

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**



**MÔN LẬP TRÌNH TRỰC QUAN**  
**BÀI TẬP THỰC HÀNH 01**

GVHD: Nguyễn Ngọc Quý

Sinh viên thực hiện: Lưu Bích Ngân

ở Tp. Hồ Chí Minh, 09/2025 óó

## This image shows a full page of a handwriting practice worksheet. It consists of multiple sets of three horizontal dashed lines, providing a guide for letter height and placement. The lines are evenly spaced across the entire page, leaving ample room for writing practice. There is no text or other markings on the page.

**Người nhận xét**  
(Ký tên và ghi rõ họ tên)

## MỤC LỤC

<b>Bài 1. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên. Cài đặt hàm thực hiện các chức năng: .....</b>	<b>6</b>
<b>Bài 2: Viết chương trình nhập số nguyên dương n, tính tổng các số nguyên tố &lt; n và xuất kết quả ra màn hình.....</b>	<b>9</b>
<b>Bài 3: Viết chương trình nhập ngày tháng năm, cho biết ngày tháng năm đó có hợp lệ không? .....</b>	<b>11</b>
<b>Bài 4: Viết chương trình nhập vào tháng và năm. In ra số ngày của tháng đó.....</b>	<b>11</b>
<b>Bài 5: Viết chương trình nhập vào ngày tháng năm, cho biết thứ trong tuần .....</b>	<b>11</b>
<b>Bài 6. Viết chương trình cho phép tạo ma trận chứa các số nguyên ngẫu nhiên gồm n dòng, m cột. Cài đặt hàm thực hiện các chức năng: .....</b>	<b>14</b>

**DANH MỤC BẢNG**

Table 1 Nội dung code của chương trình bài 1 .....	6
Table 2 Testcase của chương trình bài 1 .....	8
Table 3 Nội dung code của chương trình bài 2 .....	9
Table 4 Testcase của chương trình bài 2.....	10
Table 5 Nội dung code của chương trình bài 3+4+5.....	11
Table 6 Testcase của chương trình bài 3+4+5.....	13
Table 7 Nội dung code của chương trình bài 6.....	15
Table 8 Testcase của chương trình bài 6.....	18

## DANH MỤC HÌNH ẢNH

Hình 1: Testcase bài 1.1 .....	8
Hình 2: Testcase bài 1.2 .....	8
Hình 3: Testcase bài 1.3 .....	9
Hình 4: Testcase bài 2.1 .....	10
Hình 5: Testcase bài 2.2 .....	10
Hình 6: Testcase bài 2.3 .....	10
Hình 7: Testcase bài 3+4+5.1 .....	13
Hình 8: Testcase bài 3+4+5.2 .....	13
Hình 9: Testcase bài 3+4+5.3 .....	14
Hình 10: Testcase bài 3+4+5.4 .....	14
Hình 11: Testcase bài 3+4+5.5 .....	14
Hình 12: Testcase bài 6.1 .....	18
Hình 13: Testcase bài 6.2 .....	19
Hình 14: Testcase bài 6.3 .....	19
Hình 15: Testcase bài 6.4 .....	20

## NỘI DUNG BÀI LÀM

**Bài 1. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên. Cài đặt hàm thực hiện các chức năng sau:**

- a. Tính tổng các số lẻ trong mảng**
- b. Đếm số nguyên tố trong mảng**
- c. Tìm số chính phương nhỏ nhất (nếu không có hàm trả về -1)**

Mô tả đầu vào và đầu ra của bài toán:

- Input: Một mảng một chiều gồm n số nguyên ngẫu nhiên.
- Output:
  - o Tổng các số lẻ trong mảng,
  - o Đếm số nguyên tố trong mảng
  - o Số chính phương nhỏ nhất.

