**Problem #1**

**LCM**

In this problem your input is 2 non-negative numbers *a* and *b*. You need to output the LCM (Least Common Multiple) of *a* and *b***.** Recall that LCM and GCD (Greatest Common Divisor) has the following relationship: a \* b = GCD(a, b) \* LCM(a, b)

To complete this task, you must write the following 2 functions (design appropriate parameters and return type):

* GCD – determines the greatest common divisor of 2 non-negative integers. (Use your elementary math’s knowledge on how to implement GCD)
* LCM – determines the least common multiple of 2 non-negative integers.

You ­must not perform any *printf/scanf* in the above functions.

|  |  |
| --- | --- |
| **Sample Input(s)** | **Corresponding Output(s)** |
| 10 35  20 10  27 29  0 5 | 70  20  783  0 |

**Problem #2**

**Prime Factorization**

The prime factorization of a positive integer is a list of the integer's prime factors, together with their multiplicities. The fundamental theorem of arithmetic says that every positive integer has a single unique prime factorization. In this problem, your input will be an integer *n* such than 1 < *n* < 100001. You need to output the prime factorization of the integer. To complete this task, you must write the following 2 functions (design appropriate parameters and return type):

* isPrime – detects whether an integer is prime or not.
* isFactor – detects whether an integer is a factor of another integer.

You ­must not perform any *printf/scanf* in the above functions.

|  |  |
| --- | --- |
| **Sample Input(s)** | **Corresponding Output(s)** |
| 85000  234  7 | 2^3 X 5^4 X 17^1  2^1 X 3^2 X 13^1  7^1 |