SGSITS
Dept. of Computer Engg.
CO7101: Algorithmics
Lab Assignment 1

Tuesday 18th July, 2017

Note: Copying of assignment is not allowed. If any student is found to have copied any solution from a fellow student or internet, s/he will be awarded **zero** marks for the whole assignment.

Date of Submission: Tuesday 25^{th} July, 2017

1. Consider following series:
   $sin(x) = x - x^3/3! + x^5/5! - ..... + x^{199}/199!$

   (a) Write a program that computes above series. You can you as many loops, user defined function and library function as you like.

   (b) Calculate how many multiplications, divisions, additions and subtractions your program performs.

   (c) **Challenge:** Can you find a method to reduce number of multiplication, division, addition and subtraction operations performed by your program?
   OR

   (d) **Challenge:** Can you compute the same series with one loop only and without using math library, without any userdefined function (like int fact(int))?

2. **Amicable numbers**:
   WAP that takes two numbers and finds out if they are amicable numbers or not. Amicable numbers are two different numbers so related that the sum of the proper divisors of each is equal to the other number.
   For example, the smallest pair of amicable numbers is (220, 284). They are amicable because the proper divisors of 220 are 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 and 110, of which the sum is 284; and the proper divisors of 284 are 1, 2, 4, 71 and 142, of which the sum is 220.

3. **Permutations:**
   WAP that generates all permutations of a number. For example, all permutations of number 312 are 123,132,213,231,312 and 321.

4. **Combinations:**
   Write a function **void printCombinations(int *array, int n, int m)**, where *array* is an integer array of size n and m is size of combination. The function prints all ${}^{n}C_{m}$ combinations of size m from the array.

   Note: This problem can be solved either using recursion or using the solution of generating permutations.

5. Write a function **int ** rotateMatrix(int **inputMatrix, int m , int n, int k)**, that takes an input matrix named $inputMatrix$ of size m by n (where m and n both are positive even numbers) and returns a matrix with each element rotated k times.