Leet Code Assignment

Name: K Lalith Aditya

Regd No: 22231

Date: 31-10-2022

### Question 1)

### 1. Two Sum



Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target.

You may assume that each input would have **exactly one solution**, and you may not use the same element twice.

You can return the answer in any order.

### Answer 1)



## Question 2)

# 4. Median of Two Sorted Arrays

Given two sorted arrays nums1 and nums2 of size m and n respectively, return **the** median of the two sorted arrays.

The overall run time complexity should be  $O(\log (m+n))$ .

### Answer 2)

```
class Solution:
    def findMedianSortedArrays(self, nums1,nums2):
        for i in range(0,len(nums2)):
            nums1.append(nums2[i])
        # Bubble Sort
        for i in range(0,len(nums1)-1):
            for j in range(0,len(nums1)-i-1):
                 if (nums1[j]>nums1[j+1]):
                     nums1[j], nums1[j+1] = nums1[j+1], nums1[j]
        if((len(nums1)%2)!=0): # i.e. its odd
            a = nums1[floor(len(nums1)/2)]
            b = nums1[floor((len(nums1)/2)+1)]
            bob = int((a+b)/2)
            return bob
        else: # i.e. its even
            return int(nums1[len(nums1)/2])
```

## Question 3)

#### Palindrome Number

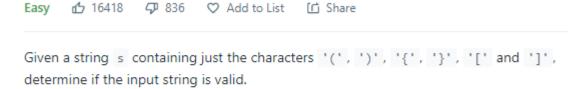
Given an integer x , return true if x is a palindrome, and false otherwise.

### Answer 3)



### Question 4)

### 20. Valid Parentheses



An input string is valid if:

- 1. Open brackets must be closed by the same type of brackets.
- 2. Open brackets must be closed in the correct order.
- 3. Every close bracket has a corresponding open bracket of the same type.

#### Answer 4)

```
class Solution:
     def isValid(self, s):
         flag = 0
for i in range(len(s)-1):
              if(i+1<=len(s)):
                  if(s[i] == '(' and s[i+1]==')'):
                      flag = 0
                  if(s[i] == '(' and s[i+1]!=')'):
                      flag = 1
                  elif(s[i] == ')'):
                      flag = 1
                  if(s[i] == '{' and s[i+1]=='}'):
                      flag = 0
                  if(s[i] == '{' and s[i+1]!='}'):
                      flag = 1
                  elif(s[i] == '}'):
                      flag = 1
                  if(s[i] == '[' and s[i+1]==']'):
                      flag = 0
                  if(s[i] == '[' and s[i+1]!=']'):
                      flag = 1
                  elif(s[i] == ']'):
                      flag = 1
         if (flag == 1):
              return False
         else:
             return True
```

### Question 5)

# 35. Search Insert Position

Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order.

You must write an algorithm with O(log n) runtime complexity.

### Answer 5)

```
class Solution:
    def searchInsert(self,nums,target): # Binary Search
         low = 0
        high = len(nums)-1
flag = 0
        while low <= high:
             mid = int((high + low)/2)
             # If x is greater, ignore left half
             if nums[mid] < target:</pre>
                 low = mid + 1
             # If x is smaller, ignore right half
             elif nums[mid] > target:
    high = mid - 1
             # means x is present at mid
             else:
                 flag = 1
                 return mid
         if flag == 0 :
             while(i<len(nums)):</pre>
                  if (target <nums[i]):</pre>
                      return i+1
                  else:
                      continue
```