

Nguyễn Minh Trí
MSSV: 20DH111773

```
using System;

namespace Tuan2
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("=====
=====");
            Console.Write("Nhap so ban muon tim: ");
            int n = int.Parse(Console.ReadLine());
            Console.Write("Ban muon random so lon nhat la bao nhieu?: ");
            int max = int.Parse(Console.ReadLine());
            Console.Write("Ban muon co bao nhieu phan tu trong mang?: ");
            int len = int.Parse(Console.ReadLine());
            MyIntArray a = new MyIntArray(len);
            a.RandomArray(max);
            a.OutputArray();
            if (a.FindContent(n) == -1)
            {
                Console.WriteLine("\nKhong tim thay gia tri can tim!");
            }
            else
            {
                Console.WriteLine($"Tim thay {n} o vi tri i = {a.FindContent
(n)} (Xuat phat i = 0)");
            }

            if(a.CheckArray() == "Increase"){
                Console.WriteLine("mang tang dan");
            }
            else if(a.CheckArray() == "Decrease")
            {
                Console.WriteLine("mang giam dan");
            }
            else if(a.CheckArray() == "Not_in_order")
            {
                Console.WriteLine("mang khong co thu tu");
            }
            Console.WriteLine($"So lan thuc hien so sanh tuan tu la: {a.CountL
inearSearchSteps(n)}");
            Array.Sort(a.Mang);
            Console.WriteLine($"So lan thuc hien so sanh nhi phan (sau khi duo
c sap xep) la: {a.CountBinarySearchSteps(n)}");
        }
    }
}
```

```

        Console.WriteLine("=====
=====");
    }
}
class MyIntArray
{
    private int[] array;
    public int[] Mang
    {
        get => this.array;
        set { this.array = value; }
    }
    public MyIntArray(int n = 8)
    {
        Mang = new int[n];
    }
    public int this[int i]
    {
        get => array[i];
        set => array[i] = value;
    }
    public void Input()
    {
        String[] tk = Console.ReadLine().Split();
        for(int i = 0; i < Mang.Length; i++)
        {
            Mang[i] = int.Parse(tk[i]);
        }
    }

    public void RandomArray(int max)           //phát sinh ngẫu nhiên mảng
    {
        for (int i = 0; i < Mang.Length; i++)
        {
            Random x = new Random();
            Mang[i] = x.Next(max);
        }
    }
    public void OutputArray()                  //Xuất mảng
    {
        Console.WriteLine("KET QUA MANG:");
        for (int i = 0; i < Mang.Length; i++)
        {
            Console.Write(Mang[i] + " ");
        }
    }
    public int FindContent(int x)              //Tìm kiếm theo phương pháp tu

```

ần tự

```

{
    for (int i = 0; i < Mang.Length; i++)
    {
        if (Mang[i] == x)
        {
            return i;
        }
    }
    return -1;
}

public int CountLinearSearchSteps(int number_being_find) //Đếm
số lần thực hiện tìm kiếm theo phương pháp tuần tự
{
    for (int i = 0; i < Mang.Length; i++)
    {
        if (Mang[i] == number_being_find)
        {
            return i + 1;
        }
    }
    return Mang.Length;
}

public int CountBinarySearchSteps(int number_being_find) //Đếm
số lần thực hiện tìm kiếm theo phương pháp nhị phân
{
    int left = 0;
    int right = Mang.Length - 1;
    int mid;
    int found = 0;
    while (left <= right)
    {
        mid = (left + right) / 2;
        found++;
        if (Mang[mid] == number_being_find)
        {
            break;
        }
        else if (number_being_find < Mang[mid])
        {
            right = mid - 1;
        }
        else if (number_being_find > Mang[mid])
        {
            left = mid + 1;
        }
    }
    return found;
}

```

```

    public string CheckArray()           //Kiểm tra xem mảng tăng dần, giảm
    dần hay không có thứ tự
    {
        int count_increase = 0;
        int count_decrease = 0;
        for(int i = 0; i< Mang.Length - 1; i++)
        {
            if(Mang[i+1]>=Mang[i])
            {
                count_increase++;
            }
            if(Mang[i+1]<=Mang[i])
            {
                count_decrease++;
            }
        }
        if(count_increase == Mang.Length - 1)
        {
            return "Increase";
        }
        else if(count_decrease == Mang.Length - 1)
        {
            return "Decrease";
        }
        return "Not_in_order";
    }
}

```