1 uzd

using System;

class Program

{

static void Main()

{

Console.WriteLine("Mans vards: NIKS");

Console.WriteLine("Mans uzvards: BORODINS");

int minRows = 5, minCols = 5;

int rows, cols;

// Pieprasa ievadit rindu skaitu, kam jabut vismaz 5

do

{

Console.Write($"Ievadi rindu skaitu, ne mazak ka {minRows}: ");

} while (!int.TryParse(Console.ReadLine(), out rows) || rows < minRows);

// Pieprasa ievadit kolonnu skaitu, kam jabut vismaz 5

do

{

Console.Write($"Ievadi kolonu skaitu, ne mazak ka {minCols}: ");

} while (!int.TryParse(Console.ReadLine(), out cols) || cols < minCols);

Console.WriteLine();

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

{

PrintLine(i, cols);

PrintNumbers(i, cols);

}

PrintLine(rows, cols);

}

static void PrintLine(int row, int totalCols)

{

int displayCols = Math.Min(row, totalCols); // Ierobezo kolonnu skaitu, lai neparsniegtu max

int spaces = Math.Max(0, totalCols - displayCols); // Izrekina nepieciesamo tabulas daudzumu

Console.Write(new string(' ', spaces \* 4));

Console.Write("+");

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

{

Console.Write("---+");

}

Console.WriteLine();

}

static void PrintNumbers(int row, int totalCols)

{

int displayCols = Math.Min(row, totalCols); // Nodrosina ka kolonnu skaits neparsniedz max

int spaces = Math.Max(0, totalCols - displayCols); // Izrekina tabulas daudzumu, lai nolidzinatu rindas

Console.Write(new string(' ', spaces \* 4));

Console.Write("|");

for (int i = row - displayCols + 1; i <= row; i++) // Ja kolonnas parsniedz robezu, izdzes pirmos skaitlus

{

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

}

Console.WriteLine();

}

}

2 uzd

using System;

class Program

{

static void Main()

{

// Lietotāja vārda un uzvārda definēšana un formatēšana

string persona = "NIKS BORODINS";

string[] vardsUzvards = persona.ToLower().Split(' ');

string vards = char.ToUpper(vardsUzvards[0][0]) + vardsUzvards[0].Substring(1);

string uzvards = char.ToUpper(vardsUzvards[1][0]) + vardsUzvards[1].Substring(1);

Console.WriteLine($"Mans vārds: {vards}");

Console.WriteLine($"Mans uzvārds: {uzvards}");

int kolonas;

do

{

Console.Write("Ievadi kolonu skaitu, ne mazāk kā 5: ");

} while (!int.TryParse(Console.ReadLine(), out kolonas) || kolonas < 5);

Console.WriteLine();

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

{

Console.Write(new string(' ', (kolonas - i - 1) \* 6));

// Rindas līnija

Console.Write("+");

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

{

Console.Write(" --- +");

}

Console.WriteLine();

Console.Write(new string(' ', (kolonas - i - 1) \* 6));

// Ciparu rinda (sākas no 1)

Console.Write("|");

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

{

Console.Write($" {j + 1} |"); // Cipari sākas no 1

}

Console.WriteLine();

}

Console.Write(new string(' ', (kolonas - kolonas) \* 6)); // Lai izlīdzinātu

Console.Write("+");

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

{

Console.Write(" --- +");

}

Console.WriteLine();

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

{

// Rindas līnija

Console.Write("+");

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

{

Console.Write(" --- +");

}

Console.WriteLine();

Console.Write("|");

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

{

Console.Write($" {j + 1} |");

}

Console.WriteLine();

}

// Otrā daļa apakšējā rinda

Console.Write("+");

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

{

Console.Write(" --- +");

}

Console.WriteLine();

}

}

3 uzd

using System;

using System.Linq;

using System.Text;

class Program

{

static void Main()

{

// Pieprasa ievadīt tekstu

Console.Write("Ievadi tekstu: ");

string input = Console.ReadLine();

// Izvada tekstu, kur vārdi salikti pretējā virzienā un teikums sākas ar Lielo burtu

string reversedWords = ReverseWordsOrder(input);

Console.WriteLine("Pretējā virzienā vārdi: " + CapitalizeFirstLetter(reversedWords));

// Izvada tekstu pretējā virzienā un samainītu reversā veidā un teikums sākas ar Lielo burtu

string reversedText = ReverseText(reversedWords);

Console.WriteLine("Pretējā virzienā vārdi un reversā: " + CapitalizeFirstLetter(reversedText));

// Izvada tekstu sākot no vidus, katrā ciklā parādot pa vienam burtam abās pusē, lai teksts paliek centrēts

CenteredTextDisplay(input);

// Sāk no vidus (ja nepāra skaits tad vienu burtu) un aizstāj ar tukšumu, kamēr visi burti aizstāti

CenteredTextReplace(input);

// Turpina līdzīgi, bet sākot rādīt burtus no beigām līdz sasniedzot sākuma burtu

RevealFromEnd(input);

}

static string ReverseWordsOrder(string text)

{

string[] words = text.Split(' ');

Array.Reverse(words);

return string.Join(" ", words);

}

static string ReverseText(string text)

{

char[] charArray = text.ToCharArray();

Array.Reverse(charArray);

return new string(charArray);

}

static string CapitalizeFirstLetter(string text)

{

if (string.IsNullOrEmpty(text))

{

return text;

}

return char.ToUpper(text[0]) + text.Substring(1);

}

static void CenteredTextDisplay(string text)

{

int length = text.Length;

StringBuilder sb = new StringBuilder(new string(' ', length));

for (int i = 0; i <= length / 2; i++)

{

sb[length / 2 - i] = text[length / 2 - i];

sb[length / 2 + i] = text[length / 2 + i];

Console.WriteLine(sb.ToString());

}

}

static void CenteredTextReplace(string text)

{

int length = text.Length;

StringBuilder sb = new StringBuilder(text);

for (int i = 0; i <= length / 2; i++)

{

sb[length / 2 - i] = ' ';

sb[length / 2 + i] = ' ';

Console.WriteLine(sb.ToString());

}

}

static void RevealFromEnd(string text)

{

int length = text.Length;

StringBuilder sb = new StringBuilder(new string(' ', length));

for (int i = length - 1; i >= 0; i--)

{

sb[i] = text[i];

Console.WriteLine(sb.ToString());

}

}

}

4 uzd

using System;

namespace TabulasNobide2

{

class Program

{

static void Main(string[] args)

{

string vards = "NIKS";

string uzvards = "BORODINS";

// Izvada vārdu un tā garumu

Console.WriteLine($"Mans vārds ir {Capitalize(vards)} un tajā ir {vards.Length} simboli");

// Izvada uzvārdu un tā garumu

Console.WriteLine($"Mans uzvārds ir {Capitalize(uzvards)} un tajā ir {uzvards.Length} simboli");

// Izvada tabulu

Console.WriteLine();

IzvadeTabulu(vards.ToLower(), uzvards.ToLower());

}

static string Capitalize(string word)

{

return char.ToUpper(word[0]) + word.Substring(1).ToLower();

}

static void IzvadeTabulu(string vards, string uzvards)

{

int rindas = vards.Length;

int kolonas = uzvards.Length;

Random random = new Random();

Console.Write(" ");

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

{

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

}

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

Console.WriteLine("+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+");

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

{

int[] values = new int[rindas];

Console.Write($"| {uzvards[j]} ");

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

{

values[i] = random.Next(vards.Length, vards.Length + uzvards.Length + 1);

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

}

Console.WriteLine($"| {Max(values)} | {Min(values)} | {Vid(values):0.0} | {Sum(values)} |");

Console.WriteLine("+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+");

}

}

static int Max(int[] values)

{

int max = values[0];

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

{

if (values[i] > max)

{

max = values[i];

}

}

return max;

}

static int Min(int[] values)

{

int min = values[0];

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

{

if (values[i] < min)

{

min = values[i];

}

}

return min;

}

static double Vid(int[] values)

{

return (double)Sum(values) / values.Length;

}

static int Sum(int[] values)

{

int sum = 0;

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

{

sum += values[i];

}

return sum;

}

}

}