1 uzd

using System;

using System.Linq;

class MasivuAnalize

{

static void Main()

{

Console.Write("Ievadi savu vārdu: ");

string vards = Console.ReadLine();

Console.Write("Ievadi savu uzvārdu: ");

string uzvards = Console.ReadLine();

string vardsFormatets = FormatetVardu(vards);

string uzvardsFormatets = FormatetVardu(uzvards);

string vardsReverss = JaunsReverss(vardsFormatets);

string uzvardsReverss = JaunsReverss(uzvardsFormatets);

int rinduSkaits = uzvardsFormatets.Length;

int kolonnuSkaits = vardsFormatets.Length;

int[,] dati = new int[rinduSkaits, kolonnuSkaits];

Random rng = new Random();

// Masīva aizpildīšana

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

{

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

{

dati[r, c] = rng.Next(kolonnuSkaits, kolonnuSkaits + rinduSkaits + 1);

}

}

// Galvenā tabula

Console.WriteLine("\n--- Galvenā tabula ---\n");

IzvadeTabula(dati, uzvardsReverss, vardsFormatets);

// Apgrieztā tabula (transponēta)

Console.WriteLine("\n--- Apgrieztā tabula ---\n");

int[,] datiTransponeti = TransponetMasivu(dati);

IzvadeTabula(datiTransponeti, vardsReverss, uzvardsFormatets);

}

// Formatēšana: katrs otrais lielais

static string FormatetVardu(string teksts)

{

char[] simboli = teksts.ToCharArray();

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

{

simboli[i] = i % 2 == 0 ? char.ToLower(simboli[i]) : char.ToUpper(simboli[i]);

}

return new string(simboli);

}

// Teksta reverss

static string JaunsReverss(string teksts)

{

return new string(teksts.Reverse().ToArray());

}

// Tabulas izvade ar sum, min, max

static void IzvadeTabula(int[,] masivs, string rindasNosaukumi, string kolonnuNosaukumi)

{

int rindas = masivs.GetLength(0);

int kolonnas = masivs.GetLength(1);

// Galvene

Console.Write("| X |");

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

{

Console.Write($" {kolonnuNosaukumi[i]} |");

}

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

Console.WriteLine();

Console.WriteLine(new string('-', 10 + kolonnas \* 8 + 24));

// Rindas

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

{

Console.Write($"| {rindasNosaukumi[r]} |");

int[] rinda = new int[kolonnas];

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

{

rinda[c] = masivs[r, c];

Console.Write($" {rinda[c],5} |");

}

Console.Write($" {rinda.Sum(),5} | {rinda.Min(),5} | {rinda.Max(),5} |");

Console.WriteLine();

}

// Apakšējās statistikas rindas

Console.WriteLine(new string('-', 10 + kolonnas \* 8 + 24));

string[] etiketes = { "Sum", "Min", "Max" };

for (int stat = 0; stat < 3; stat++)

{

Console.Write($"| {etiketes[stat]} |");

int[] statVertibas = new int[kolonnas];

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

{

int[] kolonna = new int[rindas];

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

{

kolonna[r] = masivs[r, c];

}

int vertiba = stat switch

{

0 => kolonna.Sum(),

1 => kolonna.Min(),

2 => kolonna.Max(),

\_ => 0

};

statVertibas[c] = vertiba;

Console.Write($" {vertiba,5} |");

}

int total = stat switch

{

0 => statVertibas.Sum(),

1 => statVertibas.Min(),

2 => statVertibas.Max(),

\_ => 0

};

// \* vietā starpnieka kolonnām

Console.Write(stat == 0 ? $" {total,5} | \* | \* |" :

stat == 1 ? $" \* | {total,5} | \* |" :

$" \* | \* | {total,5} |");

Console.WriteLine();

}

Console.WriteLine(new string('-', 10 + kolonnas \* 8 + 24));

}

// Masīva transponēšana

static int[,] TransponetMasivu(int[,] original)

{

int rindas = original.GetLength(0);

int kolonnas = original.GetLength(1);

int[,] transponets = new int[kolonnas, rindas];

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

{

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

{

transponets[c, r] = original[r, c];

}

}

return transponets;

}

}

2.uzd

using System;

class ParolesGenerator

{

static void Main()

{

int paroluSkaits = IevadeSkaitlis("Ievadi parolu skaitu (min 5): ", 5);

int simboluSkaits = IevadeSkaitlis("Ievadi maksimālo paroles garumu (min 10): ", 10);

bool izmantotLielosBurtus = IevadeJautajums("Vai pievienot lielos burtus? (y/n): ");

bool izmantotSimbolusUnCiparus = IevadeJautajums("Vai pievienot simbolus un ciparus? (y/n): ");

string[] paroles = new string[paroluSkaits];

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

{

paroles[i] = GeneretParoli(simboluSkaits, izmantotLielosBurtus, izmantotSimbolusUnCiparus);

}

Console.WriteLine("\n--- Ģenerētās paroles ---\n");

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

{

Console.WriteLine($"{i + 1}. parole: {paroles[i]}");

}

}

// --- Metodes zemāk ---

static int IevadeSkaitlis(string jautajums, int minimums)

{

int skaitlis;

Console.Write(jautajums);

while (!int.TryParse(Console.ReadLine(), out skaitlis) || skaitlis < minimums)

{

Console.Write($"Kļūda! Lūdzu ievadi skaitli, kas ir vismaz {minimums}: ");

}

return skaitlis;

}

static bool IevadeJautajums(string jautajums)

{

Console.Write(jautajums);

string atbilde = Console.ReadLine().Trim().ToLower();

while (atbilde != "y" && atbilde != "n")

{

Console.Write("Kļūda! Atbildei jābūt 'y' vai 'n': ");

atbilde = Console.ReadLine().Trim().ToLower();

}

return atbilde == "y";

}

static string GeneretParoli(int garums, bool ieklautLielosBurtus, bool ieklautSimbolus)

{

string burti = "abcdefghijklmnopqrstuvwxyz";

string lielieBurti = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

string simboliUnCipari = "1234567890!@#$%^&\*()-\_+=[]{}<>?";

string simboluKomplekts = burti;

if (ieklautLielosBurtus)

simboluKomplekts += lielieBurti;

if (ieklautSimbolus)

simboluKomplekts += simboliUnCipari;

Random rnd = new Random();

char[] parole = new char[garums];

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

{

int index = rnd.Next(simboluKomplekts.Length);

parole[i] = simboluKomplekts[index];

}

return new string(parole);

}

}