Viral Video Tutorial

Pre-requisites:

This problem will require dynamic programming and concepts of divisibility to solve. Make sure your basics of recursion, dynamic programming and divisibility are clear before attempting this problem. The following links can help you learn the prerequisites:

- https://www.topcoder.com/community/data-science/data-science-tutorials/mathematics-for-topcoders/
- http://mathworld.wolfram.com/Divisible.html
- http://www.geeksforgeeks.org/recursion/
- https://www.topcoder.com/community/data-science/data-science-tutorials/an-introduction-to-recursion-part-1/
- https://www.topcoder.com/community/data-science/data-science-tutorials/an-introduction-to-recursion-part-2/
- https://www.topcoder.com/community/data-science/data-science-tutorials/dynamic-programming-from-novice-to-advanced/

Problem Description:

x1, x2 and x3 are given prime numbers.
Our goal in this problem is to find the kth number in the sequence of all numbers (arranged in ascending order) that can only be divided by 1, x1, x2, and x3.
For example, in case of x1 = 2, x2 = 3, x3 = 5, the sequence will become 2, 3, 4, 5, 6, 8, 9, 10, 12, 15…..

Difficulty Level:

Medium

Editorial:

Let x1=2, x2=3, x3=5

Then, $f(1) = 2 = 2^1 3^0 5^0$

$f(2) = 3 = 2^0 3^1 5^0$

$f(3) = 4 = 2^2 3^0 5^0$

f(4) = 5 = $2^0 3^0 5^1$

f(6) = 6 = $2^1 3^1 5^0$

and so on…

The numbers can be obtained from

f(1) = min(2*1, 3*1, 5*1) = 2

f(2) = min(2*f(1), 3*1, 5*1) = 3

f(3) = min(2*f(1), 3*f(1), 5*1) = 4

f(4) = min(2*f(2), 3*f(1), 5*1) = 5

f(6) = min(2*f(2), 3*f(1), 5*f(1)) = 6

On observing the above relations, it can be inferred that

f(i) = min (x1*f(a), x2*f(b), x3*f(c))

where a is one more than the last index in which the power of x1 was incremented, b is one more than the last index in which the power of x1 was incremented and so on…

Hence, the algo to generate the numbers of the sequence will look something like:

```
1 Declare an array for the numbers:  arr[n]

2 Initialize first no:  arr[0] = min(x1, x2, x3)

3 Initialize three array index variables i1, i2, i3 to point to

  1st element of the array:

      i1 = i2 = i3 =0;

4 Initialize 3 choices for the next no:
        next_mulitple_of_x1 = arr[i1]*x1;

        next_mulitple_of_x2 = arr[i2]*x2;

        next_mulitple_of_x3 = arr[i3]*x3;
```

5 Now go in a loop to fill all numbers till k:

```
For (i = 1; i < k; i++ )

{

        next_no  = Min(next_mulitple_of_x1,

                next_mulitple_of_x2,

                next_mulitple_of_x3);

        arr[i] =  next_no

        if (next_no  == next_mulitple_of_x1)

         {

    i1 = i1 + 1;

    next_mulitple_of_x1 = arr[i1]*x1;

        }

  if (next_no  == next_mulitple_of_x2)

          {

    i2 = i2 + 1;

    next_mulitple_of_x2 = arr[i2]*x2;

        }

  if (next_no  == next_mulitple_of_x3)

          {

    i3 = i3 + 1;

    next_mulitple_of_x3 = arr[i3]*x3;

        }


}/* end of for loop */
```

```
6.return next_no
```

Complexity of solution:

The time complexity of the above solution will be O(k) where we have to find the ‘kth’ number in the sequence of given numbers.