ESC101 : Fundamental of Computing
Mock Lab Test for 17th September 2008

Instructions:

1. The duration of the test is 3 hrs (from **2:00 pm to 5:00 pm**).
2. **Directory Structure:** Create a directory and name it with your roll number. For example, if your roll number is $Y8001$, the directory should be named $Y8001$ (`Y` should be upper case). Create two files inside this directory: *DecToBinConverter.java* and *CircularPrime.java*.
3. Please use *meaningful* identifiers for variables and methods. Use comments to improve readability of the program. Properly indent your code. Otherwise some marks may get deducted irrespective of whether your program is correct.

Problems:

1. **Decimal to Binary Conversion :**
   As all of us know, in *decimal number* system numeric values are represented by the ten symbols $0, 1, \ldots, 9$. Therefore, each positive integer in decimal number system is actually the representation of the weighted summation of the powers of 10 (ten) . For example, 7409_{10}(Seven thousand four hundred and nine) is equal to $7\times10^3+4\times10^2+0\times10^1+9\times10^0$. Similarly in *binary number* system all the numbers are represented by using only two symbols 0 and 1, and hence all the numbers are representation of the weighted summation of the powers of 2_{10} (two). For example, decimal 13_{10}(thirteen) can be represented in binary number system as 1101_2 because $13_{10} = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$.

   Write a JAVA program to convert a number given in decimal number representation to its equivalent binary number representation and print the binary value. For example, if the input is 25, then the output should be 11001 (which is the binary representation of 25_{10}). Similarly, if the input is -11 then the output should be -1011 (which is the binary representation of -11_{10}). Your program has to work for all values of integer n whose binary representation has 10 digits only. You must take the input as command-line argument.

2. **Circular prime number :**
   A number is *circular prime* if it is prime and all its cyclic rotations are also prime. e.g. The number **1**193 is a circular prime number because it is prime and all its cyclic rotations 193**1**, 93**1**1, 3**1**19 are prime.

   Write a JAVA program that takes an integer n as input from the command line and prints whether it is circular prime or not. Your program has to work for all values of n which can be stored in data type `int`.

   Note: When constructing the next number to be checked, you may want to implement a `circular left rotation` method, e.g. `abcde -> bcdea` or a `circular right rotation` method e.g. `abcde -> eabcd`. You are allowed to implement any rotation.