1. Given two arrays a[] and b[] of size n and m respectively. The task is to find union between these two arrays. Union of the two arrays can be defined as the set containing distinct elements from both the arrays. If there are repetitions, then only one occurrence of element should be printed in the union.
Example 1:
Input:
5 3
1 2 3 4 5
1 2 3
Output: 5
Explanation: 1, 2, 3, 4 and 5 are the elements which comes in the union set of both arrays. So count is 5.

Example 2:
Input:
6 2
85 25 1 32 54 6
85 2
Output: 7
Explanation: 85, 25, 1, 32, 54, 6, and 2 are the elements which comes in the union set of both arrays. So count is 7.

Your Task:
Complete doUnion funciton that takes a, n, b, m as parameters and returns the count of union elements of the two arrays. The printing is done by the driver code.

Constraints:
1 ≤ n, m ≤ 10^5
0 ≤ a[i], b[i] < 10^5

Elements are not necessarily distinct.
Expected Time Complexity : O((n+m)log(n+m))
Expected Auxilliary Space : O(n+m)
==========
2.Given an array, rotate the array by one position in clock-wise direction.

Example 1:
Input:
N = 5
A[] = {1, 2, 3, 4, 5}
Output:
5 1 2 3 4

Example 2:
Input:
N = 8
A[] = {9, 8, 7, 6, 4, 2, 1, 3}
Output:
3 9 8 7 6 4 2 1

Your Task:
You don't need to read input or print anything. Your task is to complete the function rotate() which takes the array A[] and its size N as inputs and modify the array in place.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(1)

Constraints:
1<=N<=10^5
0<=a[i]<=10^5
==========
3. Given an array of N integers, and an integer K, find the number of pairs of elements in the array whose sum is equal to K.

Example 1:

Input:
N = 4, K = 6
arr[] = {1, 5, 7, 1}
Output: 2
Explanation:
arr[0] + arr[1] = 1 + 5 = 6
and arr[1] + arr[3] = 5 + 1 = 6.

Example 2:

Input:
N = 4, K = 2
arr[] = {1, 1, 1, 1}
Output: 6
Explanation:
Each 1 will produce sum 2 with any 1.

Your Task:
You don't need to read input or print anything. Your task is to complete the function getPairsCount() which takes arr[], n and k as input parameters and returns the number of pairs that have sum K.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(N)

Constraints:
1 <= N <= 10^5
1 <= K <= 10^8
1 <= Arr[i] <= 10^6

==========

4. Given an array of N integers arr[] where each element represents the max length of the jump that can be made forward from that element. Find the minimum number of jumps to reach the end of the array (starting from the first element). If an element is 0, then you cannot move through that element.

Note: Return -1 if you can't reach the end of the array.

Example 1:

Input:
N = 11
arr[] = {1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9}
Output: 3
Explanation:
First jump from 1st element to 2nd
element with value 3. Now, from here
we jump to 5th element with value 9,
and from here we will jump to the last.

Example 2:

Input :
N = 6
arr = {1, 4, 3, 2, 6, 7}
Output: 2
Explanation:
First we jump from the 1st to 2nd element
and then jump to the last element.

Your task:
You don't need to read input or print anything. Your task is to complete function minJumps() which takes the array arr and it's size N as input parameters and returns the minimum number of jumps. If not possible return -1.

Expected Time Complexity: O(N)
Expected Space Complexity: O(1)

Constraints:
1 ≤ N ≤ 10^7
0 ≤ arr ≤ 10^7

==========

5. Given an array of N integers where each value represents the number of chocolates in a packet. Each packet can have a variable number of chocolates. There are m students, the task is to distribute chocolate packets such that:

- Each student gets one packet.
- The difference between the number of chocolates in the packet with maximum chocolates and the packet with minimum chocolates given to the students is minimum.

Examples:
Input : arr[] = {7, 3, 2, 4, 9, 12, 56} , m = 3
Output: Minimum Difference is 2
Explanation:

We have seven packets of chocolates and we need to pick three packets for 3 students.
If we pick 2, 3 and 4, we get the minimum difference between maximum and minimum packet sizes.

Input : arr[] = {3, 4, 1, 9, 56, 7, 9, 12} , m = 5
Output: Minimum Difference is 6
==========
6. Print all distinct permutations of a string having duplicates.
Input: ABCA
Output: AABC AACB ABAC ABCA ACBA ACAB BAAC BACA BCAA CABA CAAB CBAA
==========
7. Given two dimensional matrix of integer and print the rectangle can be formed using given indices and also find the sum of the elements in the rectangle.

Input:
mat[M][N] = {{1, 2, 3, 4, 6},
{5, 3, 8, 1, 2},
{4, 6, 7, 5, 5},
{2, 4, 8, 9, 4}};
index = (2, 0) and (3, 4)
Output:
Rectangle = {4 6 7 5 5}{2 4 8 9 4}
Sum=54
==========
8. Find the next greater element for each element in given array.

Input: array[]={6, 3, 9, 10, 8, 2, 1, 15, 7};
Output: {7, 5, 10, 15, 9, 3, 2, _, 8}
==========
9. Given a binary string of length ‘N’, our task is to count the total number of possible substrings in which the count of 1s is strictly greater than that of 0s.
Note- The count of 1s must be greater than 0s.

Input
Length of the string (N) = 5
Given string = “11010”

Output
8

Explanation
The substrings which have a count of 1s are strictly greater than the count of 0s are “1”, “1”,”1”, “11”, “110”, “11010”, “101”, and “1101”.
==========
10. Find all occurrences of the given string in a character matrix
Given an M × N matrix of characters, find all occurrences of a given string in the matrix. We are allowed to search the string in all eight possible directions, i.e., North, West, South, East, North-East, North-West, South-East, South-West. Note that there should not be any cycles in the output path.
For example, consider the following matrix of characters,
[D E M X B]
[A O E P E]
[D D C O D]
[E B E D S]
[C P Y E N]
If the given string is CODE, following are all its occurrences in the matrix:
C(2, 2) O(1, 1) D(0, 0) E(0, 1)
C(2, 2) O(1, 1) D(2, 0) E(3, 0)
C(2, 2) O(1, 1) D(2, 1) E(1, 2)
C(2, 2) O(1, 1) D(2, 1) E(3, 0)
C(2, 2) O(1, 1) D(2, 1) E(3, 2)
C(2, 2) O(2, 3) D(2, 4) E(1, 4)
C(2, 2) O(2, 3) D(3, 3) E(3, 2)
C(2, 2) O(2, 3) D(3, 3) E(4, 3)
==========