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.

**Example 1:**

**Input:** nums = [1,3,5,6], target = 5

**Output:** 2

**Example 2:**

**Input:** nums = [1,3,5,6], target = 2

**Output:** 1

**Example 3:**

**Input:** nums = [1,3,5,6], target = 7

**Output:** 4

Solution:-

public class Solution {

    public int SearchInsert(int[] nums, int target) {

        int len =nums.Length;

        if(len==1 && nums[0]<target){

            return 1;

        }

        else if(len==1 && nums[0]>target){

            return 0;

        }

        for(int i=0;i<len-1;i++)

        {

            if(nums[i]==target){

                return i;

            }

            else if(nums[i]>target){

                return 0;

            }

            else if(nums[i]<target && nums[i+1]>target){

                return i+1;

            }

        }

        if(nums[len-1]==target){

            return len-1;

        }

            return len;

    }

}

Plus One:-

You are given a **large integer** represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.

Increment the large integer by one and return the resulting array of digits.

**Example 1:**

**Input:** digits = [1,2,3]

**Output:** [1,2,4]

**Explanation:** The array represents the integer 123.

Incrementing by one gives 123 + 1 = 124.

Thus, the result should be [1,2,4].

**Example 2:**

**Input:** digits = [4,3,2,1]

**Output:** [4,3,2,2]

**Explanation:** The array represents the integer 4321.

Incrementing by one gives 4321 + 1 = 4322.

Thus, the result should be [4,3,2,2].

**Example 3:**

**Input:** digits = [9]

**Output:** [1,0]

**Explanation:** The array represents the integer 9.

Incrementing by one gives 9 + 1 = 10.

Thus, the result should be [1,0].

Solution:-

public class Solution {

    public int[] PlusOne(int[] digits) {

        int len =digits.Length;

        int i=0;

        for(i=len-1;i>=0;i--)

        {

            if(digits[i]!=9)

            {

                digits[i]+=1;

                break;

            }

            else{

                digits[i]=0;

            }

        }

        if(i==-1){

            int [] newDigit=new int[len+1];

            newDigit[0]=1;

            return newDigit;

        }

        return digits;

        // if(digits[len-1]==9)

        // {

        //     int [] arr=new int[len+1];

        //     for(int k=0;k<arr.Length-1;k++)

        //     {

        //         arr[k]=digits[k];

        //     }

        //     arr[len-1]=1;

        //     arr[len]=0;

        //     return arr;

        // }

        // else{

        //     digits[len-1]=digits[len-1]+1;

        //     return digits;

        // }

    }

}