

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 1

GVHD: Nguyễn Ngọc Quý

Sinh viên thực hiện: Nguyễn Huỳnh Anh Trung

MSSV: 24521886

This image shows a full page of white paper with horizontal dotted lines. The lines are evenly spaced and run across the width of the page, providing a guide for handwriting practice. There are no margins, text, or other markings on the page.

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

MỤC LỤC

1. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên.....	5
2. Viết chương trình nhập số nguyên dương n , tính tổng các số nguyên tố $< n$ và xuất kết quả ra màn hình.	12
3. Viết chương trình nhập ngày tháng năm, cho biết ngày tháng năm đó có hợp lệ không?	15
4. Viết chương trình nhập vào tháng và năm. In ra số ngày của tháng đó.....	19
5. Viết chương trình nhập vào ngày tháng năm, cho biết thứ trong tuần... ..	23
6. Viết chương trình cho phép tạo ma trận chứa các số nguyên ngẫu nhiên gồm n dòng, m cột.....	28

DANH MỤC BẢNG

Bảng 1.1 Nội dung code của chương trình Bài 1.....	5
Bảng 1.2 Giải thích code của chương trình Bài 1.....	8
Bảng 1.3 Kết quả chương trình Bài 1.....	9
Bảng 2.1 Nội dung code của chương trình Bài 2.....	12
Bảng 2.2 Giải thích code của chương trình Bài 2.....	13
Bảng 2.3 Kết quả chương trình Bài 2.....	14
Bảng 3.1 Nội dung code của chương trình Bài 3.....	15
Bảng 3.2 Giải thích code của chương trình Bài 3.....	16
Bảng 3.3 Kết quả chương trình Bài 3.....	17
Bảng 4.1 Nội dung code của chương trình Bài 4.....	19
Bảng 4.2 Giải thích code của chương trình Bài 4.....	20
Bảng 4.3 Kết quả chương trình Bài 4.....	21
Bảng 5.1 Nội dung code của chương trình Bài 5.....	23
Bảng 5.2 Giải thích code của chương trình Bài 5.....	25
Bảng 5.3 Kết quả chương trình Bài 5.....	26
Bảng 6.1 Nội dung code của chương trình Bài 6.....	28
Bảng 6.2 Giải thích code của chương trình Bài 6.....	33
Bảng 6.3 Kết quả chương trình Bài 6.....	35

NỘI DUNG BÀI LÀM

1. Viết chương trình cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên.

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: Kết quả tính tổng các số lẻ trong mảng, đếm số nguyên tố trong mảng và tìm số chính phương nhỏ nhất.

➤ Nội dung code của Bài 1

Bảng 1.1 Nội dung code của chương trình Bài 1

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

namespace BTTH1_BT1 {
    class RandomArray {
        const int MIN = -100;
        const int MAX = 100;
        private int[] array;
        private int n;
        public int Lenght {
            get { return n; }
        }

        //Constructor có tham số đầu vào là số phần tử mảng
        public RandomArray(int n) {
            this.n = n;
            array = new int[n];
            Random rand = new Random();
            for(int i = 0; i < n; ++i) {
                array[i] = rand.Next(MIN, MAX);
            }
        }

        //Tổng các số lẻ trong mảng
        public int sum_Odds() {
            int sum = 0;
            for(int i = 0; i < n; ++i) {
                if (array[i] % 2 != 0) {

```

```

        sum += array[i];
    }
}
return sum;
}

// Kiem tra la so nguyen to
public static bool check_Prime(int n) {
    if (n <= 1)
        return false;
    if (n == 2)
        return true;
    for(int i = 2; i <= (int)Math.Sqrt(n); ++i) {
        if(n % i == 0) {
            return false;
        }
    }
    return true;
}

// Dem so nguyen to trong mang
public int count_Prime() {
    int count = 0;
    for(int i = 0; i < n; ++i) {
        if (check_Prime(array[i]) == true) {
            count++;
        }
    }
    return count;
}

//Kiem tra la so chinh phuong
public static bool check_Square(int n) {
    if (n < 0)
        return false;
    int square = (int)Math.Sqrt(n);
    return (square * square == n);
}