Nội dung code của Bài 1

*Table 1 Nội dung code của chương trình bài 1*

```
using System;
using System.Collections.Generic;

namespace Bai01
{
    class Mang
    {
        private List<int> a;
        public Mang()
        {
            a = new List<int>();
        }

        public void Nhap()
        {
            Console.WriteLine("Nhap so phan tu cho mang: ");
            int n = int.Parse(Console.ReadLine());

            Random rd = new Random();
            for (int i = 0; i < n; i++)
            {
                int x = rd.Next(100);
                a.Add(x);
            }
        }

        public void Xuat()
        {
            Console.WriteLine("Mang gom cac phan tu: ");
            for (int i = 0; i < a.Count; i++)
            {

```

```
        Console.WriteLine($"Phan tu thu {i+1}: {a[i]} ");
    }
}

public int TongCacSoLe()
{
    int Tong = 0;
    for (int i = 0; i < a.Count; i++)
    {
        if (a[i] % 2 != 0)
        {
            Tong += a[i];
        }
    }
    return Tong;
}

static bool LaSoNguyenTo(int n)
{
    if (n < 2) return false;
    for (int i = 2; i <= Math.Sqrt(n); i++)
    {
        if (n % i == 0) return false;
    }
    return true;
}

public int DemSoNguyenTo()
{
    int count = 0;
    for (int i = 0; i < a.Count; i++)
    {
        if (LaSoNguyenTo(a[i]))
        {
            count++;
        }
    }
    return count;
}

static bool LaSoChinhPhuong(int n)
{
    int i = (int)Math.Sqrt(n);
    if (i * i == n) return true;
    return false;
}

public int SoChinhPhuongMin()
{
    int Min = int.MaxValue;
    for (int i = 0; i < a.Count; i++)
    {
        if (LaSoChinhPhuong(a[i]) && a[i] < Min)
        {
            Min = a[i];
        }
    }
    if (Min == int.MaxValue) return -1;
    return Min;
}
}

internal static class Program
{
    static void Main(string[] args)
    {

```

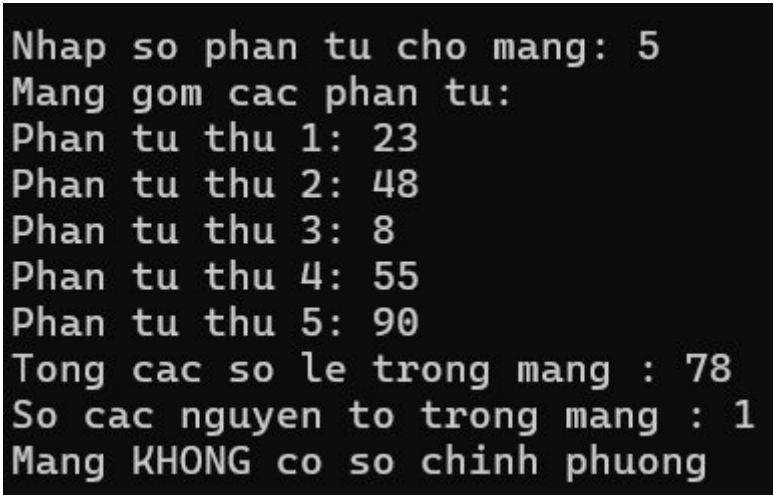
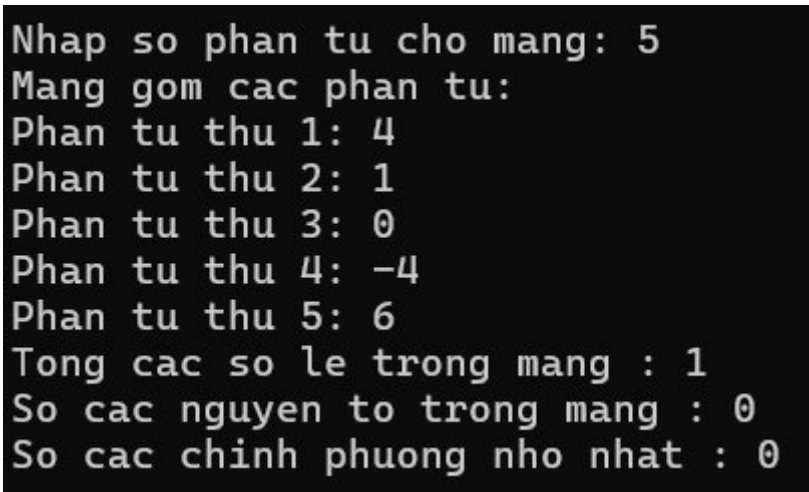
## IT008 – LẬP TRÌNH TRỰC QUAN

```
Mang a = new Mang();
a.Nhap();
a.Xuat();

Console.WriteLine($"Tong cac so le trong mang : {a.TongCacSoLe()}");
Console.WriteLine($"So cac nguyen to trong mang : {a.DemSoNguyenTo()}");
int CP_Min = a.SoChinhPhuongMin();
if (CP_Min != -1)
{
    Console.WriteLine($"So cac chinh phuong nho nhat : {CP_Min}");
}
else
{
    Console.WriteLine("Mang KHONG co so chinh phuong");
}
}
```

