

## Control Flow (loops and branches), Inputs & Outputs, and Making Decisions

*Submitted By*  
*Fahd Humayun*  
*168000889 (fh186)*  
*Submitted To*  
*Christos Mitropoulos*

### Exercise:

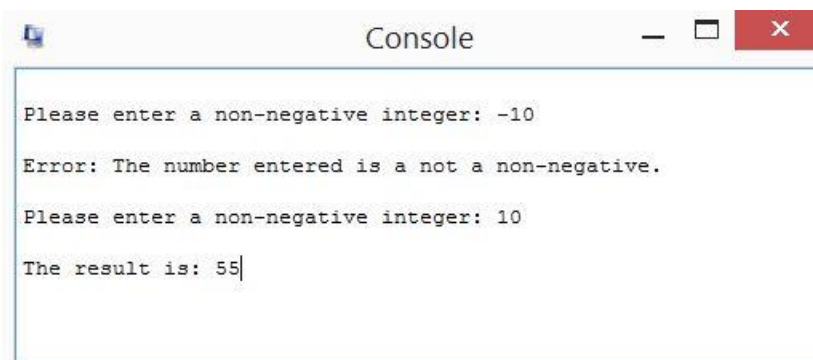
The number of iterations are 6.

Yes, it affects the number of iterations, now the number of iterations are 5, because, by swapping lines 12 and 13 the increment comes before the condition to check, and as it increments first then it checks and the loop ends rather than incrementing then jumping to the loop and then checking.

The codes for each Assignment are attached at the end. Output of the codes are below:

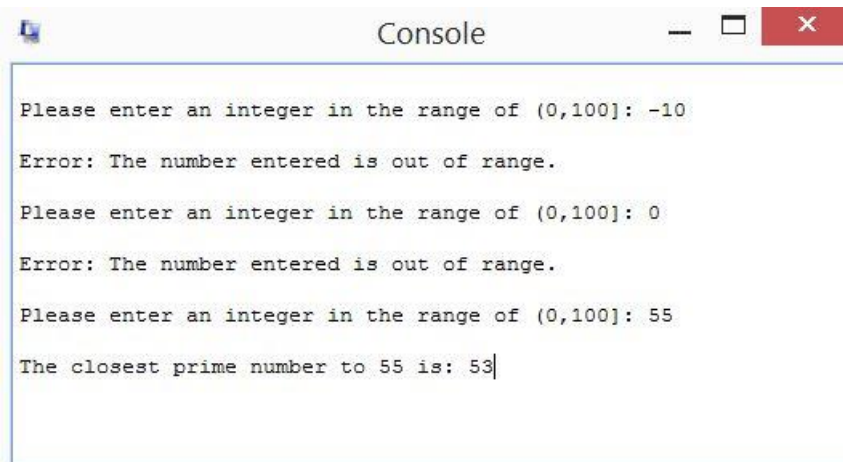
### Outputs of the codes:

#### Assignment 1:



```
Console
Please enter a non-negative integer: -10
Error: The number entered is a not a non-negative.
Please enter a non-negative integer: 10
The result is: 55|
```

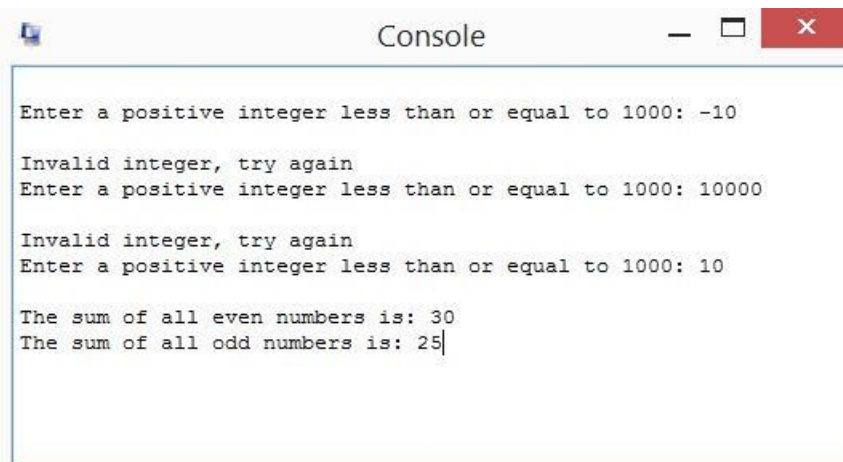
## Assignment 2:



```
Console

Please enter an integer in the range of (0,100]: -10
Error: The number entered is out of range.
Please enter an integer in the range of (0,100]: 0
Error: The number entered is out of range.
Please enter an integer in the range of (0,100]: 55
The closest prime number to 55 is: 53|
```

