Computer Science & Information Systems

**DESIGN AND ANALYSIS OF ALGORITHMS – Lab Sheet # 5**

1. **Objective**:

The students will be able to implement the string matching algorithms.

1. **Problem:** (**dividing sequence**)

This problem is about sequences of positive integers a1,a2,…,aN. A subsequence of a sequence is anything obtained by dropping some of the elements. For example, 3,7,11,3 is a subsequence of 6,**3,**11,5,**7**,4,3,**11**,5,**3** , but 3,3,7 is not a subsequence of 6,3,11,5,7,4,3,11,5,3.

A *fully dividing sequence* is a sequence a1, a2,…, aN where ai divides aj whenever i < j. For example, 3, 15, 60, 720 is a fully dividing sequence.

Given a sequence of integers your aim is to find the length of the longest fully dividing subsequence of this sequence.

Consider the sequence 2,3,7,8,14,39,145,76,320

It has a fully dividing sequence of length 3, namely 2, 8, 320, but none of length 4 or greater.

Consider the sequence 2,11,16,12,36,60,71,17,29,144,288,129,432,993 .

It has two fully dividing subsequence of length 5,

* **2**,11,16,**12,36**,60,71,17,29,**144,288**,129,432,993 and
* **2**,11,16,**12,36**,60,71,17,29,**144**,288,129,**432**,993

And none of length 6 or greater.

1. **Solution hint**

Let the input be a1, a2, …, aN. Let us define Best(i) to be the length of longest dividing sequence in a1,a2,…ai that includes ai.

Write an expression for Best(i) in terms of Best(j) with j<i, with base case Best(1) =1. Solve this recurrence using dynamic programming. [Full solution](http://www.iarcs.org.in/inoi/contests/oct2004/Advanced-1-solution.php). (*Click to get complete description*)

1. **Input format**

The first line of input contains a single positive integer N indicating the length of the input sequence. Lines 2,…,N+1 contain one integer each. The integer on line i+1 is ai.

1. **Output format**

Your output should consist of a single integer indicating the length of the longest fully dividing subsequence of the input sequence.

1. **Test data**

You may assume that N ≤ 10000.

1. **Sample input**

9

2

3

7

8

14

39

145

76

320

1. **Sample output**

3

1. **Sample input2**

14

2

11

16

12

36

60

71

17

29

144

288

129

432

993

1. **Sample output2**

5

1. **References**

*(IARCS OPC Archive, K Narayan Kumar, CMI)*