Các Testcase của chương trình:

Table 2 Testcase của chương trình bài 1

<p>Input: n = 5</p> <p>Output:</p> <p>a. 78 (23 + 55 = 78)</p> <p>b. 1 (Số nguyên tố: 23)</p> <p>c. -1 (KHÔNG có số chính phương)</p>	 <p>Hình 1: Testcase bài 1.1</p>
<p>Input: n = 5</p> <p>Output:</p> <p>a. 1 (Chỉ có 1 số lẻ : 1)</p> <p>b. 0 (Mảng không có số nguyên tố)</p> <p>c. 0 (0 = 0^2: là số chính phương nhỏ nhất)</p>	 <p>Hình 2: Testcase bài 1.2</p>



Input: n =5

Output:

a. 31

(25 + 5 + 1 = 31)

b. 1

(Số nguyên tố: 5)

c. 1

(Số chính phương: 25,  
36, 1. Số nhỏ nhất : 1)

```
Nhap so phan tu cho mang: 5
Mang gom cac phan tu:
Phan tu thu 1: 25
Phan tu thu 2: 5
Phan tu thu 3: 36
Phan tu thu 4: 12
Phan tu thu 5: 1
Tong cac so le trong mang : 31
So cac nguyen to trong mang : 1
So cac chinh phuong nho nhât : 1
```

Hình 3: Testcase bài 1.3

## Bài 2: Viết chương trình nhập số nguyên dương n, tính tổng các số nguyên tố < n và xuất kết quả ra màn hình.

Mô tả đầu vào và đầu ra của bài toán:

- Input: Số nguyên dương n
- Output: tổng các số nguyên tố < n

Nội dung code của Bài 2

Table 3 Nội dung code của chương trình bài 2

```
using System;

namespace Bai02
{
    internal static class Program
    {
        static bool LaSoNguyenTo(int n)
        {
            if (n < 2) return false;
            for (int i = 2; i <= Math.Sqrt(n); i++)
            {
                if (n % i == 0) return false;
            }
            return true;
        }

        static void Main()
        {
            int n;
            do
            {
                Console.Write("Nhap so nguyen duong: ");
                n = int.Parse(Console.ReadLine());
            } while (n <= 0);
            long Tong = 0;
```

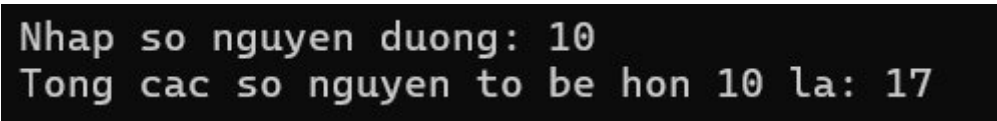
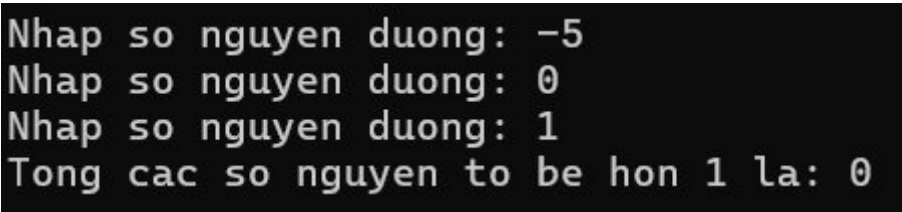
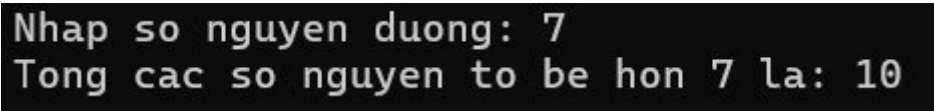
## IT008 – LẬP TRÌNH TRỰC QUAN

```
for (int i = 2; i < n; i++)
{
    if (LaSoNguyenTo(i))
        Tong += i;
}

Console.WriteLine($"Tong cac so nguyen to be hon {n} la: {Tong}");
}
}
```

Các Testcase của chương trình:

Table 4 Testcase của chương trình bài 2

Input: 10 Output: 17 (2 + 3 + 5 + 7 = 17)	 <p>Hình 4: Testcase bài 2.1</p>
Input: 1 Output: 0 (Không có số nguyên tổ < 1)	 <p>Hình 5: Testcase bài 2.2</p>
Input: 7 Output: 10 (2 + 3 + 5 = 10)	 <p>Hình 6: Testcase bài 2.3</p>

**Bài 3: Viết chương trình nhập ngày tháng năm, cho biết ngày tháng năm đó có hợp lệ không?**

**Bài 4: Viết chương trình nhập vào tháng và năm. In ra số ngày của tháng đó.**

**Bài 5: Viết chương trình nhập vào ngày tháng năm, cho biết thứ trong tuần**

Mô tả đầu vào và đầu ra của bài toán:

- Input: 3 số nguyên lần lượt tương ứng ngày, tháng, năm
- Output:
  - o Ngày tháng năm đó có hợp lệ không,
  - o Số ngày của tháng đó (nếu tháng hợp lệ)
  - o Thứ trong tuần (nếu ngày, tháng, năm hợp lệ)

Nội dung code của Bài 3+4+5

*Table 5 Nội dung code của chương trình bài 3+4+5*

```
using System;

namespace Bai03_04_05
{
    internal static class Program
    {
        static bool LaNamNhuan(int nam)
        {
            return (nam % 400 == 0) || (nam % 4 == 0 && nam % 100 != 0);
        }
        static bool ThoiGianHopLe(int ngay, int thang, int nam)
        {
            if (thang == 4 || thang == 6 || thang == 9 || thang == 11)
            {
                if (ngay >= 1 && ngay <= 30) return true;
            }
            else if (thang == 1 || thang == 3 || thang == 5 || thang == 7 ||
                thang == 8 || thang == 10 || thang == 12)
            {
                if (ngay >= 1 && ngay <= 31) return true;
            }
            else if (thang == 2)
            {
                if (LaNamNhuan(nam))
                {
                    if (ngay >= 1 && ngay <= 29) return true;
                }
                else
                {
                    if (ngay >= 1 && ngay <= 28) return true;
                }
            }
        }
        return false;
    }
}
```

```
}

static int SoNgay(int thang, int nam)
{
    if (thang == 4 || thang == 6 || thang == 9 || thang == 11)
    {
        return 30;
    }
    else if (thang == 1 || thang == 3 || thang == 5 || thang == 7 ||
        thang == 8 || thang == 10 || thang == 12)
    {
        return 31;
    }
    else if (thang == 2)
    {
        if (LaNamNhuan(nam))
        {
            return 29;
        }
        else
        {
            return 28;
        }
    }
    return 0; // TH ngay thang khong hop le
}

static long TongSoNgay (int ngay, int thang, int nam)
{
    long Tong = 0;

    for (int i = 0; i < nam; i++)
    {
        Tong += 365;
        if (LaNamNhuan(nam)) Tong++;
    }

    for (int i = 1; i < thang; i++)
    {
        Tong += SoNgay(i, nam);
    }

    Tong += ngay;
    return Tong;
}

static string LaThu(int ngay, int thang, int nam )
{
    int thu = (int) TongSoNgay(ngay, thang, nam) % 7;
    string[] CacThuTrongTuan = { "Chu Nhat", "Thu Hai", "Thu Ba", "Thu Tu", "Thu Nam", "Thu Sau",
"Thu Bay" };
    return CacThuTrongTuan[thu];
}

static void Main()
{
    Console.Write("Nhap ngay: ");
    int ngay = int.Parse(Console.ReadLine());

    Console.Write("Nhap thang: ");
    int thang = int.Parse(Console.ReadLine());

    Console.Write("Nhap nam: ");
    int nam = int.Parse(Console.ReadLine());
}
```

