Problem 1: Detect Duplicate

Description: Write a program to determine if there is a duplicate of a given target value in a sorted array of integers. If a duplicate is found, print true. If no duplicate is found, print false.

Note: Ignore the time complexity for input operations; focus on achieving O(logn) time complexity for checking duplicates.

Input:

- An integer array sorted in non-decreasing order.
- An integer `target` to check for duplicates.

Output:

- Print `true` if the target value has a duplicate in the array.
- Print `false` if the target value does not have a duplicate in the array.

Input	Output
5 1 2 3 4 5 7	false
5 1 1 2 3 5 1	true
7 1 2 3 4 5 7 7 7	true
5 1 2 2 3 4 2	true

Problem 2: LeftSum Array

Description:

Given an array of integers, calculate the sum of all elements to the left of each element in the array. If there are no elements to the left, the sum is zero

Note: Solve this problem in O(n) Complexity

Input:

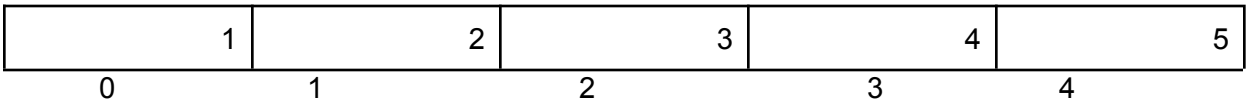
- The first line contains an integer n, the size of the array.
- The second line contains n space-separated integers representing the array.

Output:

- Print the sum of elements to the left of each element in the array.

Input	Output
5 1 2 3 4 5	0 1 3 6 10

Explanation:



Explanation:

- Left Sum of Index 0 is (there is no index left side of index 0, so the sum is 0) : 0
- Left Sum of Index 1 is : (sum of elements from index 0 to 0) = 1
- Left Sum of Index 2 is (sum of elements from index 0 to 1) : (1+2) = 3
- Left Sum of Index 3 is (sum of elements from index 0 to 2) : (1+2+3) = 6
- Left Sum of Index 4 is (sum of elements from index 0 to 3) : (1+2+3+4) = 10