**Least Prime Factor :-**

Easy Accuracy: 51.45% Submissions: 26K+ Points: 2

Given a number N, find the least prime factors for all numbers from 1 to N.  The least prime factor of an integer X is the smallest prime number that divides it.  
**Note :**

* 1 needs to be printed for 1.
* You need to return an array/vector/list of **size N+1** and need to use **1-based indexing** to store the answer for each number.

**Example 1:**

**Input**: N = 6

**Output:** [0, 1, 2, 3, 2, 5, 2]

**Explanation**: least prime factor of 1 = 1,

least prime factor of 2 = 2,

least prime factor of 3 = 3,

least prime factor of 4 = 2,

least prime factor of 5 = 5,

least prime factor of 6 = 2.

So answer is[1, 2, 3, 2, 5, 2].

**Example 2:**

**Input:** N = 4

**Output:**[0, 1, 2, 3, 2]

**Explanation**: least prime factor of 1 = 1,

least prime factor of 2 = 2,

least prime factor of 3 = 3,

least prime factor of 4 = 2.

So answer is[1, 2, 3, 2].

**Your Task:**  
You don't need to read input or print anything. Complete the function **leastPrimeFactor()**which takes N as input parameter and returns a list of integers containing all the least prime factors of each number from 1 to N.

**Expected Time Complexity:** O(NlogN)  
**Expected Auxiliary Space:** O(N)

**Constraints:**  
2<= n <=105

**Code :-**

//{ Driver Code Starts

// Initial Template for C++

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

// User function Template for C++

class Solution {

public:

vector<int> leastPrimeFactor(int n) {

vector<int> v(n+1, 0);

v[1]=1;

for(int i=2; i<=n; i++){

if(v[i]==0){

v[i] = i;

for(int j=i\*i; j<=n; j+=i){

if(v[j]==0)

v[j] = i;

}

}

}

return v;

}

};

int main() {

int t;

cin >> t;

while (t--) {

int n;

cin >> n;

Solution ob;

vector<int>ans = ob.leastPrimeFactor(n);

for(int i=1;i<=n;i++)cout<<ans[i]<<" ";

cout<<endl;

}

return 0;

}

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

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