|  |
| --- |
| **DAY 13 Assignment**  **BY**  **Nanam Vaishnavi**  **09- Feb - 2022** |

|  |
| --- |
| **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 : Nanam Vaishnavi  // Purpose : 2D Array using nested for loop  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13project1  {  internal class Program  {  static void Main(string[] args)  {    int[,] data = new int[2, 2];  {  data[0, 0] = 12;  data[0, 1] = 26;  data[1, 0] = 32;  data[1, 1] = 45;  };    for (int i=0;i<2;i++)  {  for (int j=0;j<2;j++)  {  Console.Write(data[i, j] + " ");    }  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 : Nanam Vaishnavi  // Purpose : initializing values in declaration line and printing values  // /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project2  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[3, 2] { { 5, 6 },{ 12, 24 }, { 13, 25 } };  for(int i =0;i<3;i++)  {  for(int j=0;j<2;j++)  {  Console.Write(data[i, j] + " ");  }  Console.WriteLine("\n");  }  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;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Trace of the matrix  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project3  {  internal class Program  {  static void Main(string[] args)  {    int[,] data = new int[,] { { 1, 5, 9 }, { 2, 6, 10 }, { 3, 7, 11 } };  int sum = 0;  for (int i = 0;i<3;i++)  {  for (int j = 0; j < 3; j++)  {  if (i == j)  sum = sum + data[i, j];  }  Console.Write("\n");  }  Console.WriteLine(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;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Read values from and print the matrix  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project4  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[2,2];  // Read values from user  for(int i =0;i<2;i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine($"Enter array values at ({i},{j})");  data[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  Console.WriteLine("\n");  // Print Values  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(data[i,j]+ " ");  }  Console.WriteLine("\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 Day13Project5  {  internal class Program  {  static void Main(string[] args)  {    int[,] a1 = new int[2, 2];    //Read values for arr1 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine($"Enter First Matrix ({i},{j}): ");  //Console.WriteLine(a1[i, j]);  a1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  int[,] a2 = new int[2, 2];    //Read values for arr1 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine($" Enter Second Matrix({i},{j}): ");  //Console.WriteLine(a2[i, j]);  a2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  // adding two matrixes  Console.WriteLine("Addition of Two Matrices");  int[,] a3 = new int[2, 2];  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  a3[i, j] = a1[i, j] + a2[i, j];  }    }  // printing matrix  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(a3[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;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Product of two Matrices.  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project6  {  internal class Program  {  static void Main(string[] args)  {  int[,] p1 = new int[2, 2];  //Read values for p1 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine($"Enter First Matrix ({i},{j}): ");    p1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  int[,] p2 = new int[2, 2];  //Read values for p2 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine($" Enter Second Matrix({i},{j}): ");    p2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  // Product of two matrixes  Console.WriteLine("Product of Two Matrices");  int[,] p3 = new int[2, 2];  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  p3[i, j] = p1[i, j] \* p2[i, j];  }  }  // printing matrix  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(p3[i, j] + " ");  }  Console.Write("\n \n");  }  Console.ReadLine();  }  }  } |
| **OUTPUT** |
|  |

|  |
| --- |
| **7. What is a jagged Array. What is the benefit of jagged Array.** |
| **Ans :** |
| * A Jagged array is an array of arrays such that member arrays can be of different sizes. * Each of the elements is a single – dimensional array of integers.   **Benefits of Jagged Array:**   * It makes things easy where there is a need to store data in a multidimensional way using the same variable name. * No need to mention column size. |

|  |
| --- |
| **8. 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 : Nanam Vaishnavi  // Purpose : Declare Jagged Array and print values  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13project7  {  internal class Program  {  static void Main(string[] args)  {  char[][] names = new char[3][];  names[0] = new char[] { 'A', 'b', 'h', 'i' };  names[1] = new char[] { 'R', 'a', 't', 'h', 'i', 'k', 'a' };  names[2] = new char[] { 'K', 'a', 'r', 't', 'h', 'i', 'k' };  for(int i=0;i<3;i++)  {  for(int j =0;j<names[i].Length;j++)  {  Console.Write(names[i][j]);  }  Console.Write("\n");  }  Console.ReadLine();  }  }  } |
| **OUTPUT** |
|  |

|  |
| --- |
| **9. What is Recursion ?** |
| Ans : Recursion is defined as function calling itself until a specified condition is satisfied. |

|  |
| --- |
| **10. Write a C# program to illustrate usage of Recursion. What are the benefits of recursion** |
| **Benefits of Recursion:**   1. It adds clarity and reduces the time needed to write and debug. 2. Reduces time complexity. |
| **CODE** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Prime numbers in range using recursion  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Recursion  {  internal class Program  {  static void Main(string[] args)  {  int i1, i2;  Console.WriteLine("Enter first number");  i1 = Convert.ToInt32(Console.ReadLine());  Console.WriteLine("Enter second number");  i2 = Convert.ToInt32(Console.ReadLine());  for (int i = i1; i <= i2; i++)  {  if (isPrimenumber(i))  Console.WriteLine("{0}", i);  }  Console.ReadLine();  }  //Logic and returning Output  public static Boolean isPrimenumber(int input)  {  int i;  for (i = 2; i < input; i++)  {  if (input % i == 0)  break;  }  if (i == input)  return true;  else  return false;  Console.ReadLine();  }    }  } |
| **OUTPUT:** |
|  |

|  |
| --- |
| **11. Write a C# Program to illustrate usage of Stack<>. Write couple of points about Stack** |
| **Stack :**   * A stack is an abstract data type which holds an ordered, sequence of items. * Stack uses **Last In First Out (LIFO)** structure. |
| **CODE** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Usage of Stack  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project9  {  internal class Program  {  static void Main(string[] args)  {  Stack <int> data = new Stack<int>();  {  data.Push(26);  data.Push(20);  data.Push(37);  Console.WriteLine(data.Count);  Console.WriteLine(data.Pop());  Console.WriteLine(data.Count);  Console.WriteLine(data.Pop());  Console.ReadLine();  }  }  }  } |
| **OUTPUT** |
|  |

|  |
| --- |
| **12) WACP to illustrate usage of Queue<>. Write couple of points about Queue.** |
| **Queue :**   * Queue is used when things don’t have to be processed immediately. * It follows **First In First Out (FIFO)** order. |
| **CODE** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  // Author : Nanam Vaishnavi  // Purpose : Usage of Queue  // \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project10  {  internal class Program  {  static void Main(string[] args)  {  Queue <int> data = new Queue<int>();  data.Enqueue(56);  data.Enqueue(45);  data.Enqueue(50);  Console.WriteLine(data.Count);  Console.WriteLine(data.Dequeue());    Console.ReadLine();  }  }  } |
| **OUTPUT** |
|  |