Lab 4

Text

Description automatically generated

using System;

class Program

{

    public static void Display(int[] arr)

    {

            foreach (var item in arr) {

                if(item !=0)

                {

            Console.Write(item + ", ");

                }

        }

    }

  public static void Main()

  {

    int[] numRay = new int[10];

    int[] dupe = new int[10];

    Console.Write("Input 10 numbers in the array :\n");

    for(int i=0; i<10; i++)

    {

        Console.Write("element - {0} : ",i+1);

        numRay[i] = Convert.ToInt32(Console.ReadLine());

    }

    Display(numRay);

    Console.WriteLine("");

    for (int i = 0; i < numRay.Length; i++)

        {

            numRay[numRay[i] % numRay.Length]

                = numRay[numRay[i] % numRay.Length]

                + numRay.Length;

        }

        int j = 0;

        for (int i = 0; i < numRay.Length; i++)

        {

            if (numRay[i] >= numRay.Length \* 2)

            {

                dupe[j++] = i;

            }

        }

        Display(dupe);

  }

}

Lab 5

Text

Description automatically generated

using System;

class Program

{

    static public void Display(string[] arr){

             foreach (var item in arr) {

            Console.Write(item + ", ");

        }

    }

  public static void Main()

  {

    string[] arr = new string[10];

    Console.Write("Input 10 names in the array :\n");

    int q = 1;

    for(int i=0; i<10; i++)

    {

        Console.Write("name {0} : ",q++);

        arr[i] = Console.ReadLine();

    }

    Display(arr);

    int n = arr.Length;

    for (int j = 0; j < arr.Length / 2; j++)

    {

        string tmp = arr[j];

        arr[j] = arr[arr.Length - j - 1];

        arr[arr.Length - j - 1] = tmp;

    }

    Console.WriteLine(" ");

    Display(arr);

  }

}

Lab 6



using System;

class Program

{

  static void swap(int[] arr, int i, int j)

  {

    int temp = arr[i];

    arr[i] = arr[j];

    arr[j] = temp;

  }

  static int partition(int[] arr, int low, int high)

  {

    int pivot = arr[high];

    int i = (low - 1);

    for (int j = low; j <= high - 1; j++)

    {

      if (arr[j] < pivot && arr[j] != 0)

      {

        i++;

        swap(arr, i, j);

      }

    }

    swap(arr, i + 1, high);

    return (i + 1);

  }

  static void quickSort(int[] arr, int low, int high)

  {

    if (low < high)

    {

      int pi = partition(arr, low, high);

      quickSort(arr, low, pi - 1);

      quickSort(arr, pi + 1, high);

    }

  }

  static void printArray(int[] arr, int size)

  {

    for (int i = 0; i < size; i++)

      Console.Write(arr[i] + " ");

    Console.WriteLine();

  }

  public static void Main()

  {

    int[] arr = {1, 2, 0, 3, 4, 5, 0, 6, 7, 8, 0, 9};

    int n = arr.Length;

    quickSort(arr, 0, n - 1);

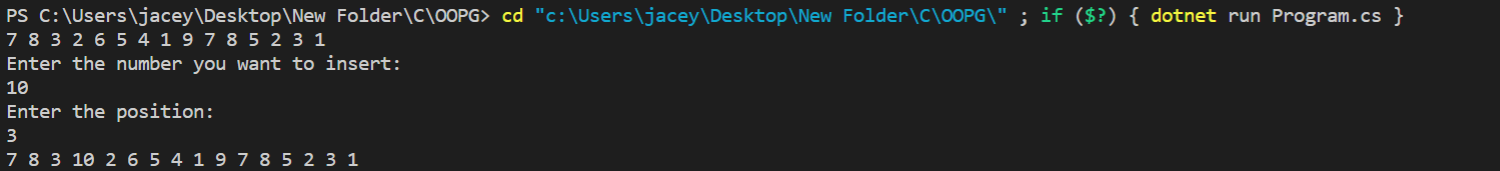
    Console.Write("Sorted array: ");

    printArray(arr, n);

  }

}

Lab 7



using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Lesson10bEx7

{

    class Program

    {

        static void Main(string[] args)

        {

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

            int[] modarray = new int[array.Count()+1];

            foreach (int i in array)

            {

                Console.Write(i + " ");

            }

            Console.WriteLine("\nEnter the number you want to insert:");

            int number = int.Parse(Console.ReadLine());

            Console.WriteLine("Enter the position:");

            int position = int.Parse(Console.ReadLine());

            for (int i = 0; i < array.Count()+1; i++)

            {

                if (i == position)

                {

                    modarray[i] = number;

                }

                else if (i != position &&  i < position)

                {

                    modarray[i] = array[i];

                }

                else if (i != position && i > position)

                {

                    modarray[i] = array[i-1];

                }

            }

            foreach (int j in modarray)

            {

                Console.Write(j + " ");

            }

        }

    }

}