Day-1

level-Easy

Q1 Remove element

```
arr[] = [3,2,2,3];  val = 3  , output: 2
int ~~t~~ s=0 { starting index }
// sort the given array

sort(arr, n);

// check each index with given val

if (arr[i] != val)
{
    arr[s++] = arr[i];
}

return s;
```

Q2 Search-Insert position

```
arr[1,3,5,6], target = 5
// if target is found return index if not then find
where it would be if it were in sorted in order.

// Using binary search

/ find mid

if (mid == target)
    return mid
if (mid > target)
    high = mid-1;
if (mid < target)
    low = mid+1;    → return low;
```

Note : totally taken hint from google, not able to create logic

Medium

Q1 Array with elements Not equal to Avg of neighbours.

// sort the given array by using STL library

```
sort(nums.begin(), nums.end());
```

// create two loop
// one for increment by 2
// one for fill the alternate position
//

1 4 2 5 3 (like that)

/ create Vector

```
Vector<int> res(n);
for(i=0,K=0;i<n;i+=2,K++)
  {
     res[i]=num[k];
  }
for(i=1;i<n;i+=2,K++)
  {
     res[i]=nom[k];
  }
  ~~return res;~~
}
```