Assignment :02 Section A Submission Date: 03rd Feb 2019

Instructions: (Read Carefully)
a. For each problem, you are supposed to do the best, average and worst-case analysis.
b. The proposed algorithm must be clearly explained and neatly written.
c. Try to show and compare the experimental results with the help of tables and suitable graphs.
d. The Serial number of a question also shows the group Id to which assignment has been allotted.
e. Do not use multiplication, division or modulus operator. {This is mandatory}

Q1. Multiply two 3 digit positive integers.
Q2. Divide two positive integers of which one must be of 4 digits and other of 2 digits.
Q3. If length of the data type is limited (number of bits is limited to store a number) then how to multiply two large numbers (particularly for the case of overflow).
Q4. Write an algorithm to find the square root of the given number.
Q5. For a given floating point number check whether the integer part is a prime number.
Q6. For a given floating point number check whether the integer part is a Pythagorean hypotenuse.
Q7. For a given floating point number check whether the integer part is a Fibonacci number.
Q8. For a given floating point number check whether the integer part is a perfect square.
Q9. For a given floating point number check whether the number formed using Least Significant Digit of integer part and Most Significant Digit of fraction part is a prime number.
Q10. Consider Q9 for Fibonacci Numbers.
Q11. Consider Q9 for Pythagorean hypotenuse.
Q12. Consider Q9 for perfect square.
Q13. If n digit number is given, design an algorithm to find mid digits and check whether the number formed by mid-digit(s) are prime numbers or not? (If n is odd, there will be single middle digit while if n is even, take pair of digits as mid digits.)
Q14. Divide two floating point numbers (may be negative) maintaining precision up to 4 decimal places.
Q15. Multiply two floating point numbers (may be negative).
Q16. Divide two integers (may be negative) maintaining precision up to 4 decimal places.
Q17. Locate decimal point in a given floating-point number.
Q18. Remove decimal point from a given floating point number and merge integer and fraction part to form an integer. Check if it is a prime number.
Q19. For any integer x (may be negative) find out x^n (n is positive).
Q20. Implement a function for rounding off a given floating point number up to 2 decimal places.
Q21. Multiply two unsigned floating point numbers.
Q22. Implement 1-D look-up table in order to get single digit multiplication. Using it multiply two positive integers.
Q23. Consider Q22 and multiply the respective integer parts of two positive floating point numbers.
Q24. Consider Q22 and multiply the respective fraction parts of two positive floating point numbers.
Q25. Given a number x find $1/x$.