**ROLL NUMBER – 119CS0102**

**NAME- SUSHREE SATARUPA**

**DAA Assignment 1**

***Assignment Write algorithms in Pseudo code.***

1. ***Present an algorithm that searches an unsorted array a[l: n] for the element x. If x occurs, then return a position in the array; else return zero.***
2. **Algorithm** Searching (a, n, x)
3. // searching for x in a[1:n]
4. {
5. **if** (n < 1) **then**
6. **{**
7. **write** (“Enter positive value of n”);
8. **return**;
9. }
10. **for** i :=1 **to** n **do**
11. {
12. **if (** a[i] = x ) **then** **return** i;
13. }
14. **return** 0;
15. }
16. ***The factorial function n! has value 1, when n ≤ 1and value n \* (n-1)!, when n > 1. Write both a recursive and an iterative algorithm to compute n!***

**A)**

1. **Algorithm** factorial\_iterative (n)
2. //finding factorial using iteration
3. {
4. **if** (n <= 1) **then** **return** 1;
5. F := 1;
6. **for** i := 2 **to** n **do**
7. {
8. F := F\* i;
9. }
10. **return** F;
11. }

**B)**

1. **Algorithm** factorial\_recursive(n)
2. //find factorial using recursion
3. {
4. **if** n <= 1 **then** **return** 1;
5. **return** n\* factorial\_recursive(n-1);
6. }
7. ***The Fibonacci numbers are defined f0 =0, f1 = 1, and fj = fj-1 + fj-2 for j > 1.Write both a recursive and an iterative algorithm*** ***to compute nth Fibonacci number.***

**A)**

1. **Algorithm** Fibonacci\_iterative(n)
2. // to compute nth Fibonacci number
3. {
4. **if** n == 0 **then** return 0;
5. **if** n == 1 **then** return 1;
6. **if** n < 0 **then**
7. {
8. **write** (“Enter positive value of n”);
9. **return**;
10. }
11. a := 0;
12. b := 1;
13. **for** i := 2 **to** n **do**
14. {
15. c := a + b;
16. a := b;
17. b := c;
18. }
19. **return** c;
20. }

**B)**

1. **Algorithm** Fibonacci\_recursive(n)
2. // to compute nth Fibonacci number
3. {
4. **if** n == 0 **then** **return** 0;
5. **if** n == 1 **then** **return** 1;
6. **if** n < 0 **then**
7. **{**
8. **write** (“Enter positive value of n”);
9. **return**;
10. }
11. **return** Fibonacci\_recursive(n-1) + Fibonacci\_recursive(n-2);
12. }
13. ***Give an algorithm to solve the following problem: Given n, a positive integer,*** ***determine whether n is the sum of all of its divisors, that is, whether n is the sum of all t such that 1≤ t < n, and t divides n.***
14. **Algorithm** Sum\_of\_div(n)
15. // to determine whether n is the sum of all of its divisors
16. {
17. **if** n > 0 **then**
18. {
19. Sum := 0;
20. **for** i := 1 **to** n^(1/2) **do**
21. {
22. **if** n mod i == 0 and i == n/i **then** Sum := Sum+ i **else** Sum := Sum+ i + n/i ;
23. }
24. **if** Sum == n **then** **write** (“number is sum of all its divisors”) **else** write (“no, number is not the sum of all its divisors”);
25. }
26. **else**
27. {
28. **write** (“Enter positive integer only”);
29. }
30. }