//Tim so chinh phuong nho nhat
public int find_min_Square() {
    int min_square = -1;
    for(int i = 0; i < n; ++i) {
        if ((check_Square(array[i]) == true) && (min_square == -1
|| array[i] < min_square)) {
            min_square = array[i];
        }
    }
    return min_square;
}

//In mang
public void Print() {
    for(int i = 0; i < n; ++i) {
        Console.Write(array[i] + " ");
    }
    Console.WriteLine();
}

}

public class Program {

```

```

//Kiểm tra số đầu vào
public static int read_int() {
    int n;
    try {
        n = int.Parse(Console.ReadLine());
        if(n <= 0) {
            throw new ArgumentOutOfRangeException();
        }
    }
    catch(Exception e) {
        Console.WriteLine(e.Message);
        throw new Exception(e.Message);
    }
    return n;
}

public static void Main() {
    //Nhập số phần tử của mảng và tạo mảng
    Console.WriteLine("Nhập số phần tử mảng (n > 0): ");
    int n = read_int();
    RandomArray rArray = new RandomArray(n);

    int choice;
    do {
        //In menu
        Console.WriteLine("\n===== MENU =====");
        Console.WriteLine("1. Thoát");
        Console.WriteLine("2. In mảng");
        Console.WriteLine("3. Tính tổng các số lẻ");
        Console.WriteLine("4. Đếm số nguyên tố");
        Console.WriteLine("5. Tìm số chính phương nhỏ nhất");
        Console.WriteLine("Chọn chức năng: ");

        choice = read_int();

        switch (choice) {
            case 1:
                Console.WriteLine("Kết thúc chương trình");
                break;
            case 2:
                Console.WriteLine("Mảng các số nguyên:");
                rArray.Print();
                break;
            case 3:
                Console.WriteLine("Tổng các số lẻ là: " +
rArray.sum_Odds());
                break;
            case 4:
                Console.WriteLine("Số nguyên tố trong mảng là: "
+ rArray.count_Prime());
                break;
            case 5:
                Console.WriteLine("Số chính phương nhỏ nhất là: "
+ rArray.find_min_Square());
                break;
            default:
                Console.WriteLine("Không có chức năng nào ngoài
MENU");
                throw new ArgumentOutOfRangeException();
        }
    }
}

```

