Day 13 Assignment

By

B.P.N.V.S.Sudheer

09-02-22

|  |
| --- |
| **1.Declare a 2 dimentional array of size (2,2) and initialize using indexes and print the values usingnested for loop** |
| **Code:** |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace Day13\_Project1  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[,]{ { 50, 60 },{ 70,80}};  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(data[i,j]+"");  }  Console.Write("\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 valuesusing nested for loop** |
| Code: |

|  |
| --- |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace Day13Project2  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[,] { { 50, 60 }, { 60, 70 }, { 80, 90 } };  for (int i = 0; i < 3; i++)  {  for (int j = 0; j < 2; j++)  {  Console.Write(data[i, j]+"");  }  Console.Write("\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;    namespace \_13thdayproject3  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[,] { { 1, 2, 3 }, { 4, 5, 4 }, { 6, 4, 8 } };  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;    namespace \_13thdayproject4  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[2, 2];  //Read 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.Write("\n");    }  Console.ReadLine();  }  }  } |
| Output: |
|  |
| 5. **Declare TWO 2-D arrays of size (2,2) and read values fromuser 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 \_13thday\_project5  {  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 1 value");  data[i, j] = Convert.ToInt32(Console.ReadLine());  }  }    for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++) { }  }  int[,] data2 = new int[2, 2];  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine("enter array 2 value");  data2[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] + data2[i, j] + " ");  }  Console.Write("\n");  }      Console.ReadLine();    }  }  } |

|  |
| --- |
| Output: |
|  |
| 6. Declare TWO 2-D arrays of size (2,2) and read values fromuser 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 \_13thdayproject6  {  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 1 value");  data[i, j] = Convert.ToInt32(Console.ReadLine());  }  }    for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++) { }  }  int[,] data2 = new int[2, 2];  for (int i = 0; i < 2; i++)  {  for (int j = 0; j < 2; j++)  {  Console.WriteLine("enter array 2 value");  data2[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] \* data2[i, j] + " ");  }  Console.Write("\n");  }  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| **7. What is a jagged array What is the benefit of jagged array** |
| * Jagged arrays may be defined as multidimensional array that is capable of storing various values under the name of a single variable |
| * The jagged arrays are very helpful in enhancing the capabilities of the applications by letting them process several values stored in a specific manner |
| 8**.What is recursion** |
| * A recursive method is a method which call itself again and again on basis of few statements which need to be true similarly when a function call itself again and again it is known as a recursive function |
| 9. **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;    namespace \_13thdayproject7  {  internal class Program  {  static void Main(string[] args)  {  char[][] names = new char[3][];  names[0] = new char[] { 'p', 'a', 'v', 'a', 'n' };  names[1] = new char[] { 'v', 'e', 'n', 'k', 'a','t','a' };  names[2] = new char[] { 's', 'a', 'i','s','u','d','h','e','e','r'};  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: |
|  |
| **10. WACP to illustrate usage of Recursion. What are the benefits of recursion** |
| Code: |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace fact  {  class Factorial  {  public int display(int n)  {  if (n == 1)  return 1;  else  return n \* display(n - 1);  }  internal class Program  {  static void Main(string[] args)  {  int value = 6;    int ret;  Factorial fact = new Factorial();  ret = fact.display(value);  Console.WriteLine("Value is : {0}", ret);  Console.ReadLine();    }  }  }  } |
| Output: |
|  |

|  |
| --- |
| Benefits of recursion |
| * Reduce time complexity |
| * Performs better in solving problems based on tree structure |
| **11. WACP to illustrate usage of Stack<> Write couple of points aboutStack** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_13thdayproject9  {  internal class Program  {  static void Main(string[] args)  {  Stack<int> data = new Stack<int>();  data.Push(1);  data.Push(2);  data.Push(3);  Console.WriteLine(data.Count);  Console.WriteLine(data.Pop());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| Output: |
|  |

|  |
| --- |
| About Stack |
| * In computer science a stack is an abstract data type that serves as a collection of elements with two main principle operations push which adds an element to the collections and pop which removes the most recently added elements that was not yet removed |
| **11. WACP to illustrate usage of Queue<> Write couple of points about Queue** |
| Code : |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;    namespace \_13thdayproject10  {  internal class Program  {  static void Main(string[] args)  {  Queue<int> data = new Queue<int>();  data.Enqueue(1);  data.Enqueue(2);  data.Enqueue(3);  Console.WriteLine(data.Count);  Console.WriteLine(data.Dequeue());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| Output: |
|  |
| About Queue |

|  |
| --- |
| * In computer science a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence |