|  |
| --- |
| DAY13 ASSIGNMENT  BY  J Siva Naga Prasanna  09-02-2022 |

|  |
| --- |
| **1.Declare a 2-d array of size [2,2] and initialize using indexes 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: j siva naga prasanna  //purpose:2-d array of size[2,2] print values using nested for loop  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro1  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[2, 2];  data[0, 0] = 5;  data[0, 1] = 6;  data[1, 0] = 45;  data[1, 1] = 12  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 2-d array of size [3,2] initialize in the same line while declaring and print values using nested for loop** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author: j siva naga prasanna  //purpose: Initializing the values in the same line using nested for loop print arrays  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro2  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int [3,2]{ { 36, 37 }, { 41, 47 }, { 51,54} };    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:j siva naga prasanna  //purpose: print trace of array  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro3  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[3, 3] { { 11, 12, 13 }, { 21, 22, 23 }, { 31, 32, 33 } };  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 of size (2,2) 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:j siva naga prasanna  //purpose:Read values from user and print the values  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro4  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[2, 2];    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());  }  }    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:j siva naga prasanna  //purpose: sum of two matrices  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Pro5  {  internal class Program  {  static void Main(string[] args)  {  int[,] a1 = new int[2, 2];  Console.WriteLine("Enter First Matrix:");    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:");    for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {    a2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }    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];  }  }    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:j siva naga prasanna  //purpose: print the product of two matrices  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace Day13Pro6  {  internal class Program  {  static void Main(string[] args)  {  int[,] a1 = new int[2, 2];  Console.WriteLine("Enter First Matrix:");    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:");    for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  a2[i, j] = Convert.ToInt32(Console.ReadLine());  }  }    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];  }  }    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 array which length of each array index can different to each other.some times jagged array is called as “array of arrays”  **Benefits**:  1.It makes easy where we store data in a multidimensional way using same variable name.  2.It helps in memory management which makes the program to be exicuted ( soft and fast) . |

|  |
| --- |
| **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:j siva naga prasanna  //purpose:declare jagged array print the values  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro8  {  internal class Program  {  static void Main(string[] args)  {  int[][] values = new int[3][];  values[0] = new int[] {1,2,3 };  values[1] = new int[] { 4,5,6,7};  values[2] = new int[] {8,9,10,11,12 };  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 until a given specified condition is satisfied. |

|  |
| --- |
| 10.write a c# program to illustrate usage of recursion |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:j siva naga prasanna  //purpose:factorial of a given number using recursion  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace fact  {  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:** |
|  |

**10.befits of recursion**

**1.Reduce unnecessary calling function**

**2.it is very useful in solving the data structure problems**

**3.Recursion reduces the length of code**

|  |
| --- |
| **11. write couple of points about stack.** |
| 1.stack allows ” last in first out” algorithm  2.pop will remove the recently added element.  3.peak will not remove the element. It displays the peak number |
| **Code:write a c# program to iilustrate usage of stack** |
| using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:j siva naga prasanna  //purpose:usage of stack<>  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro10  {  internal class Program  {  static void Main(string[] args)  {  Stack<int> data = new Stack<int>();  data.Push(25);  data.Push(36);  data.Push(47);  Console.WriteLine(data.Count);  Console.WriteLine(data.Pop());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **Code:using peek** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author: j siva naga prasanna  //purpose: usage of stack<> using peek  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro11  {  internal class Program  {  static void Main(string[] args)  {  Stack<int> data = new Stack<int>();  data.Push(11);  data.Push(22);  data.Push(33);  Console.WriteLine(data.Count);  Console.WriteLine(data.Peek());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **12. write a couple of points about queue**. |
| 1.queue allows the” first in first out” algorithm.  2. Dequeue is used to remove elements  3. Enqueue is used to add the elements |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:j siva naga prasanna  //purpose: program to illustrate usage of queue  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13pro12  {  internal class Program  {  static void Main(string[] args)  {  Queue<int> data = new Queue<int>();  data.Enqueue(65);  data.Enqueue(76);  data.Enqueue(87);  Console.WriteLine(data.Count);  Console.WriteLine(data.Dequeue());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |

|  |
| --- |
| **Code:using peek** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  //author:j siva naga prasanna  //purpose: program to illustrate usage of queue using peek  //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  namespace day13project12  {  internal class Program  {  static void Main(string[] args)  {  Queue<int> data = new Queue<int>();  data.Enqueue(105);  data.Enqueue(116);  data.Enqueue(127);  Console.WriteLine(data.Count);  Console.WriteLine(data.Peek());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| **Output:** |
|  |