```

        } while (choice != 1);
    }
}
}

```

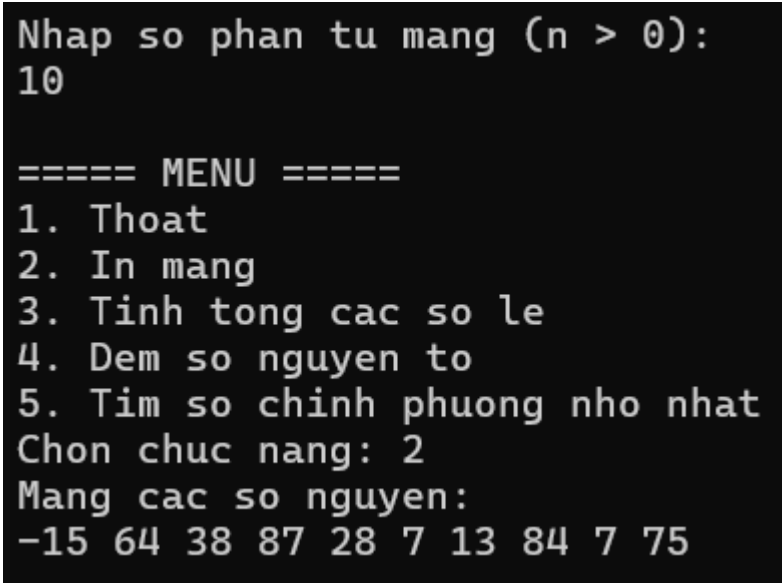
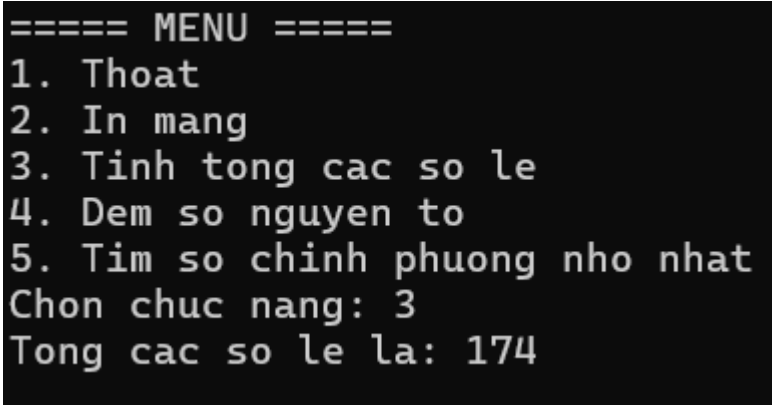
➤ Giải thích code của chương trình Bài 1

Bảng 1.2 Giải thích code của chương trình Bài 1

Constructor và phương thức	Giải thích
RandomArray(int n)	Constructor khởi tạo mảng có n phần tử ngẫu nhiên.
sum_Odds()	Phương thức tính tổng các phần tử lẻ trong mảng.
check_Prime(int n)	Phương thức kiểm tra số đó có là số nguyên tố.
count_Prime()	Phương thức đếm tổng các số nguyên tố trong mảng.
check_Square(int n)	Phương thức kiểm tra số đó có là số chính phương.
find_min_Square()	Phương thức tìm số chính phương nhỏ nhất, không có thì trả về 1.
Print()	Phương thức in toàn bộ mảng ra màn hình.
read_int()	Phương thức đọc số nguyên từ bàn phím, đồng thời kiểm tra lớn hơn 0.

➤ Các Testcase của chương trình:

Bảng 1.3 Kết quả chương trình Bài 1

Chức năng	Input Output
Tạo mảng có 10 phần tử và in ra	<p>10</p> <p>2</p> 
Tính tổng các số lẻ	<p>3</p> 
Đếm các số nguyên tố	4

	<pre> ===== MENU ===== 1. Thoat 2. In mang 3. Tinh tong cac so le 4. Dem so nguyen to 5. Tim so chinh phuong nho nhat Chon chuc nang: 4 So nguyen to trong mang la: 3 </pre>
<p> Tìm số chính phương nhỏ nhất và thoát </p>	<pre> 5 1 ===== MENU ===== 1. Thoat 2. In mang 3. Tinh tong cac so le 4. Dem so nguyen to 5. Tim so chinh phuong nho nhat Chon chuc nang: 5 So chinh phuong nho nhat la: 64 ===== MENU ===== 1. Thoat 2. In mang 3. Tinh tong cac so le 4. Dem so nguyen to 5. Tim so chinh phuong nho nhat Chon chuc nang: 1 Ket thuc chuong trinh </pre>
<p> Trường hợp gây lỗi </p>	<ul style="list-style-type: none"> ❖ Tạo mảng có 0 phần tử -> throw exception: Dừng chương trình <pre>Specified argument was out of the range of valid values.</pre> <ul style="list-style-type: none"> ❖ Chọn chức năng ngoài MENU -> Dừng chương trình

