**Arranging the array :-**

Medium Accuracy: 37.41% Submissions: 56K+ Points: 4

You are given an array of size **N**. Rearrange the given array in-place such that all the negative numbers occur before all non-negative numbers.  
(Maintain the order of all -ve and non-negative numbers as given in the original array).  
  
**Example 1:**

**Input:**

N = 4

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

**Output:**

-3 -2 3 2

**Explanation:**

In the given array, negative numbers

are -3, -2 and non-negative numbers are 3, 2.

**Example 2:**

**Input:**

N = 4

Arr[] = {-3, 1, 0, -2}

**Output:**

-3 -2 1 0

**Explanation:**

In the given array, negative numbers

are -3, -2 and non-negative numbers are 1, 0.

**Your Task:**  
You don't need to read input or print anything. Your task is to complete the function **Rearrange()** which takes the array **Arr[]** and its size **N**as inputs and returns the array after rearranging with spaces between the elements of the array.  
  
**Expected Time Complexity:** O(N. Log(N))  
**Expected Auxiliary Space:** O(Log(N))

**Constraints:**  
1 ≤ N ≤ 105  
-109 ≤ Elements of array ≤ 109

**Code :-**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

#define ll long long

// } Driver Code Ends

**//Using single storage(queue)**

class Solution

{

public:

void Rearrange(int arr[], int n){

if(n==1) return;

queue<long long int> q;

for(auto i=0; i<n; i++){

if(arr[i]<0)

q.push(arr[i]);

}

int i=0;

while(q.empty()==false){

long long int front = q.front();

q.pop();

if(arr[i]>=0)

q.push(arr[i]);

arr[i] = front;

i++;

}

return;

}

};

**//using separate storage for –ve & +ve integers**

class Solution

{

public:

void Rearrange(int arr[], int n){

if(n==1) return;

vector<int> pos, neg;

for(auto i=0; i<n; i++){

if(arr[i]>=0) pos.push\_back(arr[i]);

else neg.push\_back(arr[i]);

}

int i=0;

for(auto item:neg){

arr[i] = item;

i++;

}

for(auto item:pos){

arr[i] = item;

i++;

}

return;

}

};

//{ Driver Code Starts.

void Rearrange(int arr[], int n);

int main()

{

int t;

cin>>t;

while(t--)

{

int n;

cin>>n;

int arr[n];

for(int i=0;i<n;i++)

cin>>arr[i];

long long j=0;

Solution ob;

ob.Rearrange(arr, n);

for (int i = 0; i < n; i++)

cout << arr[i] << " ";

cout << endl;

}

return 0;

}

// } Driver Code Ends

**T.C :- O(n)**

**S.C :- O(n)**