**Question 1.**

Write a program to reverse an array or string

Given an array (or string), the task is to reverse the array/string.

Examples :

Input : arr[] = {1, 2, 3}

Output : arr[] = {3, 2, 1}

Input : arr[] = {4, 5, 1, 2}

Output : arr[] = {2, 1, 5, 4}

**Question 2.**

Maximum and minimum of an array using minimum number of comparisons

Examples:

Given an array of size N. The task is to find the maximum and the minimum element of the array using the minimum number of comparisons.

Input: arr[] = {3, 5, 4, 1, 9}

Output: Minimum element is: 1

Maximum element is: 9

Input: arr[] = {22, 14, 8, 17, 35, 3}

Output: Minimum element is: 3

Maximum element is: 35

**Question 3.**

Kth smallest element

Given an array arr[] and an integer K where K is smaller than size of array, the task is to find the Kth smallest element in the given array. It is given that all array elements are distinct.

Note :- l and r denotes the starting and ending index of the array.

Example 1:

Input:

N = 6

arr[] = 7 10 4 3 20 15

K = 3

Output : 7

Explanation :

3rd smallest element in the given

array is 7.

Example 2:

Input:

N = 5

arr[] = 7 10 4 20 15

K = 4

Output : 15

Explanation :

4th smallest element in the given

array is 15.

Your Task:

You don't have to read input or print anything. Your task is to complete the function kthSmallest() which takes the array arr[], integers l and r denoting the starting and ending index of the array and an integer K as input and returns the Kth smallest element.

Expected Time Complexity: O(n)

Expected Auxiliary Space: O(log(n))

Constraints:

1 <= N <= 105

1 <= arr[i] <= 105

1 <= K <= N

**Question 4.**

Sort an array of 0s, 1s and 2s

Given an array of size N containing only 0s, 1s, and 2s; sort the array in ascending order.

Example 1:

Input:

N = 5

arr[]= {0 2 1 2 0}

Output:

0 0 1 2 2

Explanation:

0s 1s and 2s are segregated

into ascending order.

Example 2:

Input:

N = 3

arr[] = {0 1 0}

Output:

0 0 1

Explanation:

0s 1s and 2s are segregated

into ascending order.

Your Task:

You don't need to read input or print anything. Your task is to complete the function sort012() that takes an array arr and N as input parameters and sorts the array in-place.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

1 <= N <= 10^6

0 <= A[i] <= 2

**Question 5.**

Move all negative numbers to beginning and positive to end with constant extra space

An array contains both positive and negative numbers in random order. Rearrange the array elements so that all negative numbers appear before all positive numbers.

Examples :

Input: -12, 11, -13, -5, 6, -7, 5, -3, -6

Output: -12 -13 -5 -7 -3 -6 11 6 5

Note: Order of elements is not important here.

**Question 6.**

Union of two arrays

Given two arrays a[] and b[] of size n and m respectively. The task is to find the number of elements in the 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.

Note : Elements are not necessarily distinct.

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 ≤ 105

0 ≤ a[i], b[i] < 105

Expected Time Complexity : O(n+m)

Expected Auxilliary Space : O(n+m)

**Question 7.**

Cyclically rotate an array by one

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<=105

0<=a[i]<=105

**Question 8.**

Kadane's Algorithm

Given an array Arr[] of N integers. Find the contiguous sub-array(containing at least one number) which has the maximum sum and return its sum.

Example 1:

Input:

N = 5

Arr[] = {1,2,3,-2,5}

Output:

9

Explanation:

Max subarray sum is 9

of elements (1, 2, 3, -2, 5) which

is a contiguous subarray.

Example 2:

Input:

N = 4

Arr[] = {-1,-2,-3,-4}

Output:

-1

Explanation:

Max subarray sum is -1

of element (-1)

Your Task:

You don't need to read input or print anything. The task is to complete the function maxSubarraySum() which takes Arr[] and N as input parameters and returns the sum of subarray with maximum sum.

Expected Time Complexity: O(N)

Expected Auxiliary Space: O(1)

Constraints:

1 ≤ N ≤ 106

-107 ≤ A[i] ≤ 107

**Question 9.**

Minimize the Heights II

Given an array arr[] denoting heights of N towers and a positive integer K.

For each tower, you must perform exactly one of the following operations exactly once.

Increase the height of the tower by K

Decrease the height of the tower by K

Find out the minimum possible difference between the height of the shortest and tallest towers after you have modified each tower.

You can find a slight modification of the problem here.

Note: It is compulsory to increase or decrease the height by K for each tower. After the operation, the resultant array should not contain any negative integers.

Example 1:

Input:

K = 2, N = 4

Arr[] = {1, 5, 8, 10}

Output:

5

Explanation:

The array can be modified as

{1+k, 5-k, 8-k, 10-k} = {3, 3, 6, 8}.

The difference between

the largest and the smallest is 8-3 = 5.

Example 2:

Input:

K = 3, N = 5

Arr[] = {3, 9, 12, 16, 20}

Output:

11

Explanation:

The array can be modified as

{3+k, 9+k, 12-k, 16-k, 20-k} -> {6, 12, 9, 13, 17}.

The difference between

the largest and the smallest is 17-6 = 11.

Your Task:

You don't need to read input or print anything. Your task is to complete the function getMinDiff() which takes the arr[], n, and k as input parameters and returns an integer denoting the minimum difference.

Expected Time Complexity: O(N\*logN)

Expected Auxiliary Space: O(N)

Constraints

1 ≤ K ≤ 104

1 ≤ N ≤ 105

1 ≤ Arr[i] ≤ 105

**Question 10.**

**Question 11.**

**Question 12.**

**Question 13.**

**Question 14.**

**Question 15.**

**Question 16.**

**Question 17.**

**Question 18.**

**Question 19.**

**Question 20.**