```
===== MENU =====
```

```
1. Thoat
```

```
2. In mang
```

```
3. Tinh tong cac so le
```

```
4. Dem so nguyen to
```

```
5. Tim so chinh phuong nho nhat
```

```
Chon chuc nang: 8
```

```
Khong co chuc nang nam ngoai MENU
```

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: Một số nguyên dương n .
- Output: Tổng các số nguyên tố nhỏ hơn n .

➤ Nội dung code của Bài 2

Bảng 2.1 Nội dung code của chương trình Bài 2

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

namespace BTTH1_BT2 {
    public class Program {
        //Kiểm tra số đầu vào
        public static int check_input() {
            int x;
            try {
                x = int.Parse(Console.ReadLine());
                if (x != -1 && x < 0)
                    throw new ArgumentOutOfRangeException("Phải là số
duong");
            } catch (Exception e) {
                Console.WriteLine(e.Message);
                throw new Exception(e.Message);
            }
            return x;
        }

        //Kiểm tra số nguyên tố
        public static bool check_Prime(int x) {
            if (x == 2) {
                return true;
            }
            for (int i = 2; i <= (int)Math.Sqrt(x); ++i) {
                if (x % i == 0) {
                    return false;
                }
            }
            return true;
        }

        //Tổng các số nguyên tố nhỏ hơn n
        public static int sum_Prime(int n) {
            int sum = 0;
            for (int i = 2; i < n; ++i) {
                if (check_Prime(i) == true) {
                    sum += i;
                }
            }
        }
    }
}
```

```

    }
    return sum;
}

public static void Main() {
    //Nhập số nguyên dương n
    int n;
    do {
        Console.WriteLine("Nhập một số nguyên dương:");
        n = check_input();
        if (n == -1)
            break;

        Console.WriteLine("Tổng các số nguyên tố nhỏ hơn n là: "
+ sum_Prime(n));
        Console.WriteLine("Nhập -1 để thoát.");
    } while (true);
}
}

```

➤ Giải thích code của chương trình Bài 2

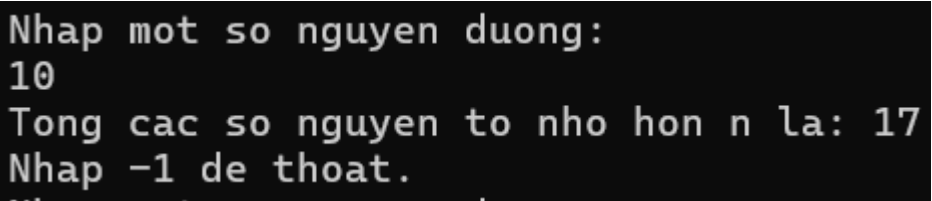
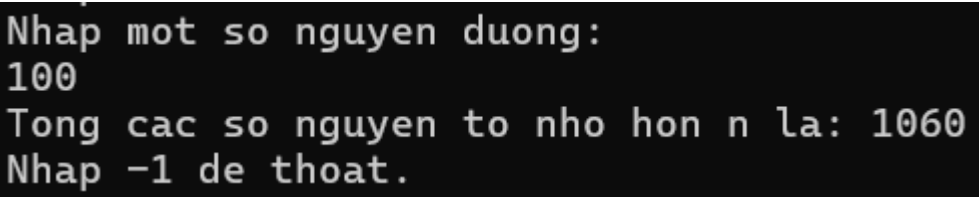
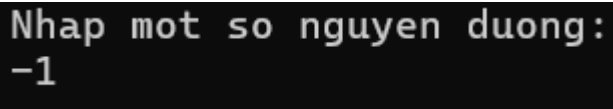
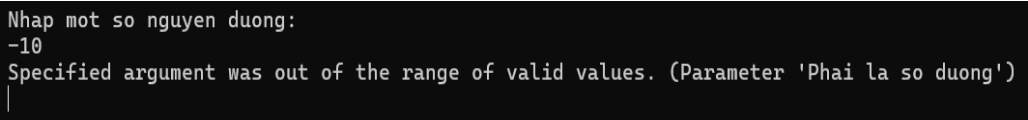
Bảng 2.2 Giải thích code của chương trình Bài 2

Constructor và phương thức	Giải thích
check_input()	Phương thức đọc số nguyên từ bàn phím, đồng thời kiểm tra số lớn hơn 0.
check_Prime(int n)	Phương thức kiểm tra số đó có phải là số nguyên tố.
sum_Prime()	Phương thức cộng các số nguyên tố nhỏ hơn n.

IT008 – LẬP TRÌNH TRỰC QUAN

➤ Các testcase của chương trình:

Bảng 2.3 Kết quả chương trình Bài 2

Chức năng	Input Output
Tính tổng các số nguyên tố nhỏ hơn n	10 
Tính tổng các số nguyên tố nhỏ hơn n	100 
Thoát	-1 
Trường hợp gây lỗi	-10 -> throw exception: Dừng chương trình 

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

Mô tả đầu vào và đầu ra của bài toán:

- Input: Nhập 3 số nguyên dương ngày tháng năm. Mỗi số trên 1 dòng.
- Output: Trả về ngày tháng năm đó có hợp lệ hay không.

