Day 13 Assignment

By K . SANJAY

|  |
| --- |
| 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 : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose : Two dimensional array and print using nested for loop \*\*\*\*\*\*\*\*/  namespace Day\_13\_Assignment\_9th\_feb\_2022  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[2,2];  data[0,0] = 2;  data[0, 1] = 5;  data[1, 0] = 7;  data[1, 1] = 10;  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 values  using nested for loop |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\*\* Author : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose : Two dimensional array initialize th values in same line using nested for loop \*\*\*\*\*\*\*\*/  namespace Day\_13\_project\_2\_new  {  internal class Program  {  static void Main(string[] args)  {  int[,] data = new int[,] { { 5, 9 }, { 6, 8 }, { 8,24} };    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;  /\*\*\*\*\*\*\*\*\*\* Author : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose :Trace of the array \*\*\*\*\*\*\*\*/  namespace Day\_13\_project\_2\_new  {  internal class Program  {  static void Main(string[] args)  {  int sum = 0;  int[,] data = new int[,] { { 5, 9,4 }, { 6, 8,1 }, { 8,24,6} };    for (int i = 0; i < 3; i++)  {  for (int j = 0; j < 3; j++)  {  if(i==j)  sum = sum + data[i,j];  }    }  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 : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose :Using two dimensional array and read values from the user and print \*\*\*\*\*\*\*\*/  namespace Day\_13\_project\_4  {  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:");  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.Write("\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;  /\*\*\*\*\*\*\*\*\*\* Author : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose :Sum of two matrices \*\*\*\*\*\*\*\*/  namespace Day\_13\_project\_5  {  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 first array:");  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 second array:");  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 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 : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose :Product of two matrices \*\*\*\*\*\*\*\*/  namespace Day\_13\_project\_6  {  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 first array:");  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 second array:");  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 array is a two dimesional array. * Which will have different sizes for different rows. * The way you declared jagged array is different when compare to normal arrays.   Benfit of Jagged array:   * You will save some memory if the element size is different. |

|  |
| --- |
| 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 : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose :Declaring jagged array and printing values \*\*\*\*\*\*\*\*/  namespace Day\_13\_project\_7  {  internal class Program  {  static void Main(string[] args)  {  char[][] names = new char[3][];  names[0] = new char[] { 'j','o','h','n'};  names[1] = new char[] { 'j', 'o', 'h', 'n','s','a','n','j','a','y' };  names[2] = new char[] { 'j', 'o', 'h', 'n', 's', 'a', 'n', 'j', 'a', 'y','s','o','n' };  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. |
| * A function calling itself repeatedly until a spcified condition is satisfy. |

|  |
| --- |
| 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 Day\_13\_Project\_8  {  internal class Program  {      static void Main(string[] args)  {  Console.WriteLine("Enter a number:");  int num = int.Parse(Console.ReadLine());  int factorial = 1;  for(int i=1;i<num;i++)  {  factorial = factorial \* i;  }  Console.WriteLine($"Factorial of a given number is :{factorial}");  Console.ReadLine();  }  }  } |
| Output |
|  |
| Benefits of Recursion:   * The Number of variables go down. * Recursion adds clarity and reduces the time needed to write and debug the code. * Recursion is better at tree traversal. |

|  |
| --- |
| 11. WACP to illustrate usage of Stack<>  Write couple of points about Stack. |
| Code |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  /\*\*\*\*\*\*\*\*\*\* Author : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose :Using Stack \*\*\*\*\*\*\*\*/  namespace Day\_13\_project\_9  {  internal class Program  {  static void Main(string[] args)  {  Stack<int> data = new Stack<int>();  data.Push(25);  data.Push(23);  data.Push(30);  Console.WriteLine(data.Count);  Console.WriteLine(data.Pop());  Console.WriteLine(data.Count);  Console.WriteLine();  Console.ReadLine();  }  }  } |
| Output |
|  |
| * It represents last-in, first out * It is used when you need a last-in,first-out access to items. * The process of adding an element to the stack is called a push opration. |

|  |
| --- |
| 12. 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;  /\*\*\*\*\*\*\*\*\*\* Author : K. Sanjay \*\*\*\*\*\*\*\*\*\*\*/  /\*\*\*\*\*\*\*\*\*\* Purpose :Using Queue \*\*\*\*\*\*\*\*/  namespace Day\_13\_\_project\_10  {  internal class Program  {  static void Main(string[] args)  { Queue<int> data = new Queue<int>();  data.Enqueue(21);  data.Enqueue(96);  data.Enqueue(74);  Console.WriteLine(data.Count);  Console.WriteLine(data.Dequeue());  Console.WriteLine(data.Count);  Console.ReadLine();  }  }  } |
| Output |
|  |
| * It follows first in first out elements * Dequeue returns an item from th beginning of the queue and removes it from th queue. * Peek returns an first item from the queue without removing it. |