Que1. Given an integer array nums, handle multiple queries of the following type:

1. Calculate the **sum** of the elements of nums between indices left and right **inclusive** where left <= right.

Implement the NumArray class:

- NumArray(int[] nums) Initializes the object with the integer array nums.
- int sumRange(int left, int right) Returns the **sum** of the elements of nums between indices left and right **inclusive** (i.e. nums[left] + nums[left + 1] + ... + nums[right]).

Input
["NumArray", "sumRange", "sumRange", "sumRange"]
[[[-2, 0, 3, -5, 2, -1]], [0, 2], [2, 5], [0, 5]]
Output
[null, 1, -1, -3]

Explanation
NumArray numArray = new NumArray([-2, 0, 3, -5, 2, -1]);
numArray.sumRange(0, 2); // return (-2) + 0 + 3 = 1
numArray.sumRange(2, 5); // return 3 + (-5) + 2 + (-1) = -1
numArray.sumRange(0, 5); // return (-2) + 0 + 3 + (-5) + 2 + (-1) = -3

Constraints:

- 1 <= nums.length <= 10^4
- -10^5 <= nums[i] <= 10^5
- 0 <= left <= right < nums.length
- At most 10^4 calls will be made to sumRange.

Que2. Given a m x n matrix mat and an integer k, return *a matrix* answer *where each* answer[i][j] *is the sum of all elements* mat[r][c] *for*:

- i - k <= r <= i + k,
- j - k <= c <= j + k, and
- (r, c) is a valid position in the matrix.

Example 1:

Input: mat = [[1,2,3],[4,5,6],[7,8,9]], k = 1
Output: [[12,21,16],[27,45,33],[24,39,28]]

Example 2:

Input: mat = [[1,2,3],[4,5,6],[7,8,9]], k = 2
Output: [[45,45,45],[45,45,45],[45,45,45]]

Constraints:

- m == mat.length
- n == mat[i].length
- 1 <= m, n, k <= 100
- 1 <= mat[i][j] <= 100

Que3. Find the longest substring of a string containing `k` distinct characters

Given a string and a positive number `k`, find the longest substring of the string containing `k` distinct characters. If `k` is more than the total number of distinct characters in the string, return the whole string.

The problem differs from the problem of finding the longest subsequence with `k` distinct characters. Unlike subsequences, **substrings** are required to occupy consecutive positions within the original string.

For example, consider string `abcbdbdbbdcdabd`.

```
For k = 2, o/p is ‘bdbdbbd’
For k = 3, o/p is ‘bcbdbdbbdcd’
For k = 5, o/p is ‘abcbdbdbbdcdabd’
```

Que4. Find the number of 1’s in a sorted binary array

Given a sorted binary array, efficiently count the total number of 1’s in it.

For example,

```
Input:  nums[] = [0, 0, 0, 0, 1, 1, 1]

Output: The total number of 1’s present is 3


Input:  nums[] = [0, 0, 1, 1, 1, 1, 1]

Output: The total number of 1’s present is 5
```

Que5. A Diophantine equation is a polynomial equation, usually in two or more unknowns, such that only the integral solutions are required. An Integral solution is a solution such that all the unknown variables take only integer values.
Given three integers a, b, c representing a linear equation of the form : ax + by = c. Determine if the equation has a solution such that x and y are both integral values.

Examples:

```
Input : a = 3, b = 6, c = 9
Output: Possible
```

```
Explanation : The Equation turns out to be,
3x + 6y = 9 one integral solution would be
x = 1 , y = 1
Input : a = 3, b = 6, c = 8
Output : Not Possible
Explanation : o integral values of x and y
exists that can satisfy the equation 3x + 6y = 8
Input : a = 2, b = 5, c = 1
Output : Possible
Explanation : Various integral solutions
possible are, (-2,1) , (3,-1) etc.
```