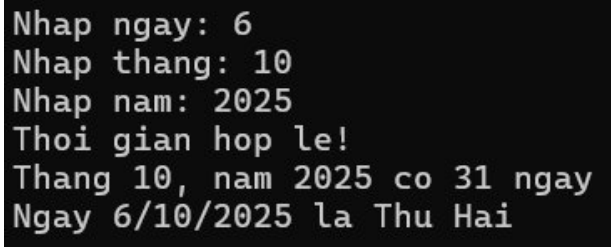
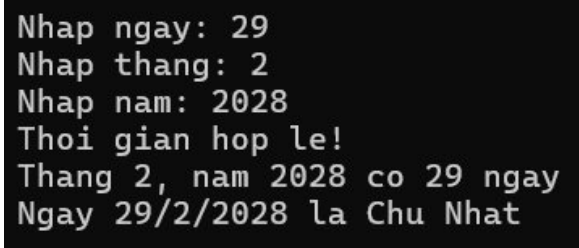
```

if (!ThoiGianHopLe(ngay, thang, nam)) {
    Console.WriteLine($"Thoi gian KHONG hop le!");
    if (thang < 1 || thang > 12)
    {
        Console.WriteLine($"Thang {thang} khong hop le!");
        if (ngay < 1 || ngay > 31)
            Console.WriteLine($"Ngay {ngay} khong hop le!");
    }
    else
    {
        Console.WriteLine($"Ngay {ngay} khong hop le!");
        Console.WriteLine($"Thang {thang}, nam {nam} chi co { SoNgay(thang, nam) } ngay.");
    }
}
else
{
    Console.WriteLine($"Thoi gian hop le!");
    Console.WriteLine($"Thang {thang}, nam {nam} co {SoNgay(thang, nam)} ngay");
    Console.WriteLine($"Ngay {ngay}/{thang}/{nam} la { LaThu(ngay, thang, nam) } ");
}
}
}
}

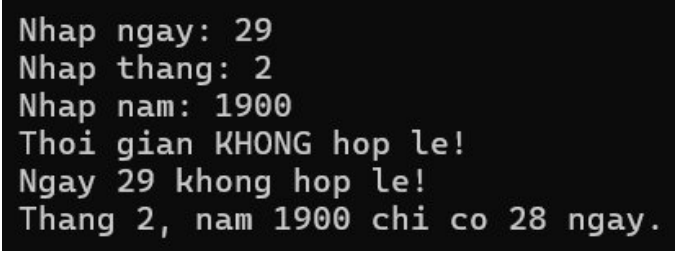
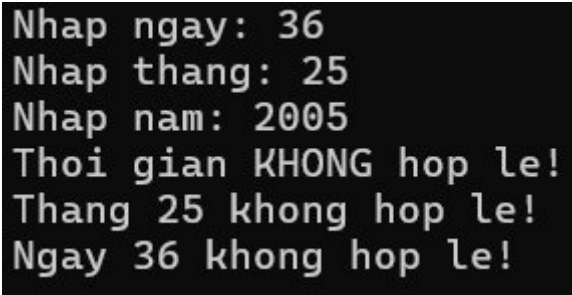
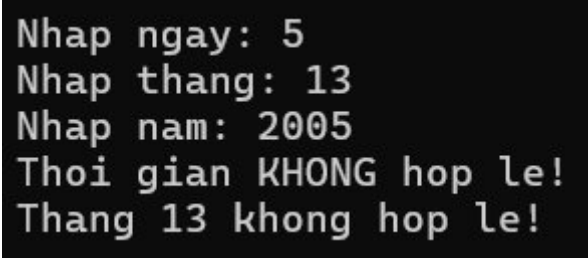
```

Các Testcase của chương trình:

Table 6 Testcase của chương trình bài 3+4+5

<p>Input: 6/10/2025</p> <p>Output:</p> <ul style="list-style-type: none"> <li>- “Thoi gian hop le”</li> <li>- Thang 10 nam 2025 co 31 ngay</li> <li>- Thứ Hai</li> </ul>	 <p>Hình 7: Testcase bài 3+4+5.1</p>
<p>Input: 29/2/2028</p> <p>Output:</p> <ul style="list-style-type: none"> <li>- “Thoi gian hop le”</li> <li>- Thang 2 nam 2028 co 29 ngay</li> <li>- Chủ Nhật</li> </ul>	 <p>Hình 8: Testcase bài 3+4+5.2</p>

## IT008 – LẬP TRÌNH TRỰC QUAN

<p>Input: 29/2/1900</p> <p>Output:</p> <ul style="list-style-type: none"><li>- “Thoi gian KHÔNG hop le”</li></ul> <p>(Ngày không hợp lệ)</p> <ul style="list-style-type: none"><li>- Tháng 2 năm 1900 có 28 ngày</li></ul>	 <p>Hình 9: Testcase bài 3+4+5.3</p>
<p>Input: 36//25/2005</p> <p>Output:</p> <ul style="list-style-type: none"><li>- “Thoi gian KHÔNG hop le”</li></ul> <p>(Tháng và ngày không hợp lệ)</p>	 <p>Hình 10: Testcase bài 3+4+5.4</p>
<p>Input: 5/13/2005</p> <p>Output:</p> <ul style="list-style-type: none"><li>- “Thoi gian KHÔNG hop le”</li></ul> <p>(Tháng không hợp lệ)</p>	 <p>Hình 11: Testcase bài 3+4+5.5</p>

