Day 13 Assignment

By

VARUN SAI KUMAR CHEGONI

NB Healthcare and Technology

Date: 09 Feb 2022

**Topics**

**C# 2-D Arrays**

**C# Jagged Arrays**

**C# Recursion**

**C# Stack and Queue**

**Content**

|  |  |  |
| --- | --- | --- |
| S.No | Content | Page No. |
| 1. | Declare a 2-dimensional array of size (2,2) and initialize using indexes and print the values using nested for loop. | 3 |
| 2. | Declare a 2-D array of size (3,2) and initialize in the same line while declaring and print the values using nested for loop. | 4 |
| 3. | Declare a 2-D array of size (3,3) and print trace of the array | 5 |
| 4. | Declare a 2-D array of size (2,2) and read values from user and print the array values. | 6 |
| 5. | Declare TWO 2-D arrays of size (2,2) and read values from user and print the sum of the two matrices. | 7 |
| 6. | Declare TWO 2-D arrays of size (2,2) and read values from user and print the product of the two matrices. | 9 |
| 7. | What is a jagged array? What is the benefit of jagged array? | 11 |
| 8. | WACP to declare a jagged array and print values |  |
| 9. | What is Recursion. |  |
| 10. | WACP to illustrate usage of Recursion. What are the benefits of recursion? |  |
| 11. | WACP to illustrate usage of Stack< >. Write couple of points about Stack. |  |
| 12. | WACP to illustrate usage of Queue< >. Write couple of points about Queue. |  |

|  |
| --- |
| 1. Declare a 2-dimensional array of size (2,2) and initialize using indexes and print the values using nested for loop. |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_2DArrayIntiIndexes  {  internal class Program  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author : Varun Sai Kumar Chegoni.  \* Purpose : 2 dimentional array of size (2,2) and initialize using indexes and print the values using nested for loop  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  static void Main(string[] args)  {  // Declaration  int[,] a = new int[2,2];  // indexes Initialization  a[0,0] = 5;  a[0,1] = 7;  a[1,0] = 4;  a[1,1] = 6;  // Nested for Loop  for(int i = 0; i < 2; i++)  {  for(int j = 0; j < 2; j++)  {  Console.Write(a[i,j]+" "); // Print Output  }  Console.WriteLine();  }  Console.ReadLine();  }  }  } |
| Output : |
|  |
| 2. Declare a 2-D array of size (3,2) and initialize in the same line while declaring and print the values using nested for loop. |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_2DArrayIntiIndexes  {  internal class Program  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author : Varun Sai Kumar Chegoni.  \* Purpose : 2-D array of size (3,2) and initialize in the same line while declaring and print the values using nested for loop  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  static void Main(string[] args)  {  // Declaration and Intialization in same line  int[,] a = new int[,] { {3,2,1}, {1,2,3} };  // Nested for Loop  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 3; j++)  {  Console.Write(a[i, j]+" "); // Print Output  }  Console.WriteLine();  }  Console.ReadLine();  }  }  } |
| Output : |
|  |

|  |
| --- |
| 3. Declare a 2-D array of size (3,3) and print trace of the array |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_2DArrayIntiIndexes  {  internal class Program  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author : Varun Sai Kumar Chegoni.  \* Purpose : Declare a 2-D array of size (3,3) and print trace of the array  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  static void Main(string[] args)  {  // Declaration and Intialization in same line  int[,] a = new int[,] { { 1, 2, 3 }, { 2, 4, 6 }, { 3, 6, 9 } };  int sum = 0;  // Nested for Loop  for (int i = 0; i <= 2; i++)  {  for (int j = 0; j <= 2; j++)  {  if (i==j)  sum = sum + a[i, j]; // Logic  }  }  Console.WriteLine("Trace of the Array= "+sum);  Console.ReadLine();  }  }  } |
| Output : |
|  |

