**1. uzdevums: Tabula izveide ar metodes un funkciju palīdzību**

Izveidojiet C# konsoles lietojumprogrammu.

Uzrakstiet **metodi** ar nosaukumu "**TabulasLinija**", kas saņem divus parametrus:

1. svītriņu skaits,
2. reižu skaits.

Piemēram :

\*---\*---\*---\*---\*

1- parametrs saņem vērtību-> 3 (trīs svītriņas starp zvaigznēm),

2- parametrs->4 ( 4x izpildās)

Uzrakstiet **metodi** ar nosaukumu "**TabulasRinda**", kas saņem 4. parametrus:

1. Rindas dati (1D masīvs),
2. Max vērtību
3. Min vertibu,
4. Sum vērtību

…un izdrukā (piemēram)

| 1 | 2 | 3 | 4 | max | min | sum |

Piemēram :

Int[] dati={1,2,3,4}

Uzrakstiet **funkciju** ar nosaukumu "**TabulasDati**", kas saņem divus parametrus:

1. vārdu
2. uzvārdu

Pēc šiem datiem **atgriež** 2D masīvu, kas uzģenerē datus ar **gadījuma** skaitļiem robežās no **vārda garuma** līdz **varda+uzvarda garumam (ieskaitot)**

Uzrakstiet **metodi** ar nosaukumu "**TabulaGalva**", kas saņem parametru:

1. Vārds

Izmantojot metodes **TabulasLinija** un **TabulasRinda** izvada tabulas galvā **vārdu un Min, Max, Sum**

Uzrakstiet **metodi** ar nosaukumu "**Tabula**", kas saņem parametru:

1. **2D masīvu**

Izmantojot metodes **TabulasLinija** un **TabulasRinda** izvada tabulas galvā **rindas datus un Min, Max, Sum vērtības**

Uzrakstiet **funkciju** ar nosaukumu "**TabulasMaina**", kas saņem parametru:

1. **2D masīvu**

Funkcija izveido 2D masīvu, kas apmaina kolonas ar rindas datiem vietām un atgriež **2D masīvu**

"**Main**" metodē izsaucam:

**TabulasDati…**

**TabulaGalva…**

**Tabula….**

**Console.WriteLine(….)**

**TabulasMaina…**

**TabulaGalva…**

**Tabula….**

**using System.Threading.Tasks;**

**namespace System {**

**public class Values //Globāli saglabājam datus**

**{**

**public int Svit\_Sk = 4;**

**public int Reiz\_Sk = 5;**

**public int Uzv = 9;**

**public static int[,] \_2d = new int[9, 9];**

**public int[] \_1d = new int[9];**

**public string Vards = "Kristians";**

**public static void TabulasDati() //Izveidojam datus 2d masīvā**

**{**

**var valve = new Values();**

**Random rand = new Random();**

**for (int i = 0; i < valve.Uzv; i++)**

**{**

**for (int b = 0; b < valve.Vards.Length; b++)**

**{**

**\_2d[i, b] = rand.Next(valve.Vards.Length, valve.Vards.Length + valve.Uzv);**

**}**

**}**

**}**

**}**

**public class Program**

**{**

**public static void Main()//norādam kādā secībā dati tiks izpildīti**

**{**

**Values.TabulasDati();**

**TabulasGalva();**

**Tabula();**

**Console.WriteLine("\t\t\t\t\t|\t|\t|\n\t\t\t\t\t\*-----SPLIT-----\*\n\t\t\t\t\t|\t|\t|");**

**TabulasMaina();**

**TabulasGalva();**

**Tabula();**

**}**

**static void TabulasLinija()//metode kas izprintēs konsolē līnijas, ar specifiskiem norādījumiem cik reižu un cik daudz līniju pa vidu**

**{**

**var valve = new Values();**

**for (int i = 0; i < valve.Vards.Length + 3; i++)**

**{**

**Console.Write("\*  ");**

**for (int a = 0; a < valve.Svit\_Sk; a++)**

**{**

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

**}**

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

**}**

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

**}**

**public static void TabulasRinda(int a)//izprintējam pa masīva punktiem datus, un sarēķinam mazāko, maksimālo un summu**

**{**

**var valve = new Values();**

**int b = a;**

**for (int i = 0; i < valve.Vards.Length; i++)**

**{**

**Console.Write($"|  {Values.\_2d[b, i]}\t");**

**valve.\_1d[i] = Values.\_2d[b, i];//pieliekam globālo vērtību pie lokālās vērtības**

**}**

**Console.Write($"|  {valve.\_1d.Min()}\t|  {valve.\_1d.Max()}\t|  {valve.\_1d.Sum()}\t|\n");**

**}**

**static void TabulasGalva()//izprintējam doto vārdu un sadaļas nosaukumus**

**{**

**var valve = new Values();**

**TabulasLinija();**

**Console.WriteLine($"|\t\t\t\t{valve.Vards}\t\t\t\t|  Min\t|  Max\t|  Sum\t|");**

**TabulasLinija();**

**}**

**static void Tabula()//norādam cik reižu, un kā izprintēt tabulu**

**{**

**int b = 0;**

**var valve = new Values();**

**for (int i = 0; i < valve.Uzv; i++)**

**{**

**TabulasRinda(b);**

**TabulasLinija();**

**b++;**

**}**

**}**

**static void TabulasMaina()// samainam vertikālo asi ar horizontālo asi**

**{**

**var valve = new Values();**

**int[,] \_2d = new int[valve.Uzv, valve.Vards.Length];**

**for (int i = 0; i < valve.Uzv; i++)**

**{**

**for (int b = 0; b < valve.Vards.Length; b++)**

**{**

**\_2d[b, i] = Values.\_2d[i, b];**

**}**

**}**

**Values.\_2d = \_2d;//pieliekam lokālo vērtību pie globālās vērtības**

**}**

**}**

**}**