**Bài 6. Viết chương trình cho phép tạo ma trận chứa các số nguyên ngẫu nhiên gồm n dòng, m cột. Cài đặt hàm thực hiện các chức năng sau:**

- Xuất ma trận**
- Tìm phần tử lớn nhất/nhỏ nhất**
- Tìm dòng có tổng lớn nhất**
- Tính tổng các số không phải là số nguyên tố**
- Xóa dòng thứ k trong ma trận**
- Xóa cột chứa phần tử lớn nhất trong ma trận**

Mô tả đầu vào và đầu ra của bài toán:

- Input: ma trận chứa các số nguyên ngẫu nhiên gồm n dòng, m cột
- Output:
  - o Xuất ma trận

## IT008 – LẬP TRÌNH TRỰC QUAN

- Phần tử lớn nhất/nhỏ nhất
- Dòng có tổng lớn nhất
- Tổng các số không phải là số nguyên tố
- Xóa dòng thứ k trong ma trận
- Xóa cột chứa phần tử lớn nhất trong ma trận

Nội dung code của Bài 6

*Table 7 Nội dung code của chương trình bài 6*

```
using System;
using System.Collections.Generic;

namespace Bai01
{
    class Matrix
    {
        private int row;
        private int col;
        private List<List<int>> m;

        public Matrix()
        {
            m = new List<List<int>>();
        }

        public Matrix(int row, int col)
        {
            this.row = row;
            this.col = col;
            m = new List<List<int>>();
            for (int i = 0; i < row; i++)
            {
                List<int> temp = new List<int>();
                for (int j = 0; j < col; j++)
                {
                    temp.Add(0);
                }
                m.Add(temp);
            }
        }

        public void Nhap()
        {
            Console.WriteLine("Nhap so hang cua ma tran: ");
            row = int.Parse(Console.ReadLine());

            Console.WriteLine("Nhap so cot cua ma tran: ");
            col = int.Parse(Console.ReadLine());

            Random rd = new Random();

            for (int i = 0; i < row; i++)
            {
                List<int> temp = new List<int>();
                for (int j = 0; j < col; j++)
                {
                    int x = rd.Next(100);
                    temp.Add(x);
                }
            }
        }
    }
}
```

```
        m.Add(temp);
    }
}

public void Xuat(string GhiChu)
{
    Console.WriteLine(GhiChu);
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            Console.Write($"m[{i}][{j}] ");
        }
        Console.WriteLine();
    }
}

public int SoLonNhat()
{
    int max = m[0][0];
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            if (max < m[i][j]) max = m[i][j];
        }
    }
    return max;
}

public int SoNhoNhat()
{
    int min = m[0][0];
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            if (min > m[i][j]) min = m[i][j];
