ESC101 : Fundamental of Computing
Lab 6 for 12th September 2008

From this lab onwards, you will have to write JAVA programs which receive input from command-line (terminal). This was demonstrated during the lecture of 8th September.

1. **Euler's Totient function :** (marks = 10)
   In number theory, the totient $\varphi(n)$ of a positive integer n is defined to be the number of positive integers less than or equal to n that are coprime to n. For example, $\varphi(9) = 6$ since the six numbers $1, 2, 4, 5, 7$ and 8 are coprime to 9. [1]

   To calculate $\varphi(n)$, we use the following formula:

   $$\varphi(n) = n \prod_{p|n} \left(1 - \frac{1}{p}\right)$$

   where p is a prime.

   Here, $p|n$ denotes p *divides* n. $\prod_{p|n} f(p)$ denotes the product of all function values $f(p)$ where p divides n. e.g.

   $$\varphi(36) = \varphi\left(3^2 \cdot 2^2\right) = 36\left(1 - \frac{1}{3}\right)\left(1 - \frac{1}{2}\right) = 36 \cdot \frac{2}{3} \cdot \frac{1}{2} = 12.$$

   Write a JAVA program that takes n as command-line argument and finds $\varphi(n)$ using the above formula. You should design and use the following method in your program:

   `boolean isPrime (int k)`: Returns true if k is prime, and false otherwise.

   Please observe that according to the definition, $\varphi(n)$ would be an integer. But if we use the formula directly, then $\left(1 - \frac{1}{p}\right)$ wuld be of type `double` or `float`. Multiplying these intermediate values to calculate $\varphi(n)$ would lead to loss of accuracy. Hence you should perform the calculations only in integers, i.e. all intermediate values should be stored in variables of type `int`.

2. **Sum of positive divisors function** (marks=10)
   The *sum of positive divisors function* $\sigma_x(n)$ is defined as the sum of the xth powers of the positive divisors of n, or

   $$\sigma_x(n) = \sum_{d|n} d^x$$

   For example, $\sigma_0(12)$ is the number of the divisors of 12:

   $$\begin{aligned} \sigma_0(12) &= 1^0 + 2^0 + 3^0 + 4^0 + 6^0 + 12^0 \\ &= 1 + 1 + 1 + 1 + 1 + 1 = 6 \end{aligned}$$

   while $\sigma_1(12)$ is the sum of all the divisors:

   $$\begin{aligned} \sigma_1(12) &= 1^1 + 2^1 + 3^1 + 4^1 + 6^1 + 12^1 \\ &= 1 + 2 + 3 + 4 + 6 + 12 = 28. \end{aligned}$$

[1] For more about *Euler's Totient function*, visit `http://en.wikipedia.org/wiki/Euler's_totient_function`.

Write a JAVA program to find $\sigma_x(n)$, where x and n are positive integers passed as first and second command-line arguments respectively. You should design and use the following method in your program:

`int power (int k, int m)`: Returns k^m.