|  |
| --- |
| 4. Declare a 2-D array of size (2,2) and read values from user and print the array values. |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_2DArrayIntiIndexes  {  internal class Program  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author : Varun Sai Kumar Chegoni.  \* Purpose : Declare a 2-D array of size (2,2) and read values from user and print the array values.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  static void Main(string[] args)  {  // Declaration  int[,] a = new int[2,2];  // Read Value from User  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine($"Enter Value of ({i},{j}): ");  a[i,j]=Convert.ToInt32(Console.ReadLine());  }  }  // Print Output  for(int i = 0; i < 2; i++)  {  for(int j = 0; j < 2; j++)  Console.Write(a[i,j]+" ");  Console.Write("\n");  }  Console.Write("\n");  Console.ReadLine();  }  }  } |
| Output : |
|  |
| 5. Declare TWO 2-D arrays of size (2,2) and read values from user and print the sum of the two matrices. |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_2DArrayMatrixAddition  {  internal class Program  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author : Varun Sai Kumar Chegoni.  \* Purpose : Declare TWO 2-D arrays of size (2,2) and read values from user and print the sum of the two matrices.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  static void Main(string[] args)  {  int i, j;  int[,] mat1 = new int[2, 2];  int[,] mat2 = new int[2, 2];  int[,] matsum = new int[2, 2];  Console.Write("Enter elements in the first matrix:\n");  for (i = 0; i < 2; i++)  {  for (j = 0; j < 2; j++)  {  mat1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  Console.Write("Enter elements in the second matrix:\n");  for (i = 0; i < 2; i++)  {  for (j = 0; j < 2; j++)  {  mat2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  Console.Write("\nFirst matrix is:\n");  for (i = 0; i < 2; i++)  {  Console.Write("\n");  for (j = 0; j < 2; j++)  Console.Write("{0} ", mat1[i, j]);  }  Console.Write("\nSecond matrix is:\n");  for (i = 0; i < 2; i++)  {  Console.Write("\n");  for (j = 0; j < 2; j++)  Console.Write("{0} ", mat2[i, j]);  }  for (i = 0; i < 2; i++)  for (j = 0; j < 2; j++)  matsum[i, j] = mat1[i, j] + mat2[i, j];  Console.Write("\nAdding two matrices: \n");  for (i = 0; i < 2; i++)  {  Console.Write("\n");  for (j = 0; j < 2; j++)  Console.Write("{0} ", matsum[i, j]);  }  Console.Write("\n\n");  Console.ReadLine();  }  }  } |
| Output : |
|  |

|  |
| --- |
| 6. Declare TWO 2-D arrays of size (2,2) and read values from user and print the product of the two matrices. |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  namespace \_2DArrayMatrixProduct  {  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Author : Varun Sai Kumar Chegoni.  \* Purpose : Declare TWO 2-D arrays of size (2,2) and read values from user and print the product of the two matrices.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  class MatrixMultiplication  {  int[,] a = new int[2, 2];  int[,] b = new int[2, 2];  int[,] c = new int[2, 2];  public void ReadMatrix()  {  Console.WriteLine("\n Enter the elements of Matrix 1:");  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  a[i, j] = Convert.ToInt16(Console.ReadLine());  }  }  Console.WriteLine("\n Enter the elements of Matrix 2:");  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  b[i, j] = Convert.ToInt16(Console.ReadLine());  }  }  }  public void PrintMatrix()  {  Console.WriteLine("\n Matrix 1:");  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(" " + a[i, j]);  }  Console.WriteLine();  }  Console.WriteLine("\n Matrix 2:");  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(" " + b[i, j]);  }  Console.WriteLine();  }  Console.WriteLine("\n Resultant Matrix after multiplying:");  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(" " + c[i, j]);  }  Console.WriteLine();  }  Console.ReadLine();  }  public void MultiplyMatrix()  {  c = new int[2, 2];  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  c[i, j] = 0;  for (int k = 0; k < 2; k++)  c[i, j] = c[i, j] + a[i, k] \* b[k, j];  }  }  }  }  class Matrices  {  public static void Main()  {  MatrixMultiplication MM = new MatrixMultiplication();  MM.ReadMatrix();  MM.MultiplyMatrix();  MM.PrintMatrix();  }  }  } |
| Output : |
|  |
| 7. What is a jagged array. What is the benefit of jagged array |
| Answer: |
|  |

|  |
| --- |
| 8. WACP to declare a jagged array and print values |
| Code : |
|  |
| Output : |
|  |

|  |
| --- |
| 9. What is Recursion |
| Answer: |
|  |

|  |
| --- |
| 10. WACP to illustrate usage of Recursion. What are the benefits of recursion? |
| Code : |
|  |
| Output : |
|  |

|  |
| --- |
| 11. WACP to illustrate usage of Stack< >. Write couple of points about Stack |
| Code : |
|  |
| Output : |
|  |
| Points about Stack |
|  |

|  |
| --- |
| 12. WACP to illustrate usage of Queue< >. Write couple of points about Queue |
| Code : |
|  |
| Output : |
|  |
| Points about Queue |
|  |