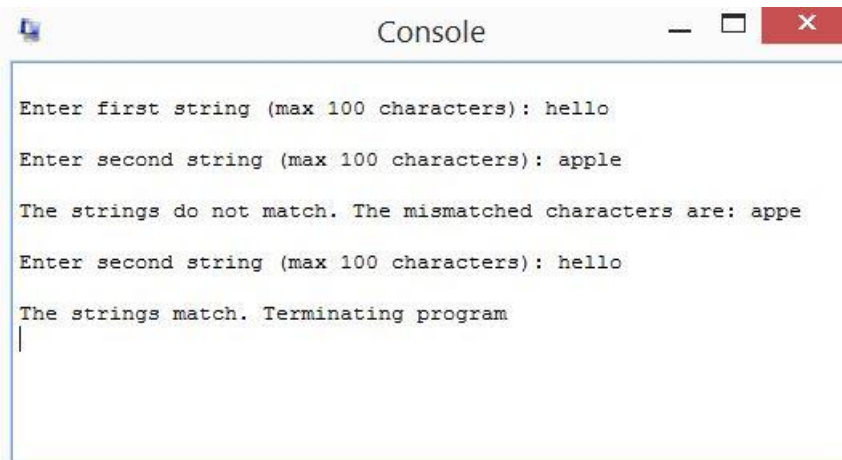
## Assignment 3:



```
Console

Enter a positive integer less than or equal to 1000: -10
Invalid integer, try again
Enter a positive integer less than or equal to 1000: 10000
Invalid integer, try again
Enter a positive integer less than or equal to 1000: 10
The sum of all even numbers is: 30
The sum of all odd numbers is: 25|
```

#### Assignment 4:



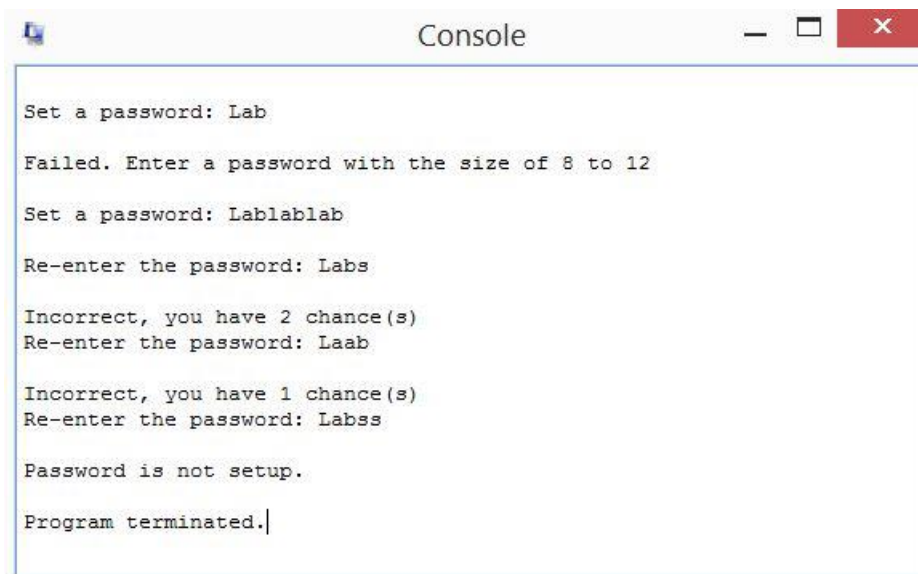
```
Console

Enter first string (max 100 characters): hello
Enter second string (max 100 characters): apple

The strings do not match. The mismatched characters are: appe
Enter second string (max 100 characters): hello
The strings match. Terminating program
|
```

