Talent Battle 100 Days Coding Series

You are given **N** integers. In each step you can choose some **K** of the remaining numbers and delete them, if the following condition holds: Let the **K** numbers you've chosen be a_1, a_2, a_3, ..., a_K in sorted order. Then, for each $i \leq K - 1$, a_{i+1} must be greater than or equal to $a_i * C$.

You are asked to calculate the maximum number of steps you can possibly make.

Input

- The first line of the input contains an integer **T**, denoting the number of test cases. The description of each testcase follows.
- The first line of each testcase contains three integers: **N**, **K**, and **C**
- The second line of each testcase contains the **N** initial numbers

Output

For each test case output the answer in a new line.

Sample Input

2

6 3 2

4 1 2 2 3 1

6 3 2

1 2 2 1 4 4

Sample Output

1

2

C++

```
#include <bits/stdc++.h>

#define ll long long

#define INF 1e18

using namespace std;
```

```
vector<ll> nums;
int T, N, K, C;
int lo, mid, hi, X;

bool check(int L) {
        vector<ll> v[L];
        int i=0;

        for(auto num: nums) {
                if(v[i].size()==K) {
                        break;
                }
                if(v[i].size()==0 || ( v[i].back()<INF/C && v[i].back()*C<=num)) {
                        v[i].push_back(num);
                        i = (i+1)%L;
                }
        }
        return v[L-1].size() == K;
}

int main() {

        ll num;
        cin>>T;
        while(T--) {
                cin>>N>>K>>C;
                nums.clear();
                while(N--) {
                        cin>>num;
```

```
                nums.push_back(num);
            }

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

            X = 0;
            lo = 1;
            hi = nums.size()/K;

            while(lo<=hi) {
                mid = (lo+hi)/2;
                if(check(mid)) {
                    X = mid;
                    lo = mid+1;
                } else {
                    hi = mid-1;
                }
            }
            cout<<X<<endl;
        }
        return 0;
}
```


Java

```
import java.util.*;
```

```
import java.lang.*;
import java.io.*;
class Main{
  public static void main(String[] args){
    Scanner sc=new Scanner(System.in);
    PrintWriter pw=new PrintWriter(System.out);
    int t=sc.nextInt();
    sc.nextLine();
    while(t-->0){
      int n=sc.nextInt();
      int k=sc.nextInt();
      long c=sc.nextInt();
      long a[]=new long[n];
      for(int i=0;i<n;i++)a[i]=sc.nextLong();
      Arrays.sort(a);
      int min=0;
      int max=n/k;
      int ans=0;
      while(min<=max){
        int mid=min+(max-min)/2;
        if(check(a,mid,k,c)){
          ans=mid;
          min=mid+1;
        }
        else
          max=mid-1;
      }
      pw.println(ans);
    }
```

```
  pw.close();
 }
 static boolean check(long[] a,int x,int k,long c){
  if(k*x>a.length) return false;
  if(x==0) return true;
  long v[][]=new long[k][x];
  for(int i=0;i<x;i++) v[0][i]=a[i];
  int s=x;
  for(int i=1;i<k;i++){
   for(int j=0;j<x;j++){
    boolean flag=false;
    while(s<a.length){
     if(a[s]>=c*v[i-1][j]){
      v[i][j]=a[s];
      s++; flag=true;
      break;
     }
     s++;
    }
     if(!flag) return false;
   }
  }

  return true;
 }
}
```