**Prime Factors**

Show Topic Tags   

Given a number N, print all its unique prime factors in increasing order.

Input : N = 100

Output: 2 5

Input : N = 35

Output: 5 7

**Input:**  
The first line of input contains an integer T denoting the number of test cases.  
The first line of each test case is N.  
  
**Output:**  
Print all prime factors in increasing order.  
  
**Constraints:**  
1 ≤ T ≤ 200  
2 ≤ N ≤ 10000  
  
**Example:**  
**Input:**  
2  
100  
35  
  
**Output:**  
2 5  
5 7

\*\*For More Examples Use Expected Output\*\*

<http://practice.geeksforgeeks.org/problems/prime-factors/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication243;

import java.io.\*;

import java.util.\*;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication243 {

/\*\*

\* @param args the command line arguments

\*/

public static void primeFactors(int n)

{

HashSet<Integer> hs = new HashSet<Integer>();

// Print the number of 2s that divide n

while (n%2==0)

{

//System.out.print(2 + " ");

hs.add(2);

n /= 2;

}

// n must be odd at this point. So we can

// skip one element (Note i = i +2)

for (int i = 3; i <= Math.sqrt(n); i+= 2)

{

// While i divides n, print i and divide n

while (n%i == 0)

{

//System.out.print(i + " ");

hs.add(i);

n /= i;

}

}

// This condition is to handle the case whien

// n is a prime number greater than 2

if (n > 2) {

//System.out.print(n);

hs.add(n);

}

/\*

Iterator<Integer> it = hs.iterator();

while(it.hasNext()){

System.out.print(it.next()+ " ");

}\*/

List<Integer> res = new ArrayList<Integer>(hs);

Collections.sort(res);

for(int i =0; i<res.size(); i++) {

System.out.print(res.get(i)+ " ");

}

}

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

int n = Integer.parseInt(br.readLine());

primeFactors(n);

System.out.println();

}

}

}