NYPLIT – Intro to Algorithms

Sequential Search

The following code is an incomplete implementation of the Sequential Search for an unsorted list of values

```
def sequentialSearch(ValueList, target):
    n = len(ValueList)
    for i in range(n):
        # When target is found, return True
        if _____________________:
            return ____

    # If not found, return False
    return False

# test program
list = [11,4,5,9,2,17,24]
print print (sequentialSearch(list,9)) # Ans: True
```

a. Complete the implementation by providing the rest of the required codes

Challenge Question:

b. Write a new function named `sortedSequentialSearch()`,u that improves that above implementation for a sorted list of values.

Binary Search

The following code is an incomplete implementation of the Binary Search

```
def binarySearch( ValueList, target ):
    # first index of ValueList
    low = 0
    # last index of ValueList
    high = _________________

    while not high < low:
        # Find the midpoint of the sequence
        mid = ____________________

        # If yes, return midpoint (i.e. index of the list)
        if target == ValueList[mid]:
            return mid

        # Or is the target smaller than midpoint value?
        elif target < ValueList[mid]:
            ________________

        # Or is the target greater than the midpoint value?
        else:
            ________________

    # target is not in the list of values
    return -1

# test program
list = [2,7,13,13,24,35,47]
print(binarySearch(list,24)) # Ans: 4
```

c. Complete the implementation by providing the rest of the required codes

Challenge Question:

d. Modify the code to return the position of the **first occurrence** of a value that can occur multiple times in the sorted list of values.