using System;

using System.Text;

namespace Pd

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.UTF8;

Console.WriteLine("Ievadiet savu vārdu:");

string vards = Console.ReadLine();

Console.WriteLine("Ievadiet savu uzvārdu:");

string uzvards = Console.ReadLine();

int[,] originalData = GenerateData(vards, uzvards);

// Pirmā tabula - ar Max, Min, Sum

DisplayTable(vards, originalData, true);

// Otrā tabula - ar Max, Min, Sum pēc transponēšanas

Console.WriteLine("\nPēc kolonnu un rindu apmaiņas:");

int[,] swappedData = TransposeData(originalData);

DisplayTable(vards, swappedData, true);

Console.ReadKey();

}

static int[,] GenerateData(string name, string surname)

{

Random rand = new Random();

int rows = name.Length;

int cols = surname.Length;

int min = name.Length;

int max = name.Length + surname.Length;

int[,] data = new int[rows, cols];

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

{

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

{

data[i, j] = rand.Next(min, max + 1);

}

}

return data;

}

static void DrawLine(int dataColumns, bool withCalculations)

{

int totalCols = dataColumns + (withCalculations ? 3 : 0);

Console.Write("\*");

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

{

Console.Write("\_\_\_\_\_\*");

}

Console.WriteLine();

}

static void DisplayTable(string name, int[,] data, bool showCalculations)

{

int rows = data.GetLength(0);

int cols = data.GetLength(1);

// Galvene

DrawLine(cols, showCalculations);

Console.Write("|");

foreach (char c in name)

{

Console.Write($" {c} |");

}

if (showCalculations)

{

Console.Write(" Max | Min | Sum |");

}

Console.WriteLine();

DrawLine(cols, showCalculations);

// Dati

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

{

int rowMax = int.MinValue;

int rowMin = int.MaxValue;

int rowSum = 0;

Console.Write("|");

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

{

Console.Write($" {data[i, j].ToString().PadLeft(3)} |");

rowMax = Math.Max(rowMax, data[i, j]);

rowMin = Math.Min(rowMin, data[i, j]);

rowSum += data[i, j];

}

if (showCalculations)

{

Console.Write($" {rowMax.ToString().PadLeft(3)} |");

Console.Write($" {rowMin.ToString().PadLeft(3)} |");

Console.Write($" {rowSum.ToString().PadLeft(3)} |");

}

Console.WriteLine();

DrawLine(cols, showCalculations);

}

}

static int[,] TransposeData(int[,] data)

{

int rows = data.GetLength(0);

int cols = data.GetLength(1);

int[,] transposed = new int[cols, rows];

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

{

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

{

transposed[j, i] = data[i, j];

}

}

return transposed;

}

}

}