WEEK 2:

Day 1-05/04/2022

1.Given an array, the task is to reverse the array

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

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

```
#include<iostream>
using namespace std;
int main()
{
int n;
cin>>n;
int arr[n];
for(int i=0;i<n;i++)
{
cin>>arr[i];
}
for(int i=n-1;i>=0;i--)
{
cout<<arr[i]<< " ";
}
return 0;
```

}

2. **Write a program to return minimum and maximum in an array.**

Input:

N = 6

A[] = {3, 2, 1, 56, 10000, 167}

Output:

min = 1, max = 10000

```
// { Driver Code Starts
#include <bits/stdc++.h>
using namespace std;
#define ll long long

pair<long long, long long> getMinMax(long long a[], int n) ;

int main() {
    int t;
    cin >> t;
    while (t--) {
        int n;
        cin >> n;
```

```
        ll a[n];
        for (int i = 0; i < n; i++) cin >> a[i];

        pair<ll, ll> pp = getMinMax(a, n);

        cout << pp.first << " " << pp.second << endl;
    }
    return 0;
}// } Driver Code Ends


pair<long long, long long> getMinMax(long long a[], int n)
{
    long long int max=INT_MIN;
    long long int min=INT_MAX;

    for(int i=0;i<n;i++)
    {
        if(a[i]>max)
        {
            max=a[i];
        }
```

```
        if(a[i]<min)
        {
            min=a[i];
        }
    }
    pair <int,int> p;
    p.first=min;
    p.second=max;
    return p;

}
```

3. 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.

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

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

```
// { Driver Code Starts
#include<bits/stdc++.h>
using namespace std;
#define ll long long

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;

      Rearrange( arr, n);

      for (int i = 0; i < n; i++)
         cout << arr[i] << " ";
      cout << endl;
   }
   return 0;
} // } Driver Code Ends
```

```
void Rearrange(int arr[], int n)
{
   vector<int>ans;
   for(int i=0;i<n;i++)
   {
      if(arr[i]<0)
      {
         ans.push_back(arr[i]);
      }
   }
   for(int i=0;i<n;i++)
   {
      if(arr[i]>=0)
      {
         ans.push_back(arr[i]);
      }
   }
   for(int i=0;i<ans.size();i++)
```

```
    {
        arr[i]=ans[i];
    }
}
```

4.You are given a list of n-1 integers and these integers are in the range of 1 to n. There are no duplicates in the list. One of the integers is missing in the list. Write a code to find the missing integer.

Input: arr[] = {1, 2, 3, 4, 6, 7, 8}

Output: 5

Explanation: The missing number from 1 to 8 is 5

```
// { Driver Code Starts
// Initial template for C++

#include <bits/stdc++.h>
using namespace std;




 // } Driver Code Ends
// User function template for C++
```

```
class Solution{
  public:
    int MissingNumber(vector<int>& array, int n)
    {
        int sum=n*((n+1)*0.5);
        for(int i=0;i<n-1;i++)
        {
            sum=sum-array[i];
        }
        return sum;
    }
};

// { Driver Code Starts.

int main() {
    int t;
    cin >> t;
    while (t--) {
        int n;
        cin >> n;
```

```
        vector<int> array(n - 1);
        for (int i = 0; i < n - 1; ++i) cin >> array[i];
        Solution obj;
        cout << obj.MissingNumber(array, n) << "\n";
    }
    return 0;
}  // } Driver Code Ends
```