

ESC101 : Fundamental of Computing
Mock Lab Test for 16th 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. e.g. If your roll number is Y8001, the directory should be named Y8001 (Y should be upper case). Create two files inside this directory: *BinToDecConverter.java* and *TrainJourney.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. **Binary To Decimal Conversion :** (marks=10)

As all of us know that, in **decimal number** system numeric values are represented by the ten symbols 0, 1, . . . , 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 $7 \times 10^3 + 4 \times 10^2 + 0 \times 10^1 + 9 \times 10^0$. 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 binary number representation to its equivalent decimal number representation. For example, if the input is 11001, then the output should be 25. Similarly, if the input is -1011 then the output should be -11.

Note: Your program has to take input from command line and by default the input from command line is considered as decimal integer (not binary). The input could be a negative number also, but the maximum number of digits will not be more than 9. You may also assume that user will enter valid input (strings of 1's and 0's only).

2. **Minimum Travel Time of a Sub-Way Train :** (marks=10)

Subway trains can move people quickly from one station to the next. The distance between two consecutive stations is **length** in meters. For safety, the train can't move faster than **maxVelocity** m/s (meters per second). For comfort, the value of acceleration or retardation can't be larger than **maxAcceleration** m/s^2 . The train starts with velocity 0 m/s , and it must stop at the next station (i.e., arrive there with a velocity of 0 m/s).

Write a JAVA program to find the time taken by a train to travel from one station to another if the **distance** between the stations, **maxVelocity** and **maxAcceleration** are given.

Note: Your Program should take the inputs - **distance**, **maxVelocity** and **maxAcceleration** - as integers from the command line in this order. You may assume that the user will provide non-negative values as input. You may use the following method provided by JAVA :

`double Math.sqrt(double)`

This method computes square root of a given number. For example, `Math.sqrt(100)` will return 10.0