1. C# program to read data from file character by character till the end of the file.

using System;

using System.IO;

class Program

{

static void Main()

{

string filePath = "sample.txt"; // Replace with your file path

try

{

using (StreamReader reader = new StreamReader(filePath))

{

int character;

while ((character = reader.Read()) != -1) // Read() returns -1 at end of file

{

Console.Write((char)character); // Convert int to char and print

}

}

}

catch (FileNotFoundException)

{

Console.WriteLine("File not found: " + filePath);

}

catch (Exception ex)

{

Console.WriteLine("An error occurred: " + ex.Message);

}

}

}

Write a C# program to find the factorial of a given number

using System;

class FactorialProgram

{

static void Main()

{

Console.Write("Enter a non-negative integer: ");

string input = Console.ReadLine();

if (int.TryParse(input, out int number) && number >= 0)

{

long factorial = 1;

for (int i = 1; i <= number; i++)

{

factorial \*= i;

}

Console.WriteLine($"Factorial of {number} is: {factorial}");

}

else

{

Console.WriteLine("Invalid input. Please enter a non-negative integer.");

}

}

}

3. Write a C# program to check whether a given number is prime or not.

using System;

class PrimeCheck

{

static void Main()

{

Console.Write("Enter a number: ");

string input = Console.ReadLine();

if (int.TryParse(input, out int number) && number > 1)

{

bool isPrime = true;

for (int i = 2; i <= Math.Sqrt(number); i++)

{

if (number % i == 0)

{

isPrime = false;

break;

}

}

if (isPrime)

Console.WriteLine($"{number} is a prime number.");

else

Console.WriteLine($"{number} is not a prime number.");

}

else

{

Console.WriteLine("Please enter an integer greater than 1.");

}

}

}

Write a C# program to check whether the given string is a palindrome or not

using System;

class PalindromeCheck

{

static void Main()

{

Console.Write("Enter a string: ");

string input = Console.ReadLine();

// Remove spaces and convert to lowercase for a case-insensitive comparison

string cleanedInput = input.Replace(" ", "").ToLower();

// Reverse the string

char[] charArray = cleanedInput.ToCharArray();

Array.Reverse(charArray);

string reversed = new string(charArray);

// Compare original and reversed strings

if (cleanedInput == reversed)

Console.WriteLine($"\"{input}\" is a palindrome.");

else

Console.WriteLine($"\"{input}\" is not a palindrome.");

}

}

Write a C# program to find minimum & maximum from array?

using System;

class MinMaxFromArray

{

static void Main()

{

// Sample array

int[] numbers = { 23, 5, 67, 89, 12, 4, 98, 1 };

// Initialize min and max with the first element

int min = numbers[0];

int max = numbers[0];

// Loop through the array to find min and max

foreach (int num in numbers)

{

if (num < min)

min = num;

if (num > max)

max = num;

}

Console.WriteLine("Array elements: " + string.Join(", ", numbers));

Console.WriteLine("Minimum value: " + min);

Console.WriteLine("Maximum value: " + max);

}

}

Write a C# program to create an MXN matrix and perform the following operation.

a. Addition

b. Multiplication

c. Transpose

using System;

class MatrixOperations

{

static void Main()

{

Console.Write("Enter number of rows (M): ");

int m = int.Parse(Console.ReadLine());

Console.Write("Enter number of columns (N): ");

int n = int.Parse(Console.ReadLine());

// Declare matrices

int[,] matrixA = new int[m, n];

int[,] matrixB = new int[m, n];

int[,] result;

// Input for Matrix A

Console.WriteLine("\nEnter elements of Matrix A:");

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

{

Console.Write($"A[{i},{j}]: ");

matrixA[i, j] = int.Parse(Console.ReadLine());

}

// Input for Matrix B

Console.WriteLine("\nEnter elements of Matrix B (for addition and multiplication):");

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

{

Console.Write($"B[{i},{j}]: ");

matrixB[i, j] = int.Parse(Console.ReadLine());

}

// a. Matrix Addition

result = new int[m, n];

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

result[i, j] = matrixA[i, j] + matrixB[i, j];

Console.WriteLine("\nResult of A + B:");

PrintMatrix(result);

// b. Matrix Multiplication (only valid if n == m)

if (n == m)

{

result = new int[m, n];

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

for (int k = 0; k < n; k++)

result[i, j] += matrixA[i, k] \* matrixB[k, j];

Console.WriteLine("\nResult of A \* B:");

PrintMatrix(result);

}

else

{

Console.WriteLine("\nMatrix multiplication is not possible (Matrix A columns != Matrix B rows).");

}

// c. Transpose of Matrix A

int[,] transpose = new int[n, m];

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

transpose[j, i] = matrixA[i, j];

Console.WriteLine("\nTranspose of Matrix A:");

PrintMatrix(transpose);

}

// Helper method to print matrix

static void PrintMatrix(int[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

Console.Write(matrix[i, j] + "\t");

Console.WriteLine();

}

}

}

Write a C# program to accept one string &character, find the occurrence of char from string using function

using System;

class CharacterOccurrence

{

static void Main()

{

Console.Write("Enter a string: ");

string input = Console.ReadLine();

Console.Write("Enter a character to search for: ");

char ch;

// Ensure exactly one character is entered

while (!char.TryParse(Console.ReadLine(), out ch))

{

Console.Write("Invalid input. Please enter a single character: ");

}

int count = CountCharacterOccurrence(input, ch);

Console.WriteLine($"\nThe character '{ch}' occurs {count} time(s) in the string \"{input}\".");

}

// Function to count occurrences of a character in a string

static int CountCharacterOccurrence(string str, char character)

{

int count = 0;

foreach (char c in str)

{

if (c == character)

count++;

}

return count;

}

}