#### Assignment 5:

For incorrect password on 3 attempts

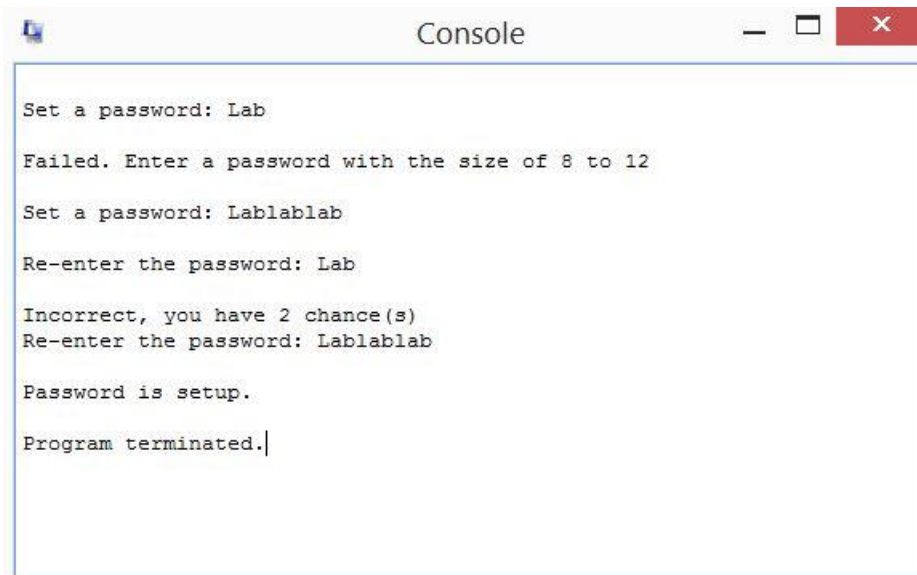


```
Console

Set a password: Lab
Failed. Enter a password with the size of 8 to 12
Set a password: Lablablab
Re-enter the password: Labs
Incorrect, you have 2 chance(s)
Re-enter the password: Laab
Incorrect, you have 1 chance(s)
Re-enter the password: Labss

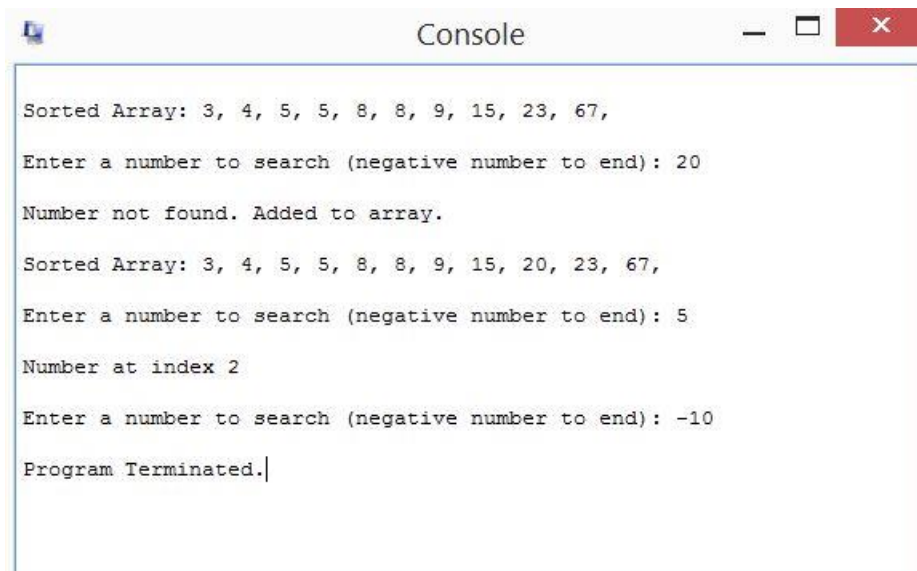
Password is not setup.
Program terminated.|
```

For correct password



```
Set a password: Lab
Failed. Enter a password with the size of 8 to 12
Set a password: Lablablab
Re-enter the password: Lab
Incorrect, you have 2 chance(s)
Re-enter the password: Lablablab
Password is setup.
Program terminated.|
```

### **Assignment 6:**



```
Sorted Array: 3, 4, 5, 5, 8, 8, 9, 15, 23, 67,
Enter a number to search (negative number to end): 20
Number not found. Added to array.
Sorted Array: 3, 4, 5, 5, 8, 8, 9, 15, 20, 23, 67,
Enter a number to search (negative number to end): 5
Number at index 2
Enter a number to search (negative number to end): -10
Program Terminated.|
```

*Note: Codes attached after this page of conclusion.*

### **Conclusion:**

The assignments of this lab were focused on how to use flow control i.e. loops and branches by combine usage of both the branch instructions and the jump instruction like *slt, beq, bne, j, jr, jal* to create iterations/loops.

The assignments of the previous lab that were focused on the inputs and outputs e.g. reading and writing integers and strings, so, the knowledge and the implementation of those codes were also involved in the assignments of this lab.

In order to understand the behavior of loops and branches each of the code was run step by step in order to observe the change in register values, which also helped in most of the debugging parts of the assignments.

For the jump instructions particularly *jal (jump and link)* instruction was used and then the code was run step by step to observe the behavior and change in values of the temporary registers, the *\$ra*, and by using stack pointers to store and retrieve the data on/from the stack.

The use and behavior of memory locations was also observed, especially for the assignments of comparing strings and for storing integers in an integer array.