➤ Nội dung code của Bài 3

Bảng 3.1 Nội dung code của chương trình Bài 3

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

namespace BTTH1_BT3 {
    public class Program {
        //Ngày của các tháng trong năm bắt đầu từ index 1
        private static int[] valid_day = { -1, 31, 28, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31 };

        //Kiểm tra năm nhuận
        public static bool check_leap_year(int year) {
            return (year % 400 == 0) || (year % 4 == 0 && year % 100 !=
0);
        }

        //Kiểm tra số đầu vào
        public static int check_input() {
            int x;
            string tmp = Console.ReadLine();
            if (String.IsNullOrEmpty(tmp)) {
                Console.WriteLine("Thoát chương trình");
                Environment.Exit(0);
            }
            try {
                x = int.Parse(tmp);
            }
            catch (Exception e) {
                Console.WriteLine(e.Message);
                throw new Exception(e.Message);
            }
            return x;
        }

        //Kiểm tra thời gian hợp lệ
        public static bool check_valid(int ngay, int thang, int nam) {
            if (ngay < 1 || thang < 1 || thang > 12)
                return false;
            int leap = 0;
            if (thang == 2 && check_leap_year(nam) == true) {
                leap = 1;
            }
        }
    }
}
```

```

    }
    return ngay <= (valid_day[thang] + leap);
}

public static void Main() {
    //Nhap ngay thang nam
    while (true) {
        Console.WriteLine("Nhap ngay, thang, nam:");
        int ngay = check_input();

        int thang = check_input();
        int nam = check_input();

        if (check_valid(ngay, thang, nam) == true) {
            Console.WriteLine("Ngày tháng năm hợp lệ");
        } else {
            Console.WriteLine("Ngày tháng năm không hợp lệ");
        }
        Console.WriteLine("Nhap Enter để thoát chương trình");
    }
}
}
}

```

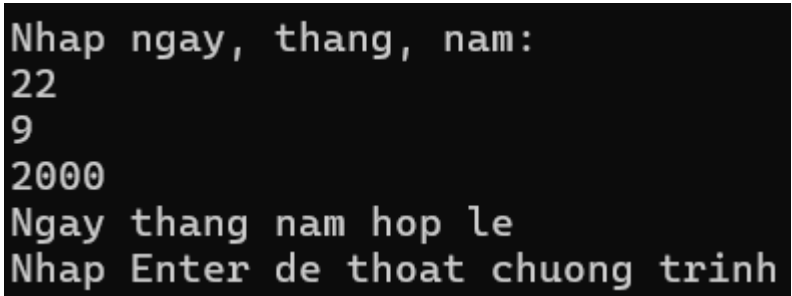
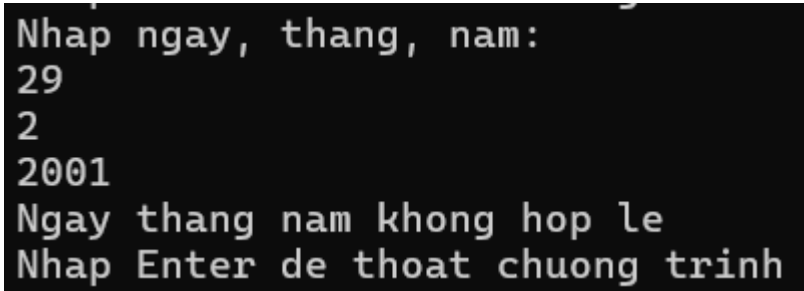
➤ Giải thích code của chương trình Bài 3

Bảng 3.2 Giải thích code của chương trình Bài 3

Constructor và phương thức	Giải thích
check_input()	Phương thức đọc số nguyên từ bàn phím, đồng thời kiểm tra phải là số nguyên. Nếu không nhập gì hoặc nhập khoảng trắng thì kết thúc chương trình.
check_leap_year(int year)	Phương thức kiểm tra năm đó có phải năm nhuận không.
check_valid(int ngay, int thang, int nam)	Phương thức kiểm tra xem ngày tháng năm hợp lệ không.

➤ Các testcase của chương trình:

Bảng 3.3 Kết quả chương trình Bài 3

Chức năng	Input Output
Kiểm tra năm hợp lệ	<div> 29 2 2000 </div> 
Kiểm tra năm hợp lệ	<div> 29 2 2001 </div> 
Kiểm tra năm hợp lệ	<div> 29 2 1900 </div>

	<pre>Nhap ngay, thang, nam: 29 2 1900 Ngay thang nam khong hop le Nhap Enter de thoat chuong trinh</pre>
Kiểm tra năm hợp lệ	<pre>-10 2 2222 Nhap ngay, thang, nam: -10 2 2222 Ngay thang nam khong hop le Nhap Enter de thoat chuong trinh</pre>
Thoát	<pre>Nhap ngay, thang, nam: Thoat chuong trinh</pre>
Trường hợp gây lỗi	<pre>Abc -> throw exception: Dừng chương trình Nhap ngay, thang, nam: Abc The input string 'Abc' was not in a correct format. </pre>

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

Mô tả đầu vào và đầu ra của bài toán:

- Input: Nhập vào 2 số nguyên dương tháng năm. Mỗi số trên 1 dòng.
- Output: Trả về số ngày của tháng trong năm.

➤ Nội dung code của Bài 4

Bảng 4.1 Nội dung code của chương trình Bài 4

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

namespace BTTH1_BT4 {
    public class Program {
        //Ngày của các tháng trong năm bắt đầu từ index 1
        private static int[] valid_day = { -1, 31, 28, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31 };

        //Kiểm tra năm nhuận
        public static bool check_leap_year(int year) {
            return (year % 400 == 0) || (year % 4 == 0 && year % 100 !=
0);
        }

        //Kiểm tra số đầu vào
        public static int check_input() {
            int x;
            string tmp = Console.ReadLine();
            if (String.IsNullOrEmpty(tmp)) {
                Console.WriteLine("Thoát chương trình");
                Environment.Exit(0);
            }
            try {
                x = int.Parse(tmp);
            }
            catch (Exception e) {
                Console.WriteLine(e.Message);
                throw new Exception(e.Message);
            }
            return x;
        }

        //Ngày trong tháng
        public static int return_day(int thang, int nam) {
            if (thang < 1 || thang > 12){
                Console.WriteLine("Không hợp lệ");
                throw new Exception("Thang không hợp lệ");
            }
        }
    }
}
```

```

    }
    int leap = 0;
    if(thang == 2 && check_leap_year(nam) == true) {
        leap = 1;
    }

    return valid_day[thang] + leap;
}

public static void Main() {
    //Nhap thang nam
    while (true) {
        Console.WriteLine("Nhap thang, nam:");
        int thang = check_input();
        int nam = check_input();

        Console.WriteLine("So ngay cua thang " + thang + " la: "
+ return_day(thang, nam));
        Console.WriteLine("Nhap Enter de thoat chuong trinh");
    }
}
}
}

