# DAY 13 ASSIGNMENT

# (BY G V S S SRI LASYA)

|  |
| --- |
| 1) Declare a 2 dimentional 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;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Declare a 2 dimentional array of size (2,2) and initialize using indices  and print the values using nested for loop  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project1  {  internal class Program  {  static void Main(string[] args)  {  //crearing array and initialising it using index  int[,] matrix1 = new int[2, 2];  matrix1[0, 0] = 1;  matrix1[0, 1] = 2;  matrix1[1,0] = 3;  matrix1[1,1] = 4;  Console.WriteLine("\n\nPrinting a 2\*2 matrix\n\n\n\n");  //printing the values of matrix using for loop  for(int i = 0; i < 2; i++)  {  for(int j = 0; j < 2; j++)  Console.Write(matrix1[i, j] + "\t");  Console.WriteLine("\n");  }  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;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : 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  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project2  {  internal class Program  {  static void Main(string[] args)  {  //creating an array and initialising it during declaration  int[,] matrix1 = new int[,] { { 1, 2 }, { 3, 4 },{ 5, 6 } };  Console.Write("\n\nPrinting a 3\*2 matrix\n\n\n\n");  //printing matrix1 using for loop  for (int i = 0; i < 3; i++)  {  for(int j = 0; j < 2; j++)  Console.Write(matrix1[i, j] + "\t");  Console.WriteLine("\n");  }  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 3) 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;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Declare a 2-D array of size (3,3) and read values from  user and print the array values.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project3  {  internal class Program  {  static void Main(string[] args)  {    int[,] matrix1 = new int[3, 3];  Console.Write("\n\nTaking user input for a 3\*3 matrix and printing out the matrix");  //taking user inputs for the matrix  Console.Write("\n\n\n\nEnter inputs for the matrix\n\n");  for(int i = 0; i< 3; i++)  {  for(int j = 0; j< 3; j++)  {  Console.Write($"Enter number matrx1[{i},{j}]: ");  matrix1[i,j] = Convert.ToInt32(Console.ReadLine());  }  }  //printing the matrix  Console.Write("\n\n\nPrinting out the matrix\n\n");  for(int i = 0; i< 3; i++)  {  for(int j = 0; j< 3; j++)  Console.Write(matrix1[i,j] + "\t");  Console.Write("\n\n");  }  Console.ReadLine();    }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 4) 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;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Declare a 2-D array of size (3,3) and print trace of the array  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project4  {  internal class Program  {  static void Main(string[] args)  {  int i,j,trace = 0;  int[,] matrix1 = new int[3, 3] { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 } };  Console.Write("\n\nPrinting out trace of the matrix\n\n\n");  //calculating trace of the matrix  for (i = 0; i < 3; i++)  {  for (j = 0; j < 3; j++)  {  if (i == j)  trace += matrix1[i, j];  }  }  Console.Write($"\nTrace of the matrix {matrix1[0,0]} + {matrix1[1,1]} + {matrix1[2,2]} : {trace}");  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;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Declare TWO 2-D arrays of size (2,2) and read values from user and  print the sum of the two matrices.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project5  {  internal class Program  {  static void Main(string[] args)  {  int[,] matrix1 = new int[2, 2];  int[,] matrix2 = new int[2, 2];  int[,] sumMatrix = new int[2, 2];  Console.WriteLine("\nTaking user inputs for 2 matrices,calculating sum and printing the matrix");  //taking user inputs for matrix1  Console.Write("\n\n\n\nTaking inputs for matrix1\n\n");  for (int i = 0; i < 2; i++)  {  for(int j = 0; j < 2; j++)  {  Console.Write($"\nEnter the number matrix1[{i},{j}]: ");  matrix1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  //taking user inputs for matrix2  Console.Write("\n\n\n\nTaking inputs for matrix2\n\n");  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write($"\nEnter the number matrix2[{i},{j}]: ");  matrix2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  //calculating sum matrix  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  sumMatrix[i, j] = matrix1[i,j] + matrix2[i,j];  }  //printing out the sum matrix  Console.Write("\n\n\n\nPrinting out the sum matrix\n\n");  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  Console.Write(sumMatrix[i, j] + "\t");  Console.WriteLine("\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;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : Declare TWO 2-D arrays of size (2,2) and read values from user and  print the product of the two matrices.