Assignment 1 - Muzan Daffaallah

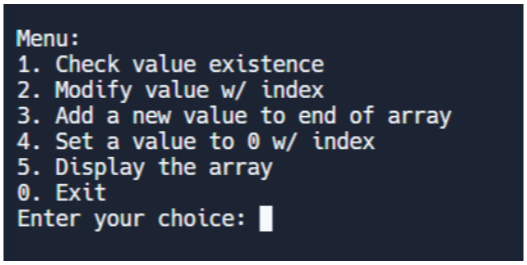
Running the code:

START. Open input file, if the file fails to open, display an error message and return 1. Initialize array with capacity of 10 and set the initial size to 0. Read an integer value from the file. If the size of the array is equal or exceeds the capcity, double the capacity, allocate a new array with the increased capacity, copy elements to array, set the array to the new array and increment the size. Close the input file. Start menu loop. While the choice is not 0, display the menu options and get user choice. If choice is 1 get value from user. If value found, print the index where it was found, otherwise print error message. If choice is 2 get index from user. Print old and new value, or if invalid print error message. If choice is 3, get new value from user. Print the new value and size of the array. If choice is 4 get index from user. If index is valid print confirmation message, or if invalid print error message. If choice is 5 print array. If choice is 0 exit program. **END**.

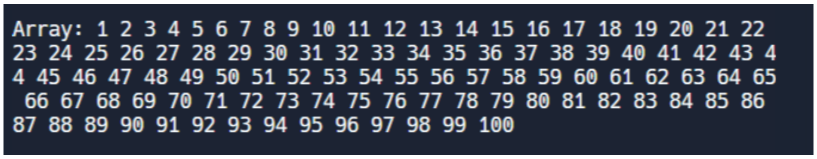
Screenshots Part One:

Additions:

I added a menu for efficiency. I also added a “display an array” function to make sure functions are working correctly and array is being updated.



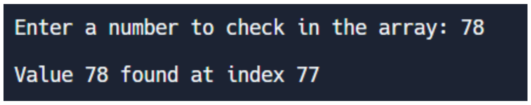
```
Menu:
1. Check value existence
2. Modify value w/ index
3. Add a new value to end of array
4. Set a value to 0 w/ index
5. Display the array
0. Exit
Enter your choice: █
```



```
Array: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 4
4 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65
 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86
87 88 89 90 91 92 93 94 95 96 97 98 99 100
```

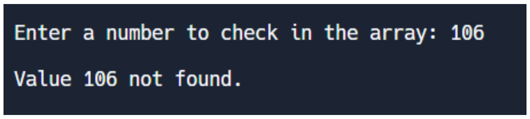
1. The progam asks the user to enter a number. Then the it checks to see if the number is in the array. If it is not found, it will return a “not found” message.

Number Found:



```
Enter a number to check in the array: 78

Value 78 found at index 77
```

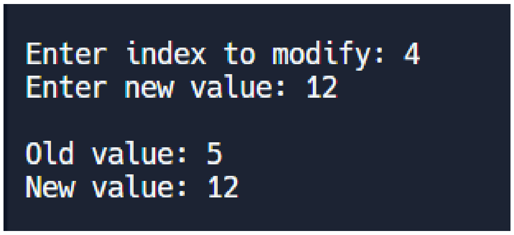
Number Not Found:



```
Enter a number to check in the array: 106

Value 106 not found.
```

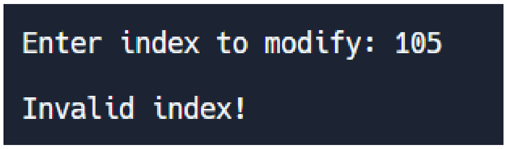
2. The program asks the user to enter an index to modify the number at that index. If the index is invalid, error message is printed.

Valid Index:



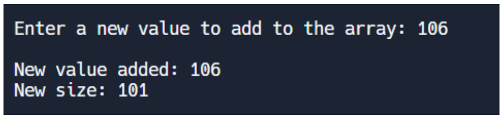
```
Enter index to modify: 4
Enter new value: 12

Old value: 5
New value: 12
```

Invalid Index:



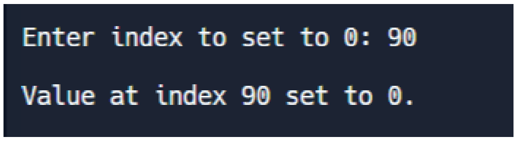
```
Enter index to modify: 105

Invalid index!
```

3. The program will ask the user for a number and will add it to the end of the array. It will print the value added and the new size.



```
Enter a new value to add to the array: 106

New value added: 106
New size: 101
```

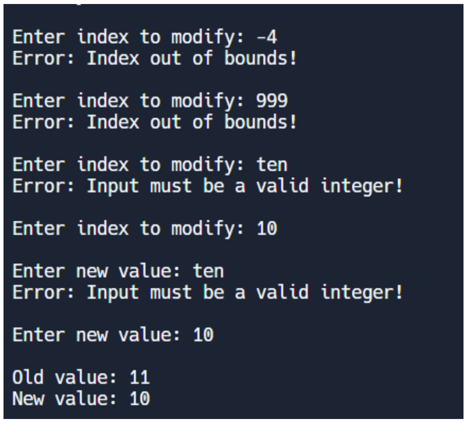
4. The program will ask the user for an index. The value at that index will be set to 0.



```
Enter index to set to 0: 90

Value at index 90 set to 0.
```

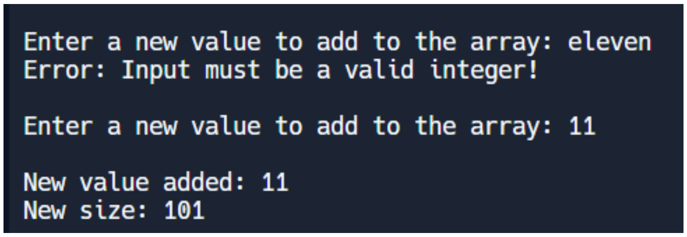
Screenshots Part Two:

1. If the user enters an index that isnt an integer or isn’t a valid index , an error message appears. If the user enters a value that isnt an integer, an error message appears. For both of these errors, the program will loop until valid data is inputted.



```
Enter index to modify: -4
Error: Index out of bounds!

Enter index to modify: 999
Error: Index out of bounds!

Enter index to modify: ten
Error: Input must be a valid integer!

Enter index to modify: 10

Enter new value: ten
Error: Input must be a valid integer!

Enter new value: 10

Old value: 11
New value: 10
```

2. If the user inputs a value that isn’t an integer, an error message will be printed, and the program will loop until valid data is entered.



```
Enter a new value to add to the array: eleven
Error: Input must be a valid integer!

Enter a new value to add to the array: 11

New value added: 11
New size: 101
```