        }
    }
    return min;
}

public void TimDong_TongMax()
{
    int TongMax = int.MinValue;
    int DongMax = 0;

    for (int i = 0; i < row; i++)
    {
        int Tong = 0;
        for (int j = 0; j < col; j++)
            Tong += m[i][j];

        if (Tong > TongMax)
        {
            TongMax = Tong;
            DongMax = i;
        }
    }

    Console.WriteLine($"Dong co tong lon nhat la dong thu {DongMax + 1} :");
    for (int j = 0; j < col; j++)
    {
```



```
        Console.WriteLine($"{m[DongMax][j]}");
        if (j != col - 1) Console.Write(" + ");
    }
    Console.WriteLine($" = {TongMax}");
}

static bool LaSoNguyenTo(int n)
{
    if (n < 2) return false;
    for (int i = 2; i <= Math.Sqrt(n); i++)
    {
        if (n % i == 0) return false;
    }
    return true;
}

public int TongSoKhongNguyenTo()
{
    int Tong = 0;
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            if (!LaSoNguyenTo(m[i][j]))
                Tong += m[i][j];
        }
    }
    return Tong;
}

public void XoaDong(int k)
{
    if (k < 0 || k >= row)
    {
        Console.WriteLine("So dong khong hop le!");
        return;
    }

    m.RemoveAt(k);
    row--;
}

public void XoaCotChuaMax()
{
    int max = SoLonNhat();
    SortedSet<int> colsMax = new SortedSet<int>(); //Tránh giá trị trùng nhau và tự sắp xếp thứ tự

    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            if (max == m[i][j])
                colsMax.Add(j);
        }
    }

    // xóa từ trái sang phải sẽ làm thay đổi vị trí cột của bảng --> xóa từ phải sang trái
    foreach (var c in colsMax.Reverse())
    {
        for (int i = 0; i < row; i++)
        {
            m[i].RemoveAt(c);
        }
    }
}
```

```

        col--;
        if (col == 0) return;
    }
}

internal static class Program
{
    static void Main(string[] args)
    {
        Matrix m = new Matrix();
        m.Nhap();
        m.Xuat("Ma tran da nhap: ");

        Console.WriteLine($"So lon nhat: {m.SoLonNhat()}");
        Console.WriteLine($"So nho nhat: {m.SoNhoNhat()}");

        m.TimDong_TongMax();

        Console.WriteLine($"Tong cac so KHONG nguyen to: {m.TongSoKhongNguyenTo()}");

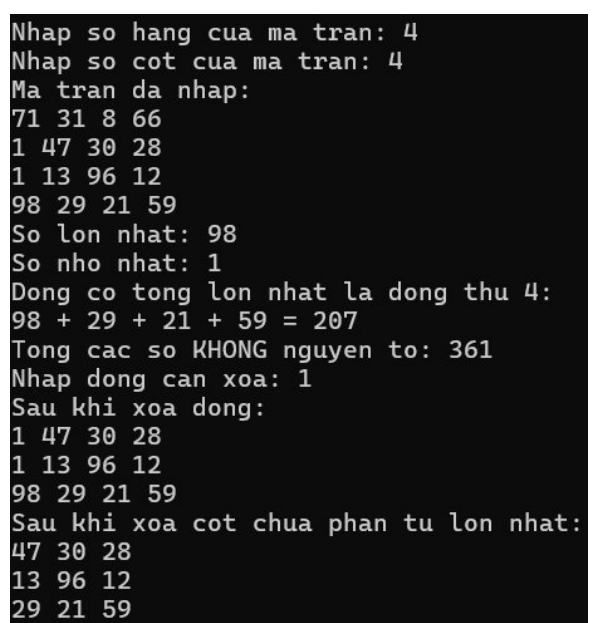
        Console.Write("Nhap dong can xoa: ");
        int dong = int.Parse(Console.ReadLine());
        m.XoaDong(dong - 1);
        m.Xuat("Sau khi xoa dong:");

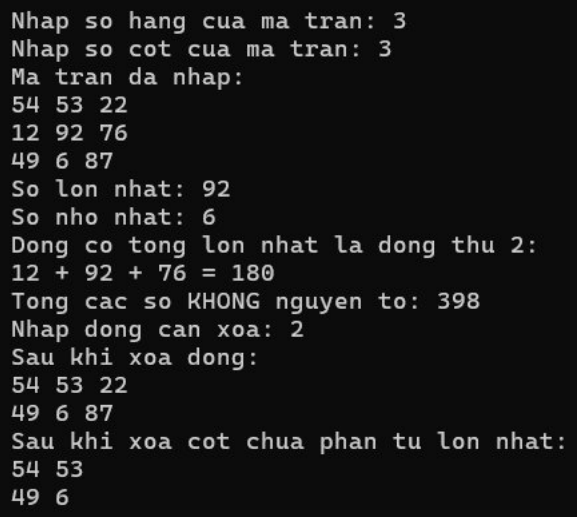
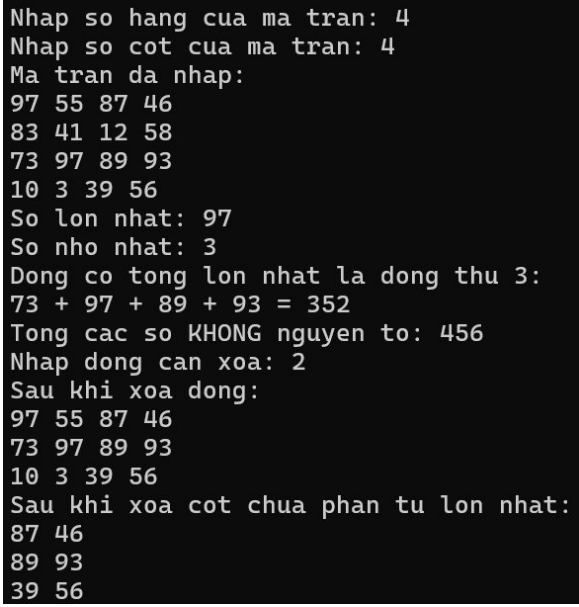
        m.XoaCotChuaMax();
        m.Xuat("Sau khi xoa cot chua phan tu lon nhat:");
    }
}

