Smallest Subarray with sum greater than a given value

```
int smallestSubWithSum(int arr[], int n, int x)
   {
      int Windowsize = INT_MAX;
      int start = 0,end = 0,sum = 0;;
      for(end=0;end<n;end++){
         sum += arr[end];
         while(sum>x){
               if(sum>x){
               Windowsize = min(Windowsize,end-start+1);
               sum -= arr[start++];
            }
         }
      }
      if(Windowsize==INT_MAX)
      return 0;

      return Windowsize;
   }
```

Count More than n/k Occurences

```
int countOccurence(int arr[], int n, int k) {
      int x = n / k,count=0;
      map<int, int> map;

   for (int i = 0; i < n; i++) {
      map[arr[i]]++;
   }
   for (auto i : map) {
      if (i.second > x) {
         count++;
      }
   }
   return count;
   }
```

Maximum Product Subarray

```
long long maxProduct(vector<int> arr, int n) {
            long long maxi = INT_MIN;
      long long prod=1;
```

```
for(int i=0;i<n;i++)
{
  prod*=arr[i];
  maxi=max(prod,maxi);
  if(prod==0)
   prod=1;
}
prod=1;
for(int i=n-1;i>=0;i--)
{
  prod*=arr[i];

  maxi=max(prod,maxi);
  if(prod==0)
   prod=1;
}
return maxi;
   }
```