COMPETITIVE CODING
with
RAMAN CLASSES

... An interactive series for those willing to learn and code

Problem : Longest Subsequence LCM

Topic	: **Arrays**
Difficulty	: **EASY**
Programming Language	: **C++**
Time to Spend	: **25 min**

Problem Statement

Problem:

Given an array a[] of n elements and a positive integer k. The task is to find the longest sub-sequence in the array having LCM(Least Common Multiple) at most k. Print the LCM and the length of the sub-sequence, following the indexes (starting from 0) of the elements of the obtained sub-sequence. Print -1 if it is not possible to do so.

For example: if array a=[2,3,4,5] and k=14

Output: LCM = 12, Length = 3
Indexes = 0 1 2

Problem Statement

Input :

```
4
2 3 4 5
14
```

Output :

```
LCM = 12, Length = 3
Indexes = 0 1 2
```

Input Description :

The first line contains an integer n denoting the number of elements.

The next line contains n space-separated denoting the array.

The last line contains an integer k denoting the maximum value for LCM.

Let Us Revise

In order to solve this problem, go through the following concepts.

1. Unordered Map
2. LCM
3. What is Subsequence?

Problem Description

We are given an array a of n integers and we need to find the longest subsequence possible whose LCM is less than the given value k.

Subsequence is same as subarray but not contiguous.

LCM of two integers is Lowest Common Multiple of both integers.

If there is no subsequence present then print “-1” else print the LCM and length of subsequence in one line and the indexes of the integers in the subsequence in another.

Let Us Think

In order to solve this problem, let us think and analyse how to get started with this problem.

Try to draw different examples on paper. On careful observation, you can see to solve this problem you need subsequence with maximum number of elements and LCM of that subsequence < k.

So we can start by finding how many elements from the array results in LCM of [1,k-1] and find the LCM having highest number of elements present.

And that LCM is the answer.

Let Us Think

Now you know the logic. Lets proceed with the code.

Let Us Code

Now you know the logic. Lets proceed with the code.

Things we need to do for this problem:

1. Find the frequency of all the unique elements.

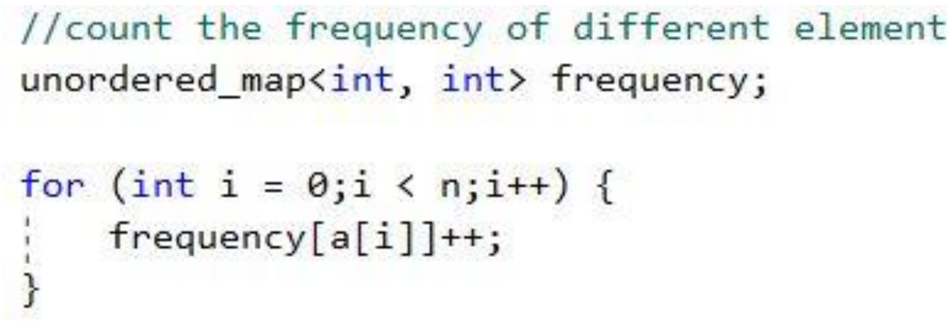
2. Count the number of elements resulting in LCM of 1 to k-1 in an array say A.

3. Find the largest element in the array A whose index represent LCM and value represent Length.

4. Loop to print all the indexes of longest subsequence.

Let Us Code

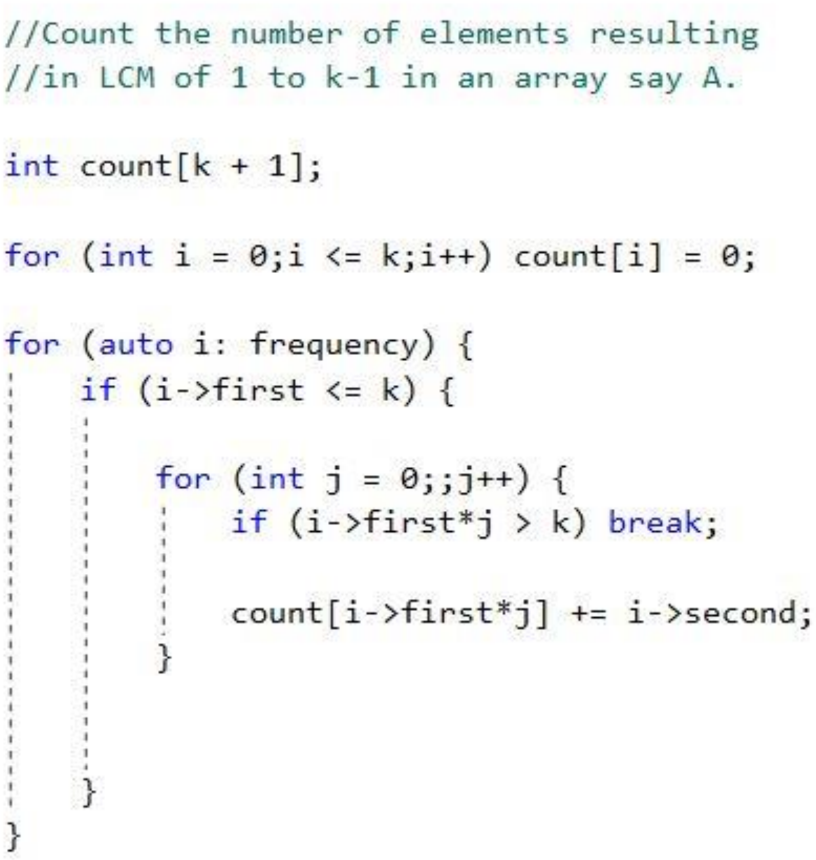
1. Find the frequency of all the unique elements.



```
//count the frequency of different element
unordered_map<int, int> frequency;

for (int i = 0;i < n;i++) {
    frequency[a[i]]++;
}
```

