Assignment :01 Section A: Submission Date: 27th January 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.***

1) Design an algorithm to check whether a given number is prime or not?
2) Design an algorithm to check whether a given number is a Fibonacci number or not?
3) If given number is not a prime number, design an algorithm to locate the nearest prime number.
4) If given number is not a Fibonacci number, design an algorithm to find the nearest Fibonacci number.
5) Design an algorithm to check that a given number is the exact square number or not? (Without using sqrt() function.
6) Design an algorithm to check that a given number is the exact cube number or not? (Without using any library function).
7) Design an algorithm to check that a given number is the exact k^{th} root or not? (Without using any library function).
8) Design an algorithm to check that a given number is the Pythagorean hypotenuse. If it satisfies the condition, then find out the other sides.
9) Decompose a given number N as the sum of largest prime numbers (say A+ B). Where A is a prime number and if B is also a prime number stop the process otherwise repeat the same process for B.
10) Decompose a given number N as the sum of largest Fibonacci numbers (say A+ B). Where A is a Fibonacci number and if B is also a Fibonacci number stop the process otherwise repeat the same process for B.
11) Decompose a given number N as the sum of largest hypotenuse number (say A+ B). Where A is a Pythagorean hypotenuse number and if B is also a hypotenuse number stop the process otherwise repeat the process.
12) Decompose a given number N as the sum of largest exact square numbers (say A+ B). Where A is an exact square number and if B is also an exact square number stop the process otherwise repeat the process.
13) For all 2 or 3 digits positive integers N (9 < N < 1000), check if N is of the form:

$$\mathrm{d}_1^2 + \mathrm{d}_2^2 + \mathrm{d}_3^2 = \mathrm{N}$$

Where, d_i ∀ i ϵ {1,2,3} are the digits of the number N.

14) For all 2 or 3 digits positive integers N (9< N < 1000), check if Armstrong number N is of the form:

$$\mathrm{d}_1 \ + \mathrm{d}_2 \ + \mathrm{d}_3 \ = \mathrm{N}$$

Where d_i ∀ i ϵ {1,2,3} are the digits of the number N.

15) Design an algorithm for all 4 – 7 digit positive integers N to check whether N can be partitioned into two subsets (may be unequal length) such that sum of subsets is same.

16) Let N be an integer number. Check if mid two digits and extreme two digits (for both ends where applicable) make a prime number or not. For example if N = 214678 = 21 46 78 then sum(21) = **3**, sum(46) = sum(10) = **1**, sum(78) = sum(15) = **6**, and further check whether any permutations of 1, 3, and 6 make a prime number or not.

17) Let N be an integer number. Check if mid two digits and extreme two digits (for both ends where applicable) make a Fibonacci number or not. And further, check whether any permutation of these partitions makes a Fibonacci number or not.

18) Let N be an integer number. Check if mid two digits and extreme two digits (for both ends where applicable) make Fibonacci perfect square numbers. And further, check whether any permutation of these partitions makes a Fibonacci perfect square number or not.

19) Let N be an integer number. Check if mid two digits and extreme two digits (for both ends where applicable) make Fibonacci Pythagorean numbers. And further, check whether any permutation of these partitions makes a Fibonacci Pythagorean number or not.

20) Take a 5 digit number, make 3 partitions and check whether it is a Pythagorean triplet. If not, define nearly Pythagorean. Check how nearly it is right-angled. Check for all possible partitions.

21) Given an array of randomly generated positive integers of size 1000 such that, $10^4 < A[i] < 10^6$ for any i, where 1 <= i <= 1000. Locate the smallest Prime number.

22) Given an array of randomly generated positive integers of size 1000 such that, $10^4 < A[i] < 10^6$ for any i, where 1 <= i <= 1000. Locate the smallest Fibonacci number.

23) Given an array of randomly generated positive integers of size 1000 such that, 104< A[i] < 106 for any i, where 1 <= i <= 1000. Locate the smallest perfect square.