Important Practical Programs for CP

1. Write a program to read Title, Author and Price of 5 books using array of structures. Display the records in ascending order of Price.
2. Implement a program to perform addition of two matrices.
3. Write a program to check whether a word is palindrome or not.
4. Implement a program to find transpose of a matrix.
5. Write a program to print Fibonacci series.
6. Write a C program to perform multiplication of two matrices.
7. Write a program to find the power of x raised to n that is: x^n, using recursive function.
8. Write a program to print the following pattern.

```
A
B B
C C C
D D D D
```

9. Write a program to find largest element of an 1D array.
10. Write a Program to calculate and display sum of all the elements of the matrix.
11. Define a structure called player with data members as player name, team name, batting average. Store and display the information of at least 10 players.
12. Write a program to display the following for the user specified number of lines.

```
*
**
***
****
*****
******
```

13. Write a program in C to find the reverse of a given string without using inbuilt string function.
14. Write a program to store and display at least 10 records of the name, roll number and fees of a student using structure.
15. Explain String function for the following operations with example.
    a. Copy string from source to destination.
    b. Merging of two strings.
16. Write a program to print the following pattern. (Note- Not only 4 lines, it should print N lines taken from the user.)

```
A
B   B
```

C C C
D D D D

17. Write a C-program to create array of structures in order to store details of almost 100 books. The book details are book name, book price, book page number and book author name.
18. Write a program that will accept two-dimensional square matrix and find the sum of diagonal elements.
19. Write a C program to accept 10 integers from the user and arrange them in ascending order and display them.
20. Write a C program to find GCD of two numbers using recursion.