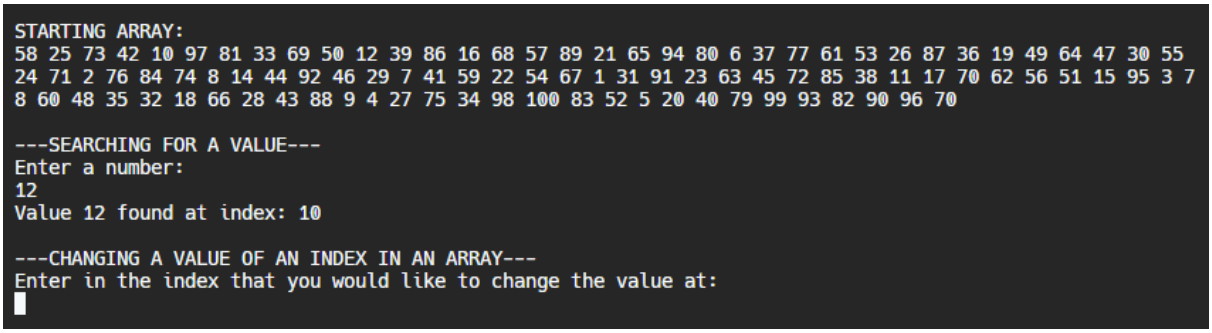
Running the Code (with screenshots):

When initially running the program, the array (read in from the .txt file) will appear on the screen.

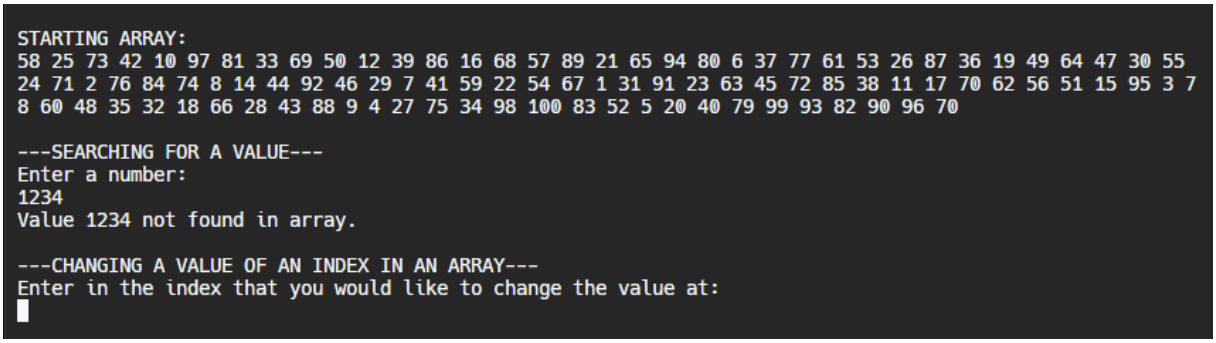
1. The program will ask the user for a number.



2. After input, the program will search for the number in the array and print the value and the index at which the value occurs. If it is not found in the array, there will be a message printed onto the screen. The program will continue with either path.
    a. If value is found



    b. If value is not found



3. Then the program will prompt the user to enter in an index value. This will be used to locate the value that the user will change. This is the first block of code that there is a

try/catch block implemented. To test it, when asking for user input for this section, simply enter any non integer.

When the user enters a value, if an integer is entered, the program will return the value at that location and prompt the user to enter a value to replace the old value at the given index.

```
---CHANGING A VALUE OF AN INDEX IN AN ARRAY---
Enter in the index that you would like to change the value at:
8

Previous Value: 69
Enter in new value:
```

a. To test the try/catch block, when the program prompts for an index, input any non-integer. Entering a non-integer throws a runtime error for the program to catch, therefore displaying an error message. This will terminate the program.

```
---CHANGING A VALUE OF AN INDEX IN AN ARRAY---
Enter in the index that you would like to change the value at:
m
Input is not an integer. Please try again.
exit status 255
```

b. Another way to test the try/catch block is if the user inputs a value higher than the array size (100). Entering a number higher than 100 throws an out of range error, to which an error message is displayed on screen. This will terminate the program.

```
---CHANGING A VALUE OF AN INDEX IN AN ARRAY---
Enter in the index that you would like to change the value at:
123456
Input is higher than array size. Please try again.
exit status 255
```

After a correct index is entered, the user will be prompted to enter in a value. This can be any integer, and there is a try/catch block to validate the user input. After the user enters an integer, the program will change the value at the specified index and display the new array to the screen.

```
---CHANGING A VALUE OF AN INDEX IN AN ARRAY---
Enter in the index that you would like to change the value at:
10

Previous Value: 12
Enter in new value:
963
New Value at index 10: 963

CURRENT ARRAY:
58 25 73 42 10 97 81 33 69 50 963 39 86 16 68 57 89 21 65 94 80 6 37 77 61 53 26 87 36 19 49 64 47 30 55
 24 71 2 76 84 74 8 14 44 92 46 29 7 41 59 22 54 67 1 31 91 23 63 45 72 85 38 11 17 70 62 56 51 15 95 3
78 60 48 35 32 18 66 28 43 88 9 4 27 75 34 98 100 83 52 5 20 40 79 99 93 82 90 96 70
```

a. To test the try/catch block, it is the same process as the last step. The user will need to enter in any non-integer, and a runtime error will be thrown with an error message displayed on screen. This will terminate the program.

```
---CHANGING A VALUE OF AN INDEX IN AN ARRAY---
Enter in the index that you would like to change the value at:
10

Previous Value: 12
Enter in new value:
m
Input is not an integer. Please try again.
exit status 255
```

After displaying the array with the modified value to the screen, the program then prompts the user to enter in a number that will set the size of the new array (it does need to be greater than 100). The program will then prompt the user to enter in a value that will be added to the end of the current array, filling the remaining space in with zeros.

```
---MAKING THE NEW ARRAY TO ADD MORE VALUES---
Please enter an array size, must be greater than 100.
150

---ADDING A VALUE TO THE END OF THE ARRAY---
Please enter a numerical value:
482


CURRENT ARRAY:
58 25 73 42 10 97 81 33 69 50 963 39 86 16 68 57 89 21 65 94 80 6 37 77 61 53 26 87 36 19 49 64 47 30 55
 24 71 2 76 84 74 8 14 44 92 46 29 7 41 59 22 54 67 1 31 91 23 63 45 72 85 38 11 17 70 62 56 51 15 95 3
78 60 48 35 32 18 66 28 43 88 9 4 27 75 34 98 100 83 52 5 20 40 79 99 93 82 90 96 70 482 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

The program will then prompt the user to enter in a valid index (one that is less than the array size that was defined in the previous step). The program will then display the old value at that given index and change the value to 0. It will then display the final array to the screen.

```
---CHANGING A VALUE AT AN INDEX TO 0 (REMOVING THE VALUE)---
Please enter in an index:
13

Old value at index 13: 16
New value at index 13: 0

FINAL ARRAY:
58 25 73 42 10 97 81 33 69 50 963 39 86 0 68 57 89 21 65 94 80 6 37 77 61 53 26 87 36 19 49 64 47 30 55
24 71 2 76 84 74 8 14 44 92 46 29 7 41 59 22 54 67 1 31 91 23 63 45 72 85 38 11 17 70 62 56 51 15 95 3 7
8 60 48 35 32 18 66 28 43 88 9 4 27 75 34 98 100 83 52 5 20 40 79 99 93 82 90 96 70 482 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

END OF PROGRAM.
```