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>

using namespace std;


#define ll long long

#define ar array
```

```
const int mxN=3e5;
const ll INF=2e18;
int n, k, c;
ll a[mxN];

ll mul(ll x, ll y) {
	if ((long double)x*y>=INF)
		return INF;
	return min(x*y, INF);
}

bool ok(int x) {
	vector<ll> v(a, a+x);
	int ind=x;
	for (int r=1; r<k; ++r) {
		for (ll& i : v) {
			ll need=mul(i, c);
			while(ind<n&&a[ind]<need)
				++ind;
			if (ind==n)
				return 0;
			i=a[ind++];
		}
	}
	return 1;
}

void solve() {
	cin >> n >> k >> c;
```

```
        for (int i=0; i<n; ++i)
                cin >> a[i];
        sort(a, a+n);
        int lb=0, rb=n/k;
        while(lb<rb) {
                int mid=(lb+rb+1)/2;
                if (ok(mid))
                        lb=mid;
                else
                        rb=mid-1;
        }
        cout << lb << "\n";
}

int main() {
        ios::sync_with_stdio(0);
        cin.tie(0);
        int t;
        cin >> t;
        while(t--)
                solve();
        return 0;
}
```

Java

```
import java.util.*;
import java.lang.*;
import java.io.*;
```

```
class Main
{
   static boolean isPoss(int x,long []arr,int k,int c)
   {
      ArrayList<ArrayList<Long>> list=new ArrayList<>();
      int cur=0,n=arr.length;
      for(int i=0;i<x;i++)
      {
         list.add(new ArrayList<Long>());
      }
      for(int i=0;i<n;i++)
      {
         cur=cur%x;
         int sz=list.get(cur).size()-1;
         if(sz<0||list.get(cur).get(sz)*c<=arr[i])
         {
            list.get(cur).add(arr[i]);
            cur=(cur+1)%x;
         }
      }
      if(list.get(x-1).size()>=k)
       return true;

      return false;
   }

   static long divset(long []arr,int k,int c)
   {
      int n=arr.length;
```

```
        int l=1,r=n;//To avoid zero x
        int res=0;
        Arrays.sort(arr);
        while(l<=r)
        {
           int mid=l+(r-l)/2;

           if(isPoss(mid,arr,k,c)){
            l=mid+1;
            res=mid;
           }
           else
            r=mid-1;
        }
        return res;
    }

            public static void main (String[] args) throws java.lang.Exception
            {
                    BufferedReader bf=new BufferedReader(new InputStreamReader(System.in));
int t=Integer.parseInt(bf.readLine());
StringBuffer str=new StringBuffer("");
while(t-->0)
{
  String s[]=bf.readLine().trim().split("\\s+");
  int n=Integer.parseInt(s[0]);
  int k=Integer.parseInt(s[1]);
  int c=Integer.parseInt(s[2]);
  long arr[]=new long[n];
```

```
 s=bf.readLine().trim().split("\\s+");
 for(int i=0;i<n;i++)
  arr[i]=Long.parseLong(s[i]);


 str.append(divset(arr,k,c)+"\n");
}
System.out.println(str);
          }
}
```

Python

```
from sys import stdin
def check(x):
   ls = [[] for _ in range(x)]; cIND = 0
   for i in range(n):
      if len(ls[cIND])==k:
         return True
      if len(ls[cIND])==0 or ls[cIND][-1]*c<=a[i]:
         ls[cIND].append(a[i])
         cIND = (cIND+1)%x
   if len(ls[cIND])==k:
      return True
   return False

for _ in range(int(input())):
   n,k,c = map(int,stdin.readline().split())
   a = list(map(int,stdin.readline().split()))
   a.sort(); l = 0; r = n+1
   while r-1>l:
```

```
    mid = (l+r)//2
    if check(mid):
        l = mid
    else:
        r = mid
print(l)
```