Date : 09/08/2023

Given an array that is increasing and decreasing, find the peak

Example:

input : [10,20,30,15,10]

output : 30

```
def findPeak(nums):
        peak= -1
        if len(nums) == 0:
                return peak
        low = 0
        high =  len(nums) - 1
        while low <= high:
                mid = (low + high)//2
                if mid == len(nums) - 1 or nums[mid] > nums[mid + 1]:
                        peak = nums[mid]
                        high = mid - 1
                else:
                        low = mid + 1
        return peak
```

Test cases:

input : [1,2,3,4,5,6,1]

output : 6

input : []

output: -1

input : [1]

output : 1

==========

2 Given an input string , output the count of vowels and consonants only containing alphanet characters

example : apple

output : [2, 3]

```
def countVowelAndConsonants(s):
        if len(s) == 0:
                return [ 0, 0]
        vowelCount = 0
        consonantCount = 0
        s=s.lower()
        for ch in s:
                if ch in ‘aeiou’:
                        vowelCount += 1
                else:
                        consonantCount += 1
        res = [ vowelCount, consonantCount]
        return res
```

TimeComplexity : O(n)
SpaceComplexity : O(1)

Test Cases:
input : qwerty
output : [1,5]

input : []
output :[0,0]

input : a
output : [1,0]

****

2.1.Given input string , output the count of vowels and consonants

```
def countVowelAndConsonants(s):
        if len(s) == 0:
                return [ 0, 0]
        vowelCount = 0
        consonantCount = 0
```

```
s=s.lower()
for ch in s:
        if ch in ‘aeiou’:
                vowelCount += 1
        elif ch in ‘bcdfghjklmnpqrstvwxyz’:
                consonantCount += 1
res = [ vowelCount, consonantCount]
return res
```

TimeComplexity : O(n)
SpaceComplexity : O(1)

input : apple$
output : [2,3]

input : []
output : [0,0]

input : [a]
output : [1,0]

==========

3. Given a string print it in a reverse order

```
def reverseString(s):
    strList = list(s)
    if len(s) == 0:
        return s
    left = 0
    right = len(strList) - 1
    while left <= right:
        temp = strList[left]
        strList[left] = strList[right]
        strList[right] = temp
        left += 1
```

```
        right -= 1
    return ''.join(strList)
```

TimeComplexity : O[n]
SpaceComplexity : O[1]

Test Cases:

input : apple
output : elppa

input : a
output : a

input : ‘’
output : ‘’