

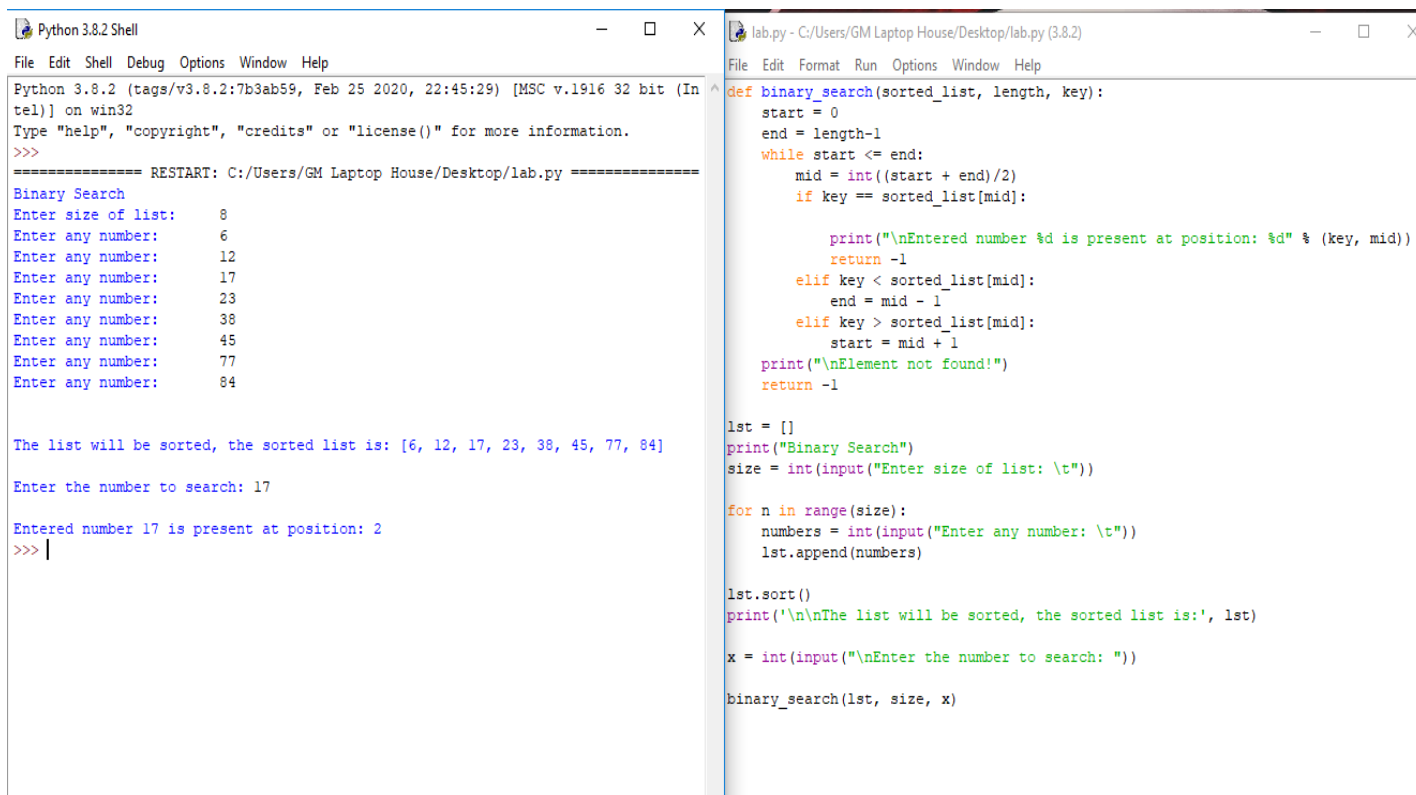
Task Lab

Name : Zaryab

Roll num : BSCS/S17/0109

CMS ID : 293-2017

Q1. Implement Binary Search Algorithm by Python Programming, use values of figure 1.



The image shows two side-by-side windows from a Python 3.8.2 Shell. The left window displays the execution of a script, and the right window shows the source code of the script.

Left Window (Python 3.8.2 Shell):

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/GM Laptop House/Desktop/lab.py =====
Binary Search
Enter size of list: 8
Enter any number: 6
Enter any number: 12
Enter any number: 17
Enter any number: 23
Enter any number: 38
Enter any number: 45
Enter any number: 77
Enter any number: 84

The list will be sorted, the sorted list is: [6, 12, 17, 23, 38, 45, 77, 84]

Enter the number to search: 17

Entered number 17 is present at position: 2
>>> |
```

Right Window (lab.py - C:/Users/GM Laptop House/Desktop/lab.py (3.8.2)):

```
File Edit Format Run Options Window Help

def binary_search(sorted_list, length, key):
    start = 0
    end = length-1
    while start <= end:
        mid = int((start + end)/2)
        if key == sorted_list[mid]:
            print("\nEntered number %d is present at position: %d" % (key, mid))
            return -1
        elif key < sorted_list[mid]:
            end = mid - 1
        elif key > sorted_list[mid]:
            start = mid + 1
    print("\nElement not found!")
    return -1

lst = []
print("Binary Search")
size = int(input("Enter size of list: \t"))

