**Dynamic Programming 1 :-**

**Question 1 Min Steps To 1**

**Send Feedback**

Given a positive integer n, find the minimum number of steps s, that takes n to 1. You can perform any one of the following 3 steps.

1.) Subtract 1 from it. (n= n - ­1) ,

2.) If its divisible by 2, divide by 2.( if n%2==0, then n= n/2 ) ,

3.) If its divisible by 3, divide by 3. (if n%3 == 0, then n = n / 3 ).

The time complexity of your code should be O(n).

**Input format :**

Line 1 : A single integer i.e. n

**Output format :**

Line 1 : Single integer i.e number of steps

**Constraints :**

1 <= n <= 10^5

**Sample Input 1 :**

4

**Sample Output 1 :**

2

**Sample Output 1 Explanation :**

For n = 4

Step 1 : n = 4/2 = 2

Step 2 : n = 2/2 = 1

**Sample Input 2 :**

7

**Sample Output 2 :**

3

**Sample Output 2 Explanation :**

For n = 7

Step 1 : n = 7 ­ - 1 = 6

Step 2 : n = 6 / 3 = 2

Step 3 : n = 2 / 2 = 1

**Question 2 Minimum Number Of Squares**

**Send Feedback**

A number can always be represented as a sum of squares of other numbers. Note that 1 is a square and we can always break a number as [(1 \* 1) + (1 \* 1) + (1 \* 1) + …]. Given a number n, find the minimum number of squares that sum to n.

**Input format:**

The first and only line of input contains an integer N (1 <= N <= 10000)

**Constraints:**

Time Limit: 1 seconds

**Output format:**

The first and only line of output contains the minimum number if squares that sum to n.

**Sample Test Cases:**

Sample Input 1:

100

Sample Output 1:

1

Explanation:

We can write 100 as 10^2 also, 100 can be written as (5^2) + (5^2) + (5^2) + (5^2), but this representation requires 4 squares. So, in this case, the expected answer would be 1, that is, 10^2.

**Question 3 Longest Increasing Subsequence**

**Send Feedback**

Given an array with N elements, you need to find the length of the longest subsequence of a given sequence such that all elements of the subsequence are sorted in strictly increasing order.

**Input Format**

Line 1 : An integer N

Line 2 : Elements of arrays separated by spaces

**Output Format**

Line 1 : Length of longest subsequence

**Input Constraints**

1 <= n <= 10^3

**Sample Input :**

6

5 4 11 1 16 8

**Sample Output 1 :**

3

**Sample Output Explanation**

Length of longest subsequence is 3 i.e. (5,11,16) or (4,11,16).

**Sample Input 2:**

3

1 2 2

**Sample Output 2 :**

2