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($"\nTim thay {n} o vi tri i = {a.FindContent(n)} (Xuat phat i = 0)");

            }

            if(a.CheckArray() == "Increase"){

                Console.WriteLine("Mang tang dan nen ta co the tim kiem tuan tu hay tuyen tinh deu duoc");

            }

            else if(a.CheckArray() == "Decrease")

            {

                Console.WriteLine("Mang giam dan nen ta chi co the tim kiem tuan tu");

            }

            else if(a.CheckArray() == "Not\_in\_order")

            {

                Console.WriteLine("Mang khong co thu tu nen ta chi co the tim kiem tuan tu");

            }

            Console.WriteLine($"So lan thuc hien so sanh tuan tu la: {a.CountLinearSearchSteps(n)}");

            Array.Sort(a.Mang);

            Console.WriteLine($"So lan thuc hien so sanh nhi phan (sau khi duoc 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 tuần 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";

        }

    }

}