Problem Result

Code : Min Steps to 1

Send Feedback

Given a positive integer 'n', find and return the minimum number of steps that 'n' has to take to get reduced 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).

Write brute-force recursive solution for this.

Input format :

The first and the only line of input contains an integer value, 'n'.

Output format :

Print the minimum number of steps.

Constraints :

1 <= n <= 200

Time Limit: 1 sec

Sample Input 1 :

4

Sample Output 1 :

2

Explanation of Sample Output 1 :

For n = 4
Step 1 : n = 4 / 2 = 2
Step 2 : n = 2 / 2 = 1

Sample Input 2 :

7

Sample Output 2 :

3

Explanation of Sample Output 2 :

For n = 7
Step 1 : n = 7 - 1 = 6
Step 2 : n = 6 / 3 = 2
Step 3 : n = 2 / 2 = 1

```cpp
#include<bits/stdc++.h>
int countStepsTo1(int n){
    int *ans=new int[n+1];
    ans[1]=0;
    for(int i=2;i<=n;i++){
        int x,y=INT_MAX,z=INT_MAX;
        x=ans[i-1];
        if(i%2==0)
            y=ans[i/2];
        if(i%3==0)
            z=ans[i/3];
        ans[i]=min(x,min(y,z))+1;
    }
    return ans[n];
}

```