**Assignment 8**

**Objective:**

Write a programwhich take 3 string as input and sum the result of three string

**Code:**

using System;

usingSystem.Collections.Generic;

usingSystem.Linq;

usingSystem.Text;

usingSystem.Threading.Tasks;

namespace ToCS\_Assignment\_9

{

classProgram

{

//This Function make displays the maze

publicstaticvoiddisplay\_maze(int[,] arr)

{

Console.WriteLine("--Maze--");

for (int r = 0; r < 16; r++)

{

Console.Write(arr[r, 4] + " ");

if((r+1)%4==0 && r>2)

Console.WriteLine();

}

Console.WriteLine();

}

staticvoid Main(string[] args)

{

inti = 0, input\_count = 0, r = 0, next\_state, sum;

charchr;

//This stores the result of each of three strings input

int[] result = newint[3];

//Initializing Matrix of order 16 by 5

int[,] transition\_table = newint[16, 5];

//Filling the matrix according to given problem (Maze)

{

/\* Maze

\* | 5 | 2 | 3 | 4 |

\* | 3 | 0 | 1 | 2 |

\* | 4 | 5 | 1 | 7 |

\* | 6 | 0 | 2 | 4 |

\*/

//0 = Move Left, 1 = Move Right, 2 = Move Top, 3 = Move Down, 4 = Output of the current state

// This is first row of Transition Table

transition\_table[0, 0] = 0; // Move to state 0 when input is 0

transition\_table[0, 1] = 1; // Move to state 1 when input is 1

transition\_table[0, 2] = 0; // Move to state 0 when input is 2

transition\_table[0, 3] = 4; // Move to state 4 when input is 3

transition\_table[0, 4] = 5; // This is output of first state

//First Column (0)

transition\_table[++i, 0] = 0;

transition\_table[++i, 0] = 1;

transition\_table[++i, 0] = 2;

transition\_table[++i, 0] = 4;

transition\_table[++i, 0] = 4;

transition\_table[++i, 0] = 5;

transition\_table[++i, 0] = 6;

transition\_table[++i, 0] = 8;

transition\_table[++i, 0] = 8;

transition\_table[++i, 0] = 9;

transition\_table[++i, 0] = 10;

transition\_table[++i, 0] = 12;

transition\_table[++i, 0] = 12;

transition\_table[++i, 0] = 13;

transition\_table[++i, 0] = 14;

i = 0;

//Second Column (1)

transition\_table[++i, 1] = 2;

transition\_table[++i, 1] = 3;

transition\_table[++i, 1] = 3;

transition\_table[++i, 1] = 5;

transition\_table[++i, 1] = 6;

transition\_table[++i, 1] = 7;

transition\_table[++i, 1] = 7;

transition\_table[++i, 1] = 9;

transition\_table[++i, 1] = 10;

transition\_table[++i, 1] = 11;

transition\_table[++i, 1] = 11;

transition\_table[++i, 1] = 13;

transition\_table[++i, 1] = 14;

transition\_table[++i, 1] = 15;

transition\_table[++i, 1] = 15;

i = 0;

//Third Column (2)

transition\_table[++i, 2] = 1;

transition\_table[++i, 2] = 2;

transition\_table[++i, 2] = 3;

transition\_table[++i, 2] = 0;

transition\_table[++i, 2] = 1;

transition\_table[++i, 2] = 2;

transition\_table[++i, 2] = 3;

transition\_table[++i, 2] = 4;

transition\_table[++i, 2] = 5;

transition\_table[++i, 2] = 6;

transition\_table[++i, 2] = 7;

transition\_table[++i, 2] = 8;

transition\_table[++i, 2] = 9;

transition\_table[++i, 2] = 10;

transition\_table[++i, 2] = 11;

i = 0;

//Fourth Column (3)

transition\_table[++i, 3] = 5;

transition\_table[++i, 3] = 6;

transition\_table[++i, 3] = 7;

transition\_table[++i, 3] = 8;

transition\_table[++i, 3] = 9;

transition\_table[++i, 3] = 10;

transition\_table[++i, 3] = 11;

transition\_table[++i, 3] = 12;

transition\_table[++i, 3] = 13;

transition\_table[++i, 3] = 14;

transition\_table[++i, 3] = 15;

transition\_table[++i, 3] = 12;

transition\_table[++i, 3] = 13;

transition\_table[++i, 3] = 14;

transition\_table[++i, 3] = 15;

i = 0;

//Fifth Column (4) OR Output column

transition\_table[++i, 4] = 2;

transition\_table[++i, 4] = 3;

transition\_table[++i, 4] = 4;

transition\_table[++i, 4] = 3;

transition\_table[++i, 4] = 0;

transition\_table[++i, 4] = 1;

transition\_table[++i, 4] = 2;

transition\_table[++i, 4] = 4;

transition\_table[++i, 4] = 5;

transition\_table[++i, 4] = 1;

transition\_table[++i, 4] = 7;

transition\_table[++i, 4] = 6;

transition\_table[++i, 4] = 0;

transition\_table[++i, 4] = 2;

transition\_table[++i, 4] = 4;

}

Console.WriteLine("INSTRUCTIONS\n" +

"0 = Move Left, 1 = Move Right, 2 = Move Top, 3 = Move Down\n");

display\_maze(transition\_table);

//This is main loop which take three input string and perform transition

while (input\_count< 3)

{

Console.Write("Enter String " + (input\_count + 1) + " : ");

string str = Console.ReadLine(); //Take User Input

i = 0; r = 0;

//This Loop traverse the Nodes/Transition Table and reaches the end of the string

while (i<str.Length)

{

chr = str[i]; //chr contains the i'th character of input string

//next\_state have the [r , (i'thcharater of string)] which points to next state in transition table

next\_state = transition\_table[r, (int)char.GetNumericValue(chr)];

r = next\_state; //This next state will be the new row (r)

i++;

}

//Output of the final state is saved in result array

result[input\_count] = transition\_table[r, 4];

Console.Write(result[input\_count]+"\n");

input\_count++;

}

sum = result[0] + result[1] + result[2];

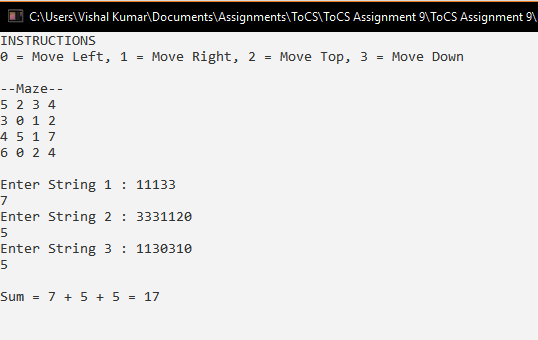
Console.WriteLine("\nSum = " + result[0] + " + " + result[1] + " + " + result[2] + " = "+sum);

Console.ReadKey();

}

}

}

****