Day 2-Week 2- 6th April

1. Party of Couples

```
// { 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 findSingle(int N, int arr[])
   {
      int single_digit=0;
      for(int i=0;i<N;i++)
      {
         single_digit=single_digit ^ arr[i];
      }
```

```
        return single_digit;


    }
};

// { Driver Code Starts.
int main()
{
    int t;
    cin>>t;
    while(t--)
    {
        int N, X;
        cin >> N;
        int arr[N];
        for(int i = 0; i < N; i++){
            cin >> arr[i];
        }

        Solution ob;
        cout << ob.findSingle(N, arr) << endl;
    }
    return 0;
}   // } Driver Code Ends
```

2. Cyclically rotate an array by one

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

#include <bits/stdc++.h>
using namespace std;
void rotate(int arr[], int n);

int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int n;
        scanf("%d",&n);
        int a[n] , i;
        for(i=0;i<n;i++)
        scanf("%d",&a[i]);
        rotate(a, n);
        for (i = 0; i < n; i++)
          printf("%d ", a[i]);
        printf("\n");
    }
        return 0;
}
// } Driver Code Ends



//User function Template for C++

void rotate(int arr[], int n)
{
    for(int i=0;i<n;i++)
    swap(arr[i],arr[n-1]);
}
```

3. Segregate 0s and 1s

```
// { 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:
   void segregate0and1(int arr[], int n)
   {
      int start=0;
      int end=n-1;

      while(start<=end)
      {
         if(arr[start]==0)
```

```
            {
               start++;
            }
            else
            {
               swap(arr[start],arr[end--]);
            }
         }
      }
   };

   // { Driver Code Starts.

   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];
        }
        Solution ob;
        ob.segregate0and1(arr, n);
        for (int i = 0; i < n; i++) {
            cout << arr[i] << " ";
        }
        cout << "\n";
    }
    return 0;
}  // } Driver Code Ends
```

4. Kth smallest element

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

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

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


class Solution{
   public:
   // arr : given array
   // l : starting index of the array i.e 0
   // r : ending index of the array i.e size-1
   // k : find kth smallest element and return using this
function
   int kthSmallest(int arr[],int l,int r, int k)
   {
      vector<int>v;
      for(int i=l;i<=r;i++)
      {
         v.push_back(arr[i]);
      }
      sort(v.begin(),v.end());
      return v[k-1];
   }
};
```

```
// { Driver Code Starts.

int main()
{
   int test_case;
   cin>>test_case;
   while(test_case--)
   {
      int number_of_elements;
      cin>>number_of_elements;
      int a[number_of_elements];

      for(int i=0;i<number_of_elements;i++)
         cin>>a[i];

      int k;
      cin>>k;
      Solution ob;
      cout<<ob.kthSmallest(a, 0, number_of_elements-1,
k)<<endl;
```

```
    }
    return 0;
}  // } Driver Code Ends
```