using System;

using System.Linq;

class Program

{

static void Main()

{

Console.Write("Ievadi vārdu: ");

string vards = Console.ReadLine();

Console.Write("Ievadi uzvārdu: ");

string uzvards = Console.ReadLine();

string FormatWord(string text)

{

char[] result = new char[text.Length];

for (int i = 0; i < text.Length; i++)

result[i] = i % 2 == 0 ? Char.ToUpper(text[i]) : Char.ToLower(text[i]);

return new string(result);

}

string vardsFormatted = FormatWord(vards);

string uzvardsFormatted = FormatWord(uzvards);

string uzvardsReversedFormatted = FormatWord(new string(uzvards.Reverse().ToArray()));

int rows = vards.Length + 3;

int cols = uzvards.Length + 4;

string[,] tabula = new string[rows + 1, cols];

tabula[0, 0] = "X";

for (int i = 0; i < uzvardsFormatted.Length; i++)

tabula[0, i + 1] = uzvardsFormatted[i].ToString();

tabula[0, cols - 3] = "sum";

tabula[0, cols - 2] = "min";

tabula[0, cols - 1] = "max";

Random rand = new Random();

for (int r = 1; r <= vards.Length; r++)

{

tabula[r, 0] = uzvardsReversedFormatted[r - 1].ToString();

int sum = 0, min = int.MaxValue, max = int.MinValue;

for (int c = 1; c <= uzvards.Length; c++)

{

int val = rand.Next(vards.Length, vards.Length + uzvards.Length + 1);

tabula[r, c] = val.ToString();

sum += val;

if (val < min) min = val;

if (val > max) max = val;

}

tabula[r, cols - 3] = sum.ToString();

tabula[r, cols - 2] = min.ToString();

tabula[r, cols - 1] = max.ToString();

}

tabula[rows, 0] = "\*";

for (int c = 1; c < cols; c++)

{

if (c >= cols - 3)

{

int[] colVals = new int[vards.Length];

for (int r = 1; r <= vards.Length; r++)

colVals[r - 1] = int.Parse(tabula[r, c]);

if (tabula[0, c] == "sum")

tabula[rows, c] = colVals.Sum().ToString();

else if (tabula[0, c] == "min")

tabula[rows, c] = colVals.Min().ToString();

else if (tabula[0, c] == "max")

tabula[rows, c] = colVals.Max().ToString();

}

else

{

tabula[rows, c] = "\*";

}

}

void PrintLine(char left, char mid, char right)

{

Console.Write(left);

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

{

Console.Write(new string(mid, 6));

Console.Write(i == cols - 1 ? right : '+');

}

Console.WriteLine();

}

void PrintTable(string[,] t, int rCount, int cCount, bool isCalcRow)

{

PrintLine('+', '-', '+');

for (int r = 0; r < rCount; r++)

{

Console.Write('|');

for (int c = 0; c < cCount; c++)

{

string val = t[r, c] ?? "";

Console.Write(isCalcRow && r == rCount - 1 && c >= cCount - 3 ? val.PadLeft(6) : val.PadRight(6));

Console.Write('|');

}

Console.WriteLine();

if (r == 0 || r == rCount - 2)

PrintLine('+', '-', '+');

else

PrintLine('|', '\*', '|');

}

}

PrintTable(tabula, rows + 1, cols, true);

string[,] transposed = new string[cols, rows + 1];

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

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

transposed[c, r] = tabula[r, c];

Console.WriteLine("\nTransponētā tabula ar vārdu un uzvārdu apmainītiem:");

string[,] swappedVU = new string[cols, rows + 1];

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

{

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

{

if (r == 0 && c == 0)

swappedVU[r, c] = "X";

else if (r == 0 && c > 0 && c <= vards.Length)

swappedVU[r, c] = vardsFormatted[c - 1].ToString();

else if (c == 0 && r > 0 && r <= uzvards.Length)

swappedVU[r, c] = new string(vards.Reverse().ToArray())[r - 1].ToString();

else if (r > 0 && c > 0 && r <= uzvards.Length && c <= vards.Length)

{

int val = rand.Next(uzvards.Length, uzvards.Length + vards.Length + 1);

swappedVU[r, c] = val.ToString();

}

else

swappedVU[r, c] = "";

}

}

PrintTable(swappedVU, cols, rows + 1, false);

}

}