3. Sum of even no. using while loop

Sample-1:

enter no. of ~~given~~ even no 4

o/p ~~given~~ even Sum of ~~given~~ no is 6

Sample-2:

enter no. of ~~gi~~ even no: 7

Sum of even no: 12

1. Find the given no is even or odd.

Sample-1:

~~ent~~ Input: 100

Output: given no is even

2. Sum of n natural no

Input: 5

Output: sum of natural no. is 15

4. Reverse a no.

Input: 12345

Output: 54321

5) check whether palindrome or not

1: Input: 111
Output: The number is palindrome

2: Input: 123
Output: The no. is not palindrome.

6. Armstrong

Input: 371
Output: It is a armstrong no.

7. factorial

Input: 3
Output: 6

8. Factorial [recursion]

Input: 3
Output: 6

9. Fibbonacci series

Input: 5
O/P: 0, 1, 1, 2, 3, 5

Day-3

1) Singly linked list:

O/P: Insert: 24, 78, 92, 40

delete: 24, 78

after deletion: 92, 40

o/p

2) Stack: push, pop, display:

push: 1

push: 2

push: 3

push: 4

pop: 6

Display stack: 5
2

o/p

3) Queue data structure:

enqueue: 1, 2, 3, 4, 5

Queue is full

Dequeue: 1

elements in queue are,
2, 3, 4, 5

o/p

4) Conversion of Infix to postfix

Infix: "A*(B+C)/D"

postfix: ABC+D/

o/p

5) Evaluate elements using stack.

Enter arithmetic expression: 45+

Result: 9

o/p

Day-4

1. Binary tree traversal [C program]
2. C program to impliment AVL tree with all rotations
3. C program to implement hashing using linear proving
4. C program to implement Bubble sort
5. " " Selection "
6. " " insertion "
7. " " Quick "
8. " " Merge "

Outputs

1) Tree traversal
Elements: 25, 82, 56, 41, 14, 19

Pre-order: 25, 14, 13, 19, 82, 56, 41
Inorder: 13, 14, 19, 25, 41, 56, 82
Post order: 13, 19, 14, 41, 56, 82, 25

2) AVL tree: 4, 2, 1, 3, 7, 5, 8
After deletion: 4, 2, 1, 7, 5, 8

3) Hashing using linear proving
1. Insert
Enter a value: 3, 6, 8, 9
2. Display
index 0 value = 0
index 1 value = 0
index 2 value = 2
index 3 value = 3
index 4 value = 0
index 5 value = 0
index 6 value = 6
index 7 value = 0
index 8 value = 0
index 9 value = 9

i) Bubble sort
No of elements: 2
Enter elements: 3, 2
Bubble sorting: 2, 3

ii) Selection sort
Enter no. of elements: 3
Enter the elements: 3, 6, 2
Sorted: 2, 3, 6

iii) Insertion sort
Enter elements: 9, 2, 11, 15, 12
Sorted: 2, 9, 11, 12, 15

iv) Quick sort
No. of elements: 4
Enter elements: 3, 4, 2, 6
Order of elements: 2, 3, 4, 6

v) Merge sort
12, 11, 13, 5, 6, 7, 8, 9
Sorted array: 5, 6, 7, 8, 9, 11, 12, 13

Day-5

~~1. Shortest path prims~~

2. Minimum spanning tree

Edge	weight
0-1	3
1-2	4
2-3	5
1-4	7