```

Các Testcase của chương trình:

Table 8 Testcase của chương trình bài 6

<p>Input: m = 4, n = 4, k = 1 (dòng không chứ Max)</p> <p>Output:</p> <ol style="list-style-type: none"> <li>71 31 8 66 1 47 30 28 1 13 96 12 98 29 21 59</li> <li>Max: 98 Min: 1</li> <li>4</li> <li>361</li> <li>1 47 30 28 1 13 96 12 98 29 21 59</li> <li>(Max = 98 cột 1) 47 30 28 13 96 12</li> </ol>	 <p>The screenshot shows the following output from the program:</p> <pre> Nhap so hang cua ma tran: 4 Nhap so cot cua ma tran: 4 Ma tran da nhap: 71 31 8 66 1 47 30 28 1 13 96 12 98 29 21 59 So lon nhat: 98 So nho nhat: 1 Dong co tong lon nhat la dong thu 4: 98 + 29 + 21 + 59 = 207 Tong cac so KHONG nguyen to: 361 Nhap dong can xoa: 1 Sau khi xoa dong: 1 47 30 28 1 13 96 12 98 29 21 59 Sau khi xoa cot chua phan tu lon nhat: 47 30 28 13 96 12 29 21 59 </pre> <p>Hình 12: Testcase bài 6.1</p>
---	--

<p>29 21 59</p> <p>Input: m = 3, n = 3, k = 2 (dòng chữ Max)</p> <p>Output:</p> <ul style="list-style-type: none"> <li>a. 54 53 22 12 92 76 49 6 87</li> <li>b. Max: 91 Min: 6</li> <li>c. 2</li> <li>d. 398</li> <li>e. 54 53 22 49 6 87</li> <li>f. (Max = 87 cột 3) 54 53 49 6</li> </ul>	 <p>Hình 13: Testcase bài 6.2</p>
<p>Input: m = 4, n = 4, k = 2</p> <p>Output:</p> <ul style="list-style-type: none"> <li>a. 97 55 87 46 83 41 12 58 73 97 89 93 10 3 39 56</li> <li>b. Max: 97 Min: 3</li> <li>c. 3</li> <li>d. 456</li> <li>e. 97 55 87 46 73 97 89 93 10 3 39 56</li> <li>f. (Max = 97 cột 1, cột 2) 87 46 89 93 39 56</li> </ul>	 <p>Hình 14: Testcase bài 6.3</p>

## IT008 – LẬP TRÌNH TRỰC QUAN

Input:  $m = 4, n = 4, k = 4$

Output:

- a. 49 89 46 16  
44 92 92 26  
52 54 92 36  
12 15 78 28
- b. Max: 92  
Min: 12
- c. 2
- d. 732
- e. 49 89 46 16  
44 92 92 26  
52 54 92 36
- f. (Max = 92 cột 1, cột 2 – xuất  
hiện 2 lần)  
49 16  
44 26  
52 36

```
Ma tran da nhap:
49 89 46 16
44 92 92 26
52 54 92 36
12 15 78 28
So lon nhat: 92
So nho nhat: 12
Dong co tong lon nhat la dong thu 2:
44 + 92 + 92 + 26 = 254
Tong cac so KHONG nguyen to: 732
Nhap dong can xoa: 4
Sau khi xoa dong:
49 89 46 16
44 92 92 26
52 54 92 36
Sau khi xoa cot chua phan tu lon nhat:
49 16
44 26
52 36
```

Hình 15: Testcase bài 6.4

LinkCode: [https://github.com/noc-turne-git/LTTQ\\_TH\\_BT1](https://github.com/noc-turne-git/LTTQ_TH_BT1)