```

➤ Giải thích code của chương trình Bài 4

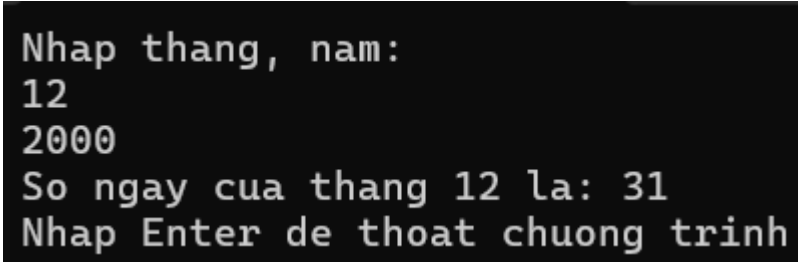
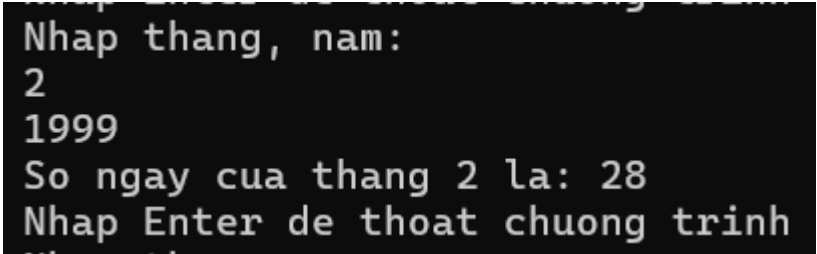
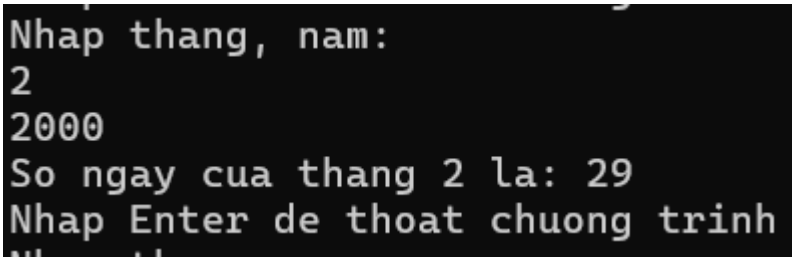
Bảng 4.2 Giải thích code của chương trình Bài 4

Constructor và phương thức	Giải thích
check_input()	Phương thức đọc số nguyên từ bàn phím, đồng thời kiểm tra phải là số nguyên. Nếu không nhập gì hoặc nhập khoảng trắng thì kết thúc chương trình.
check_leap_year(int year)	Phương thức kiểm tra năm đó có phải năm nhuận không.
return_day(int thang, int nam)	Phương thức trả về số ngày trong tháng, nếu không hợp lệ sẽ gây lỗi dừng chương trình.

IT008 – LẬP TRÌNH TRỰC QUAN

➤ Các testcase của chương trình

Bảng 4.3 Kết quả chương trình Bài 4

Chức năng	Input Output
Trả về số ngày của tháng	12 2000 
Trả về số ngày của tháng	2 1999 
Trả về số ngày của tháng	2 2000 

Thoát	<pre>Nhap thang, nam: Thoat chương trình</pre>
Trường hợp gây lỗi	<p>❖ Lỗi nhập tháng -> throw exception: Dừng chương trình</p> <pre>-2 2000</pre> <pre>Nhap thang, nam: -2 2000 Khong hop le </pre> <p>❖ Lỗi input -> throw exception: Dừng chương trình</p> <pre>Abc</pre> <pre>Nhap thang, nam: Abc The input string 'Abc' was not in a correct format. </pre>

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: Nhập vào 3 số nguyên dương ngày tháng năm. Mỗi số trên 1 dòng.
- Output: Trả về thứ của ngày trong tuần.

➤ Nội dung code của Bài 5

Bảng 5.1 Nội dung code của chương trình Bài 5

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

namespace BTTH1_BT5 {
    class Program {
        private static int[] valid_day = { -1, 31, 28, 31, 30, 31, 30,
31, 31, 30, 31, 30, 31 };

        //Kiểm tra năm nhuận
        public static bool check_leap_year(int year) {
            return (year % 400 == 0) || (year % 4 == 0 && year % 100 !=
0);
        }

        //Kiểm tra số đầu vào
        public static int check_input() {
            int x;
            string tmp = Console.ReadLine();
            if (String.IsNullOrEmpty(tmp)) {
                Console.WriteLine("Thoát chương trình");
                Environment.Exit(1);
            }
            try {
                x = int.Parse(tmp);
            }
            catch (Exception e) {
                Console.WriteLine(e.Message);
                throw new Exception(e.Message);
            }
            return x;
        }

