**Frequencies of Limited Range Array Elements: -**

**Easy Accuracy: 27.64% Submissions: 202K+ Points: 2**

Given an array **arr[]** of **N** positive integers which can contain integers from **1 to P** where elements can be repeated or can be absent from the array. Your task is to count the frequency of all numbers from **1 to N**. Make in-place changes in **arr[],** such that **arr[i] = frequency(i).** Assume 1-based indexing.  
**Note:** The elements greater than N in the array can be ignored for counting and **do modify the array** **in-place.**

**Example 1:**

**Input:**

N = 5

arr[] = {2, 3, 2, 3, 5}

P = 5

**Output:**

0 2 2 0 1

**Explanation:**

Counting frequencies of each array element

We have:

1 occurring 0 times.

2 occurring 2 times.

3 occurring 2 times.

4 occurring 0 times.

5 occurring 1 time.

**Example 2:**

**Input:**

N = 4

arr[] = {3,3,3,3}

P = 3

**Output:**

0 0 4 0

**Explanation:**

Counting frequencies of each array element

We have:

1 occurring 0 times.

2 occurring 0 times.

3 occurring 4 times.

4 occurring 0 times.

**Example 3:**

**Input:**

N = 2

arr[] = {8,9}

P = 9

**Output:**

0 0

**Explanation:**

Counting frequencies of each array element

We have:

1 occurring 0 times.

2 occurring 0 times.  
Since here P=9, but there are no 9th Index present so can't count the value.

**Your Task:**  
You don't need to read input or print anything. Complete the function **frequencyCount()**that takes the array **arr,**and integers **n**,and**p** as input parameters and **modify the array** **in-place**to denote the frequency count of each element from **1**to **N.**

**Expected time complexity:** O(N)  
**Expected auxiliay space:** O(1)

**Constraints:**  
1 ≤ N ≤ 105  
1 ≤ P ≤ 4\*104  
1 <= arr[i] <= P

**Code: -**

//{ Driver Code Starts

#include<bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution{

public:

//Function to count the frequency of all elements from 1 to N in the array.

void frequencyCount(vector<int>& arr,int N, int P)

{

// code here

for(auto &i : arr)

i -= 1;

for(int i=0; i<N; ++i){

if(arr[i] % P >= N){

arr[i] = (arr[i] / P) \* P;

continue;

}

arr[ arr[i] % P ] += P;

}

for(auto &i : arr)

i /= P;

}

};

//{ Driver Code Starts.

int main()

{

long long t;

//testcases

cin >> t;

while(t--){

int N, P;

//size of array

cin >> N;

vector<int> arr(N);

//adding elements to the vector

for(int i = 0; i < N ; i++){

cin >> arr[i];

}

cin >> P;

Solution ob;

//calling frequncycount() function

ob.frequencyCount(arr, N, P);

//printing array elements

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

cout << arr[i] << " ";

cout << endl;

}

return 0;

}

// } Driver Code Ends

**T.C : - O(N)**

**S.C : - O(1)**