

Experiment: 1.3

Student Name: Sandeep Kumar

Branch: BE-CSE

Semester: 6th

Subject Name: Dotnet Lab

UID: 20BCS4885

Section/Group: 603/A

Date of Performance: 08/03/2023

Subject Code: 20CSP-381

Aim of the practical:

A rectangular integer array is given. Implement the "merge" sorting method (do not use the Array class method!) so that you can arrange the rows of the matrix

- in ascending (descending) order of the sums of the elements of the rows of the matrix;
- in ascending order (descending) of the maximum elements of the matrix rows;
- in ascending (descending order) of the minimum elements of the matrix rows.

Program:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApp1
{
    public static class Experiment3
    {
        static void sortByRow(int[,] m)
        {
            for (int i = 0; i < m.GetLength(0); i++)
            {
                for (int j = 0; j < m.GetLength(1); j++)
                {
                    for (int k = 0; k < m.GetLength(1) - j - 1; k++)
                    {
                        if (m[i, k] > m[i, k + 1])
                        {
                            int t = m[i, k];
                            m[i, k] = m[i, k + 1];
                            m[i, k + 1] = t;
                        }
                    }
                }
            }
        }
    }
}
```

```
static void reverseArray(int[,] arr)
{
    for (int i = 0; i < arr.GetLength(1); i++)
    {
        int start = 0;
        int end = arr.GetLength(0) - 1;

        while (start < end)
        {
            int temp = arr[i, start];
            arr[i, start] = arr[i, end];
            arr[i, end] = temp;

            start++;
            end--;
        }
    }
}

public static void transpose(int[,] mat, int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = i + 1; j < n; j++)
        {
            int temp = mat[i, j];
            mat[i, j] = mat[j, i];
            mat[j, i] = temp;
        }
    }
}

public static void sortMatRowAndColWise(int[,] mat,
                                         int n)
{
    sortByRow(mat);

    transpose(mat, n);

    sortByRow(mat);
    reverseArray(mat);

    transpose(mat, n);
}

public static void printMat(int[,] mat, int n)
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            Console.Write(mat[i, j]);
            Console.Write(" ");
        }
    }
}
```

```
        Console.WriteLine("\n");
    }
}

internal static void Main()
{
    int n = 3;

    int[,] mat
        = { { 3, 2, 1 }, { 9, 8, 7 }, { 6, 5, 4 } };

    Console.WriteLine("Original Matrix:\n");
    printMat(mat, n);

    sortMatRowAndColWise(mat, n);

    Console.WriteLine("\nMatrix After Sorting:\n");
    printMat(mat, n);
    Console.ReadLine();
}
}
```

Output:



C:\Users\ankul\source\repos\Conso

Original Matrix:

```
3 2 1
9 8 7
6 5 4
```

Matrix After Sorting:

```
7 8 9
4 5 6
1 2 3
```