**Uzd 1.///**

using System;

using System.Linq;

class Program

{

static void Main()

{

Console.Write("Ievadi vardu: ");

string vards = Console.ReadLine();

Console.Write("Ievadi uzvardu: ");

string uzvards = Console.ReadLine();

string formattedVards = FormatAlternateUpper(vards);

string formattedUzvards = FormatAlternateUpper(new string(uzvards.Reverse().ToArray()));

int rowCount = uzvards.Length;

int colCount = vards.Length;

int[,] table = new int[rowCount, colCount];

Random rand = new Random();

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

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

table[i, j] = rand.Next(vards.Length, vards.Length + uzvards.Length + 1);

PrintTable(formattedVards, formattedUzvards, table);

int[,] transposedTable = TransposeMatrix(table);

PrintTable(formattedUzvards, formattedVards, transposedTable);

}

static string FormatAlternateUpper(string input)

{

char[] result = input.ToCharArray();

for (int i = 1; i < result.Length; i += 2)

result[i] = char.ToUpper(result[i]);

return new string(result);

}

static void PrintTable(string vards, string uzvards, int[,] table)

{

int rowCount = table.GetLength(0);

int colCount = table.GetLength(1);

string topBottomBorder = "+" + string.Concat(Enumerable.Repeat("-------+", colCount + 2));

string midBorder = "\*" + string.Concat(Enumerable.Repeat("======", colCount + 4 )) + "\*";

Console.WriteLine();

Console.WriteLine(midBorder);

Console.Write("| X ");

foreach (char c in vards)

Console.Write($"| {c,-2} ");

Console.WriteLine(" | sum | min | avg | max |");

Console.WriteLine(midBorder);

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

{

Console.Write($"| {uzvards[i],-2} ");

int sum = 0, min = table[i, 0], max = table[i, 0];

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

{

int num = table[i, j];

sum += num;

if (num < min) min = num;

if (num > max) max = num;

Console.Write($"| {num,-2} ");

}

double avg = (double)sum / colCount;

Console.Write($"| {sum,-3} | {min,-3} | {avg,4:F1} | {max,-3} |");

Console.WriteLine();

Console.WriteLine(topBottomBorder);

}

PrintColumnStats(table);

}

static void PrintColumnStats(int[,] table)

{

int rowCount = table.GetLength(0);

int colCount = table.GetLength(1);

Console.Write("| sum ");

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

Console.Write($"| {Enumerable.Range(0, rowCount).Sum(i => table[i, j]),-2} ");

Console.WriteLine("|");

Console.Write("| min ");

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

Console.Write($"| {Enumerable.Range(0, rowCount).Min(i => table[i, j]),-2} ");

Console.WriteLine("|");

Console.Write("| max ");

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

Console.Write($"| {Enumerable.Range(0, rowCount).Max(i => table[i, j]),-2} ");

Console.WriteLine("|");

Console.WriteLine("\*" + string.Concat(Enumerable.Repeat("====", colCount + 3)) + "\*");

}

static int[,] TransposeMatrix(int[,] matrix)

{

int rows = matrix.GetLength(0);

int cols = matrix.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] = matrix[i, j];

return transposed;

}

}

**2.uzd**

using System;

using System.Collections.Generic;

using System.Linq;

class Program

{

static void Main()

{

Console.Write("Ievadiet paroļu SKAITU, kas lielāks par 4: ");

int passwordCount = GetValidNumber(5);

Console.Write("Ievadiet GARĀKO paroles simbolu skaitu, kas lielāks par 9: ");

int maxPasswordLength = GetValidNumber(10);

Console.Write("Vajag LIELUS burtus simboliem (y/n): ");

bool useUpperCase = Console.ReadLine().Trim().ToLower() == "y";

Console.Write("Vajag ciparus un simbolus simboliem (y/n): ");

bool useDigitsAndSymbols = Console.ReadLine().Trim().ToLower() == "y";

char[] lowerCaseLetters = "abcdefghijklmnopqrstuvwxyz".ToCharArray();

char[] upperCaseLetters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ".ToCharArray();

char[] digits = "0123456789".ToCharArray();

char[] symbols = "!@#$%^&\*()\_+-=[]{}|;:',.<>?/".ToCharArray();

Random rand = new Random();

List<char[]> passwords = new List<char[]>();

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

{

int passwordLength = rand.Next(5, maxPasswordLength + 1);

List<char> availableChars = lowerCaseLetters.ToList();

if (useUpperCase) availableChars.AddRange(upperCaseLetters);

if (useDigitsAndSymbols)

{

availableChars.AddRange(digits);

availableChars.AddRange(symbols);

}

char[] password = new char[passwordLength];

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

password[j] = availableChars[rand.Next(availableChars.Count)];

passwords.Add(password);

}

Console.WriteLine("\nĢenerētās paroles:");

int index = 1;

foreach (char[] password in passwords)

{

Console.WriteLine($"{index}. parole: {new string(password)}");

index++;

}

}

static int GetValidNumber(int minValue)

{

int number;

while (!int.TryParse(Console.ReadLine(), out number) || number < minValue)

Console.Write($"Ievadiet skaitli, kas ir vismaz {minValue}: ");

return number;

}

}