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project6  {  internal class Program  {  static void Main(string[] args)  {  int m,n,p;  int[,] matrix1 ;  int[,] matrix2;  int[,] productMatrix;  Console.Write("\n\nTaking user inputs for 2 2D matrices,calculating product and printing the product matrix");  Console.Write("\n\n\n\nEnter the number of rows of the matrix1 : ");  m = Convert.ToInt32(Console.ReadLine());  Console.Write("\nEnter the number of columns of the matrix2 : ");  p = Convert.ToInt32(Console.ReadLine());  Console.Write("\nPlease ensure that the number of columns of matrix1 and number of rows of matrix2 are equal,give a number : ");  n = Convert.ToInt32(Console.ReadLine());  matrix1 = new int[m,n];  matrix2 = new int[n,p];  productMatrix = new int[m,p];  //taking user inputs for matrix1  Console.Write("\n\n\n\nTaking inputs for matrix1\n");  for (int i = 0; i < m; i++)  {  for (int j = 0; j < n; j++)  {  Console.Write($"\nEnter the number for matrix1[{i},{j}]: ");  matrix1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  //taking user inputs for matrix2  Console.Write("\n\n\n\nTaking inputs for matrix2\n");  for (int i = 0; i < n; i++)  {  for (int j = 0; j < p; j++)  {  Console.Write($"\nEnter the number for matrix2[{i},{j}]: ");  matrix2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  //calculating product matrix  for (int i = 0; i < m; i++)  {  for(int j = 0; j < p; j++)  {  for(int k = 0; k < n; k++)  {  productMatrix[i, j] += matrix1[i, k] \* matrix2[k, j];  }  }  }  //printing out the product matrix  Console.Write("\n\n\n\nPrinting out the product matrix\n\n");  for (int i = 0; i < m; i++)  {  for (int j = 0; j < p; j++)  Console.Write(productMatrix[i,j] + "\t");  Console.WriteLine("\n");  }  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 7) WACP to declare a jagged array and print values |
| CODE |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : WACP to declare a jagged array and print values  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13project7  {  internal class Program  {  static void Main(string[] args)  {    int[][] matrix1 = new int[5][];  Console.Write("\n\nPrinting a jagged array\n\n\n");  //giving user inputs for jagged array  matrix1[0] = new int[] { 1, 2, 3 };  matrix1[1] = new int[] { 4, 5, 6, 7, 5};  matrix1[2] = new int[] { 7, 8, 9 };  matrix1[3] = new int[] { 11, 21, 1, 3, 53, 23, 4 };  matrix1[4] = new int[] { 2, 4, 12, 34, 22, 3 };  //printing the jagged array  for (int i = 0; i < 5; i++)  {  for (int j = 0; j < matrix1[i].Length; j++)  Console.Write(matrix1[i][j] + "\t");  Console.WriteLine("\n");  }  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 8) WACP to illustrate usage of Recursion. |
| CODE |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : WACP to illustrate usage of Recursion.  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13project8  {  internal class Program  {  static void Main(string[] args)  {  int number,factorial;  Console.Write("\n\nFinding factorial for a user given input");  Console.Write("\n\n\nEnter the number to find factorial : ");  number = Convert.ToInt32(Console.ReadLine());  factorial = FindFactorial(number);  Console.Write($"\nFactorial of {number} is : {factorial}");  Console.ReadLine();  }  /// <summary>  /// Finding factorial for user given input  /// </summary>  /// <param name="n"></param>  /// <returns>  /// int  /// </returns>  static int FindFactorial(int n)  {  int factorial = 1;  if (n == 0)  return 1;  else  factorial = n \* FindFactorial(n - 1);  return factorial;  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 9) WACP to illustrate usage of Stack<> |
| CODE |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : WACP to illustrate usage of Stack<>  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project9  {  internal class Program  {  static void Main(string[] args)  {  Stack<int> numbers =new Stack<int>();  numbers.Push(10);  numbers.Push(20);  numbers.Push(34);  numbers.Push(24);  numbers.Push(46);  numbers.Push(58);  Console.Write("\n\nAdding integers to the stack and removing them based on last in-first out algorithm");  Console.Write("\n\n\n\nLast number added to the stack is : {0}", numbers.Peek());  Console.Write("\n\n\n\nTotal count of numbers in the stack is : {0}", numbers.Count());    Console.Write("\n\n\n\nNumbers of the stack by displaying recently added ones first\n");  for(int i = 1; i <= 6; i++)  Console.Write("\n{0}",numbers.Pop());  Console.Write("\n\n\n\nUpdated total count of numbers in the stack is : {0}", numbers.Count());  Console.ReadLine();  }  }  } |
| OUTPUT |
|  |

|  |
| --- |
| 10) WACP to illustrate usage of Queue<> |
| CODE |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Author : G V S S SRI LASYA  Purpose : WACP to illustrate usage of Queue<>  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  namespace Day13Project10  {  internal class Program  {  static void Main(string[] args)  {  Queue<int> numbers = new Queue<int>();  numbers.Enqueue(10);  numbers.Enqueue(20);  numbers.Enqueue(30);  numbers.Enqueue(40);  Console.Write("\n\nAdding integers to the queue and removing them using first in-first out algorithm");  Console.Write("\n\n\n\nTotal count of elements of queue : {0}\n\n\n",numbers.Count);  Console.Write("\nFirst number added to the queue : {0}\n\n\n\n", numbers.Peek());      Console.Write("\nValues in order of addition to the queue\n");  for(int i = 1; i <= 4; i++)  Console.Write("\n" + numbers.Dequeue());    Console.Write("\n\n\n\n\nUpdated total count of elements of queue : {0}", numbers.Count);    Console.ReadLine();  }  }  } |
| OUTPUT |
|  |