Let Us Code

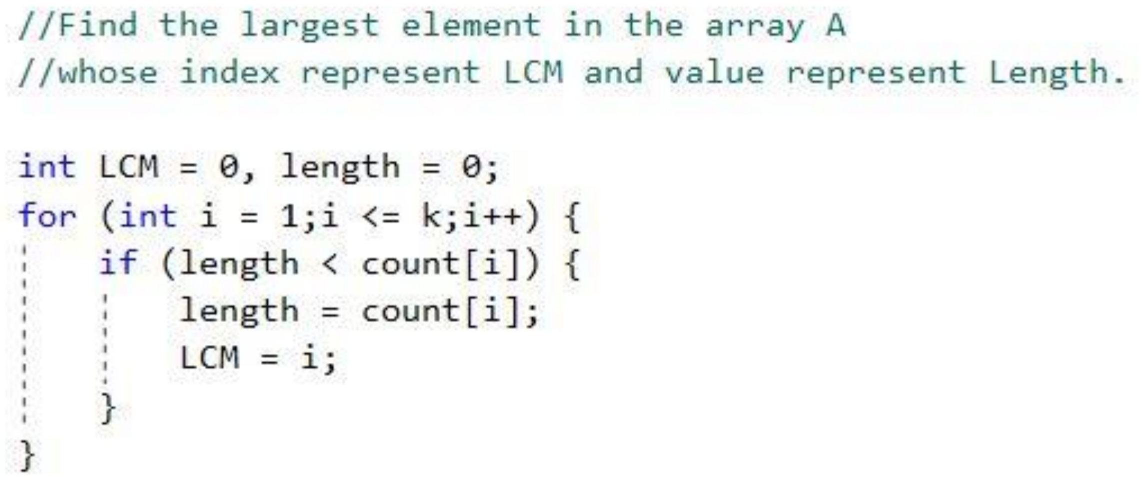
2. Count the number of elements resulting in LCM of 1 to k-1 in an array say A.



```
//Count the number of elements resulting
//in LCM of 1 to k-1 in an array say A.

int count[k + 1];

for (int i = 0;i <= k;i++) count[i] = 0;

for (auto i: frequency) {
    if (i->first <= k) {

        for (int j = 0;;j++) {
            if (i->first*j > k) break;

            count[i->first*j] += i->second;
        }


    }
}
```

Let Us Code

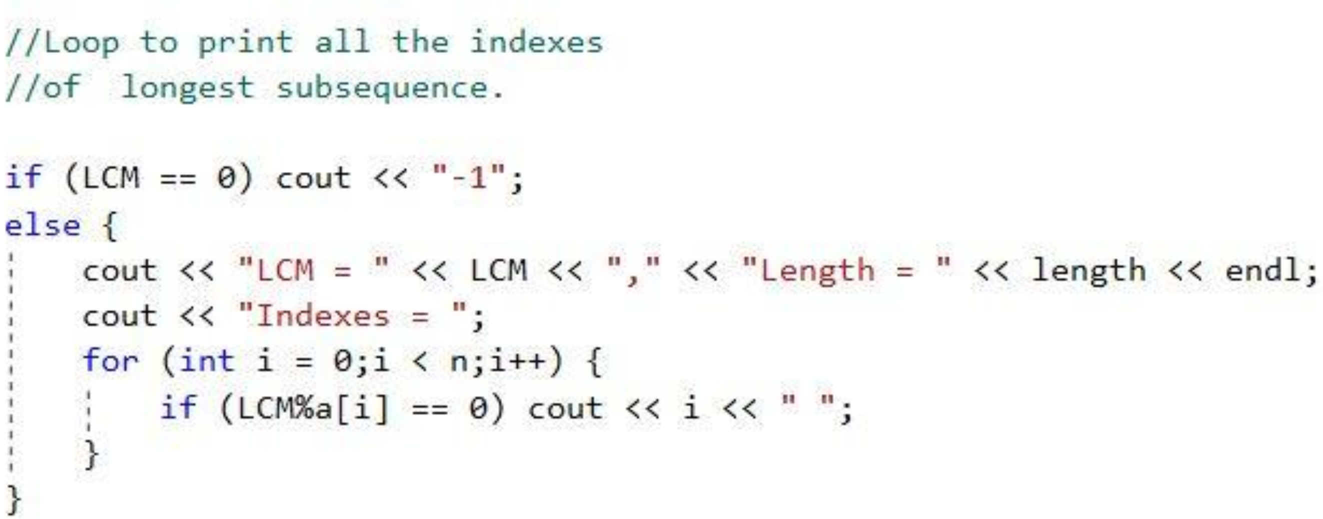
3. Find the largest element in the array A whose index represent LCM and value represent Length.



```
//Find the largest element in the array A
//whose index represent LCM and value represent Length.

int LCM = 0, length = 0;
for (int i = 1;i <= k;i++) {
    if (length < count[i]) {
        length = count[i];
        LCM = i;
    }
}
```

Let Us Code

4. Loop to print all the indexes of longest subsequence.



```cpp
//Loop to print all the indexes
//of  longest subsequence.

if (LCM == 0) cout << "-1";
else {
    cout << "LCM = " << LCM << "," << "Length = " << length << endl;
    cout << "Indexes = ";
    for (int i = 0;i < n;i++) {
        if (LCM%a[i] == 0) cout << i << " ";
    }
}
```

Thank You !

... brought to you by

RAMAN CLASSES

FULFILL YOUR DREAMS OF SUDYING IN IIT's

www.ramanclasses.in