

Exercise (a)

```
# Self-Defined:
def binary_search(Arr, Item):
    BEG = 0
    END = len(Arr) - 1
    MID = int((BEG + END)/2)
    while True:
        if BEG > END or Arr[MID] == Item:
            break
        if Item < Arr[MID]:
            END = MID - 1
        else:
            BEG = MID + 1

        MID = int((BEG + END)/2)

    if Arr[MID] == Item:
        LOC = MID
    else:
        LOC = None
    return LOC

lis = eval(input("Enter an Array in array-format: "))
search_item = int(input('Enter Search Item: '))
print(f"Searched in Arr: {lis} for {search_item}, Found at:{binary_search(lis, 3)}")

import timeit
print("Difference")
print("Self-Made Searching: ",
timeit.timeit("binary_search([1,2,3,4,5],5)",globals=globals(), number=500))
print("Built-in Searching: ",
timeit.timeit("[1,2,3,4,5].index(5)",globals=globals(), number=500))

# Output

Searched in Arr: [1, 2, 3, 4] for 3, Found at:2
Difference
Self-Made Searching:  0.0007406999939121306
Built-in Searching:  0.00014240003656595945
```

Exercise (c)

```
def binary_search_and_insertion(arr, item):
    if arr != sorted(arr):
        print("Input List was UnSorted, Sorting and
continuing")
    arr = sorted(arr)
    beg, end = 0 , len(arr)-1
    while True:
        mid = (beg+end)//2
        if (beg>end or arr[mid]==item):
            if arr[mid] == item:
                loc = mid
            else:
                loc = None
            break

        if item < arr[mid]:
            end = mid - 1
        else:
            beg = mid + 1
        mid = int((beg + end)/2)

    if loc is None:
        print("Item Not Found, Inserting it in a sorted-
way")
        for idx in range(len(arr)):
            if item < arr[idx]:
                arr.insert(idx, item)
        return arr
    return loc

print(binary_search_and_insertion([1,2,3,4,6],5))
```

```
Item Not Found, Inserting it in a sorted-way
[1, 2, 3, 4, 5, 6]
```