**Numbers with prime frequencies greater than or equal to k**

Submissions: [909](https://practice.geeksforgeeks.org/problem_submissions.php?pid=3044)  Accuracy:

25.69%

   Difficulty: [Easy](https://practice.geeksforgeeks.org/Easy/0/0/)   Marks: 2

Show Topic Tags   

You are given an array of size N. You need to find elements which appear prime number of times in the array with minimum K frequency (frequency >= K).

**Input:**  
The first line of the input contains a single integer T, denoting the number of test cases. Then T test case follows. Each testcase contains two lines of input:-  
The size of the array N and minimum frequency K separated by a space.  
The elements of the array separated by spaces.

**Output:**  
For each testcase, print the elements that have prime frequency with their frequency >=K. Print the output in sorted order. If there are no such elements then print -1.

**Constraints:**  
1<=T<=105  
1<=N<=1000  
1<=K<=10  
1<=A[i]<=10000

**Example:**

**Input:**  
2  
13 2  
11 11 11 23 11 37 51 37 37 51 51 51 51  
3 1  
11 22 33

**Output:**  
37 51  
-1

**Explanation:**  
For testcase 1: 11's count is 4, 23 count 1, 37 count 3, 51 count 5. 37 and 51 are two number that appear prime number of time and frequencies greater than or equal to K=2.  
For testcase 2: K=1, and all three elements occur extactly 1 times. Unfortunately, 1 is not prime, so we don't print any element. We, instead, print a -1.

\*\* For More Input/Output Examples Use ['Expected Output'](https://practice.geeksforgeeks.org/problems/numbers-with-prime-frequencies-greater-than-or-equal-to-k/0#ExpectOP) option \*\*

Contributor: Ankit Sharma  
[Author: Soul\_xhacker](https://auth.geeksforgeeks.org/user/Soul_xhacker/practice/)

<https://practice.geeksforgeeks.org/problems/numbers-with-prime-frequencies-greater-than-or-equal-to-k/0>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp3

{

class Program

{

static bool EsPrimo(int n)

{

// Corner cases

if (n <= 1)

return false;

if (n <= 3)

return true;

// This is checked so

// that we can skip

// middle five numbers

// in below loop

if (n % 2 == 0 ||

n % 3 == 0)

return false;

for (int i = 5;

i \* i <= n; i = i + 6)

if (n % i == 0 ||

n % (i + 2) == 0)

return false;

return true;

}

static HashSet<int> hash\_primos =

new HashSet<int>();

static void PrimeFrequencies(int[] arr, int k)

{

int max = arr.Max();

int[] count = new int[max + 1];

for (int i = 0; i < arr.Length; i++)

{

count[arr[i]]++;

}

bool flag = false;

for (int i = 0; i < max + 1; i++)

{

if (hash\_primos.Contains(count[i]))

{

Console.Write(i + " ");

flag = true;

}

}

if (!flag)

{

Console.Write(-1);

}

Console.WriteLine();

}

static void Main(string[] args)

{

//List<int> primos = new List<int>();

for (int i = 0; i < 10001; i++)

{

if (EsPrimo(i))

{

hash\_primos.Add(i);

}

}

//foreach (int item in hash\_primos)

//{

// Console.Write(item + " ");

//}

//int[] arr = { 1111,2222,3333,4444 };

//PrimeFrequencies(arr, 2);

//int[] arr = { 11, 11, 11, 23, 11, 37, 51, 37, 37, 51, 51, 51, 51 };

//PrimeFrequencies(arr, 2);

//int[] arr = { 9851, 3682, 3560, 3525, 1369, 1875, 6158, 7825, 8391, 6629, 264, 3423, 9697, 2166, 5576, 2120, 8790, 1758, 2931, 9483 };

//PrimeFrequencies(arr, 4);

int t = int.Parse(Console.ReadLine());

while (t-- > 0)

{

string[] input = Console.ReadLine().Trim().Split(' ');

int n = int.Parse(input[0]);

int k = int.Parse(input[1]);

int[] arr = Array.ConvertAll(Console.ReadLine().Trim().Split(' '), e => int.Parse(e));

PrimeFrequencies(arr, k);

}

Console.ReadLine();

}

}

}