|  |
| --- |
| DAY13 ASSIGNMENT  BY  PALURU MOUNIKA  09-02-2022 |

|  |
| --- |
| **1.Declare a 2-d array of size [2,2] and initialize using indexes print the values using nested forloop.** |
| **Code**: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose:2-d array of size[2,2] print values using nested forloop  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[2, 2] { { 4, 5 }, { 6, 7 } };  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] initialize in the same line while declaring and print values using nested forloop** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author: paluru mounika  //purpose:intializing the value in same linle using nested forloop print arrays  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project2  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int [3,2]{ { 23, 34 }, { 42, 25 }, { 46,75} };    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 2-d array of size (3,3) and print trace of array** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //pupose:intiliaze the array print trace of array  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13\_project3  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[3, 3] { { 34, 23, 45 }, { 44, 87, 94 }, { 34, 23, 56 } };  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.WriteLine("\n");  }  Console.WriteLine(sum);  Console.ReadLine();  }    }  } |
| **Output:** |
|  |

|  |
| --- |
| **4.declare 2-d array (2,2) read value from user and print the array** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose:Read values from user and print the values  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project4  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = 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 array value at({i},{j})");  data[i,j]=Convert.ToInt32(Console.ReadLine());  }  }  //print the 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 array of size(2,2) and read the values from user and print the sum of two matrices.** |
| **Code:** |
| sing System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose: sum of two matrices  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project5  {  internal class Program  {  static void Main(string[] args)  {  int[,] a1 = new int[2, 2];  Console.WriteLine("Enter First Matrix:");  //Read values for arr1 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {    a1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  int[,] a2 = new int[2, 2];  Console.WriteLine(" Enter Second Matrix:");  //Read values for arr1 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; 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 array of size (2,2) and read the value 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:paluru mounika  //purpose: sum of two matrices  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Project5  {  internal class Program  {  static void Main(string[] args)  {  int[,] a1 = new int[2, 2];  Console.WriteLine("Enter First Matrix:");  //Read values for arr1 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  a1[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  int[,] a2 = new int[2, 2];  Console.WriteLine(" Enter Second Matrix:");  //Read values for arr1 from user  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  a2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }  // adding two matrixes  Console.WriteLine("product 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**: |
|  |

|  |
| --- |
| **7. What is jagged array and benefit of jagged array?** |
| **-** A jagged array is an array of an array in which length of each array idex can differ.  **Benefits**:  -It makes thinks easy where there is a need to store data in a multidimensional way using the same variable name.  -It helps in memory management which makes the program to be excuted very smoothly and fast as well. |

|  |
| --- |
| **8.Write a c# program 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:paluru mounika  //purpose:declare jagged array print the values  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project8  {  internal class Program  {  static void Main(string[] args)  {  int[][] values = new int[3][];  values[0] = new int[] {2,4,6 };  values[1] = new int[] { 4,2,3,9};  values[2] = new int[] {7,9,4,2,7 };  for (int i = 0; i < 3; i++)  {  for (int j = 0; j < values[i].Length; j++)  {  Console.WriteLine(values[i][j] + " ");  }  Console.WriteLine("\n");  }  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **9.What recursion?** |
| A function calling itself repeatedly until a specified condition satisfied. |

|  |
| --- |
| 10.write a c# program to illustrate usage of recursion what are the benidfits of recursion? |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose:factorial of a number using recursion  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace factorial\_using\_recursion  {  internal class Program  {  static int Factorial(int n)  {  int fact = 1;  for (int i = 1; i <= n; i++)  fact = fact \* i;  return fact;  }  static void Main(string[] args)  {  Console.WriteLine("enter any number");  int input = Convert.ToInt32(Console.ReadLine());  int fact = Factorial(input);  Console.WriteLine("{0} factorial {1}", input, fact);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **Benefits of recursion:**  1.the code may be easier to write.  2.reduce unnecessary calling ofv function.  3.recursion reduce the length of the code.  4.extreamly usefull when applying the same solution. |

|  |
| --- |
| **11.write a c# program to illustrate usage of stack<> write couple of points about stack.** |
| 1.stack fallows” last in first out” alogorith  2.pop will remove the recently added element.  3.peak will not remove the element. It shows the peak number |
| **Code:** |
| using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose:program to illustrate the usage of stack<>  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project11  {  internal class Program  {  static void Main(string[] args)  {  Stack<int> data = new Stack<int>();  data.Push(13);  data.Push(21);  data.Push(17);  Console.WriteLine(data.Count);  Console.WriteLine(data.Pop());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose:program to illustrate the usage of stack<>  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project11  {  internal class Program  {  static void Main(string[] args)  {  Stack<int> data = new Stack<int>();  data.Push(13);  data.Push(21);  data.Push(17);  Console.WriteLine(data.Count);  Console.WriteLine(data.Peek());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  }  9 |
| **Output:** |
|  |

|  |
| --- |
| **12.write a c# program to illustrate usage of queue<>.write a couple points about queue**. |
| 1.queue fallows the” first in first out” algorithm.  2.when removing the element is dequeue.  3.when adding the element is enqueue. |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose: program to illustrate usage of queue  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project12  {  internal class Program  {  static void Main(string[] args)  {  Queue<int> data = new Queue<int>();  data.Enqueue(21);  data.Enqueue(25);  data.Enqueue(13);  Console.WriteLine(data.Count);  Console.WriteLine(data.Dequeue());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:paluru mounika  //purpose: program to illustrate usage of queue  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project12  {  internal class Program  {  static void Main(string[] args)  {  Queue<int> data = new Queue<int>();  data.Enqueue(21);  data.Enqueue(25);  data.Enqueue(13);  Console.WriteLine(data.Count);  Console.WriteLine(data.Peek());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |