**Assignment C Programming Language**

**Set 1**

**Problem 1: Armstrong Numbers**

Write a C program to find all Armstrong numbers between 1 and n (inclusive), where n is a positive integer input by the user.

An Armstrong number is a number that is equal to the sum of its own digits raised to the power of the number of digits. For example, 153 is an Armstrong number because 1^3 + 5^3 + 3^3 = 153.

Problem 2: Caesar Cipher

Write a C program that accepts a string and a key as input and encrypts the string using a Caesar cipher.

A Caesar cipher is a simple encryption technique that shifts each letter of the string by the key number of positions. For example, if the key is 3, then 'a' becomes 'd', 'b' becomes 'e', 'c' becomes 'f', and so on.

**Problem 2: Array Operations**

Write a C program to perform the following operations on an array of integers:

a) Display the array

b) Find the sum of all elements in the array

c) Find the average of all elements in the array

d) Find the maximum element in the array

e) Find the minimum element in the array

**Problem 3: Pointer Operations**

Write a C program to perform the following operations on a pointer to an integer:

a) Initialize the pointer to a specific integer value

b) Print the value of the integer pointed to by the pointer

c) Increment the pointer and print the new value of the integer pointed to by the pointer

d) Decrement the pointer and print the new value of the integer pointed to by the pointer

**Problem 4: Input/Output Operations**

Write a C program to read and write data from a file. The program should perform the following operations:

a) Open a file for writing

b) Write a string to the file

c) Close the file

d) Open the same file for reading

e) Read the string from the file

f) Print the string to the console

**Problem 5: Sorting Algorithm**

Write a C program to implement the sort algorithm to sort an array of integers in ascending order.

The program should perform the following operations:

* Read an array of n integers from the user
* Sort the array using the insertion sort algorithm
* Display the sorted array

1. Write a select sort algorithm
2. Write a insertion sort algorithm
3. Write a bubble sort algorithm
4. Write a quick sort algorithm

**Problem 6: Matrix Transpose**

Write a C program to transpose a matrix of size m x n, where m and n are positive integers input by the user.

The transpose of a matrix A is a matrix B such that B(i,j) = A(j,i) for all i and j.

Example Input/Output:

Enter value of m: 3 Enter value of n: 2

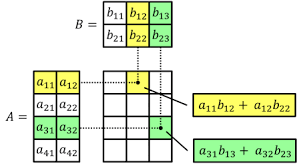
Enter elements of matrix A: 1 2 3 4 5 6

Transpose of matrix A: 1 3 5 2 4 6

Note: You may assume that the input matrix has at most 10 rows and 10 columns.

**Problem 7: Matrix Multiplication**

Write a C program to multiply two matrices of size m × n and n × p, where m, n, and p are positive integers input by the user.

The product of two matrices A and B is a matrix C such that C(i,j) = Σ A(i,k) \* B(k,j) for k = 1 to n.

Example Input/Output:

Enter value of m: 2 Enter value of n: 3 Enter value of p: 2

Enter elements of matrix A: 1 2 3 4 5 6

Enter elements of matrix B: 7 8 9 10 11 12

Resultant matrix C: 58 64 139 154

**Problem 8: The shape**

a) Write a C program to generate a multiplication table in the shape of a square. The program should perform the following operations:

* Read a positive integer n from the user
* Generate the multiplication table for n from 1 to n
* Display the multiplication table in the following format:

Multiplication table for [n]:

[1 x 1] [1 x 2] ... [1 x n]

[2 x 1] [2 x 2] ... [2 x n] ...

... ...

[n x 1] [n x 2] ... [n x n]

Example Input/Output:

Enter a positive integer: 5

Multiplication table for 5:

[1 x 1] [1 x 2] [1 x 3] [1 x 4] [1 x 5]

[2 x 1] [2 x 2] [2 x 3] [2 x 4] [2 x 5]

[3 x 1] [3 x 2] [3 x 3] [3 x 4] [3 x 5]

[4 x 1] [4 x 2] [4 x 3] [4 x 4] [4 x 5]

[5 x 1] [5 x 2] [5 x 3] [5 x 4] [5 x 5]

Note: You may assume that the input number is at most 10.

b) Write a C program to generate a triangle shape using asterisks (\*). The program should perform the following operations:

* Read a positive integer n from the user
* Generate the triangle shape using asterisks (\*) as shown below:

\* \* \* \*

\*\* \*\* \*\*\* \*\*\*

\*\*\* \*\*\* \*\*\*\*\* \*\*\*\*\*

\*\*\*\* \*\*\*\* \*\*\*\*\*\*\* \*\*\*\*\*\*\*

\*\*\*\*\* \*\*\*\*\* \*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*

(1) (2) (3) (4)

Note: You may assume that the input number is at most 20.

c) Write a C program to generate a triangle shape

1

212

32123

4321234

543212345

……………….

109876543212345678901

Note: You may assume that the input number is at most 20.

\*

\*\*\*

\*\*\*\*\*

\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*