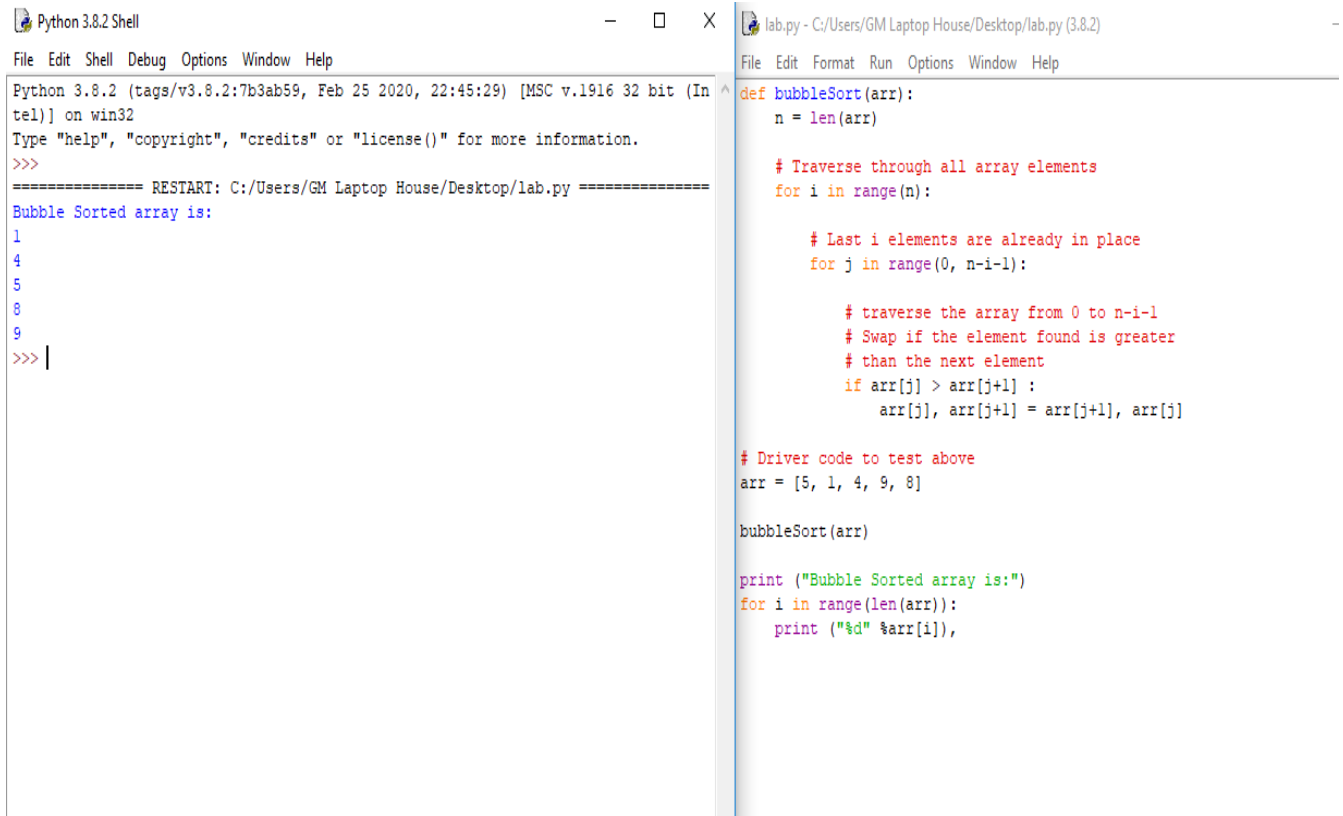
for n in range(size):
    numbers = int(input("Enter any number: \t"))
    lst.append(numbers)

lst.sort()
print('\n\nThe list will be sorted, the sorted list is:', lst)

x = int(input("\nEnter the number to search: "))

binary_search(lst, size, x)
```

Q2. Implement Bubble sort Algorithm by Python Programming, Take an array of 5 elements.



The image shows two windows from a Windows operating system. The left window is titled 'Python 3.8.2 Shell' and displays the output of a Python script. The right window is titled 'lab.py - C:/Users/GM Laptop House/Desktop/lab.py (3.8.2)' and shows the source code for the bubble sort algorithm.

Python 3.8.2 Shell Output:

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/GM Laptop House/Desktop/lab.py =====
Bubble Sorted array is:
1
4
5
8
9
>>> |
```

lab.py Source Code:

```
def bubbleSort(arr):
    n = len(arr)

    # Traverse through all array elements
    for i in range(n):

        # Last i elements are already in place
        for j in range(0, n-i-1):

            # traverse the array from 0 to n-i-1
            # Swap if the element found is greater
            # than the next element
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

# Driver code to test above
arr = [5, 1, 4, 9, 8]

bubbleSort(arr)

print ("Bubble Sorted array is:")
for i in range(len(arr)):
    print ("%d" %arr[i],
```