        //Kiểm tra thời gian hợp lệ
        public static bool check_valid(int ngay, int thang, int nam) {
            if (ngay < 1 || thang < 1 || thang > 12 || nam < 1)
                return false;
            int leap = 0;
            if (thang == 2 && check_leap_year(nam) == true) {
                leap = 1;
            }
        }
    }
}
```

```

        return ngay <= (valid_day[thang] + leap);
    }

    //Tính tổng số ngày
    private static int total_day(int ngay, int thang, int nam) {
        if (check_valid(ngay, thang, nam) == false) {
            Console.WriteLine("Không hợp lệ");
            Environment.Exit(1);
        }

        int total = 0;
        for(int i = 1; i < nam; ++i) {
            if (check_leap_year(i)) {
                total++;
            }
            total += 365;
        }
        for (int i = 1; i < thang; ++i) {
            if(i == 2 && check_leap_year(nam) == true)
                total++;
            total += valid_day[i];
        }
        return total + ngay;
    }

    //Tra về ngày trong tuần
    private static string return_ngay(int ngay, int thang, int nam) {
        int thu = (total_day(ngay, thang, nam) - 1) % 7;
        switch (thu) {
            case 0:
                return ("Thu hai");
            case 1:
                return ("Thu ba");
            case 2:
                return ("Thu tu");
            case 3:
                return ("Thu nam");
            case 4:
                return ("Thu sau");
            case 5:
                return ("Thu bay");
            default:
                return ("Chu nhat");
        }
    }

    public static void Main() {
        //Nhap ngay thang nam
        while (true) {
            Console.WriteLine("Nhap ngay, thang, nam:");
            int ngay = check_input();
            int thang = check_input();
            int nam = check_input();

            Console.WriteLine("{0}/{1}/{2} là thu: {3}", ngay, thang,
nam, return_ngay(ngay, thang, nam));
            Console.WriteLine("Nhap Enter để thoát chương trình");
        }
    }

```



```

    }
}

```

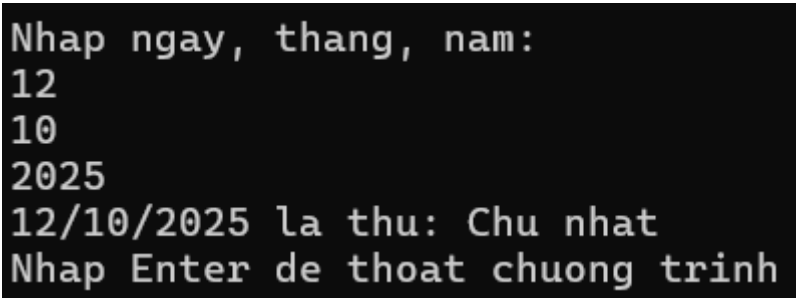
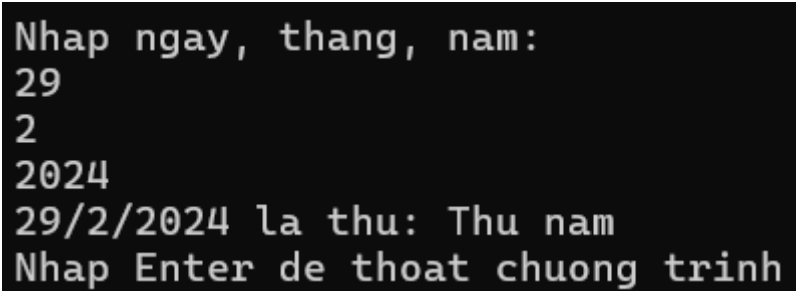
➤ Giải thích code của chương trình bài 5

Bảng 5.2 Giải thích code của chương trình Bài 5

Constructor và phương thức	Giải thích
check_input()	Phương thức đọc số nguyên từ bàn phím, đồng thời kiểm tra phải là số nguyên. Nếu không nhập gì hoặc nhập khoảng trắng thì kết thúc chương trình.
check_leap_year(int year)	Phương thức kiểm tra năm đó có phải năm nhuận không.
check_valid(int ngay, int thang, int nam)	Phương thức kiểm tra xem ngày tháng năm hợp lệ không.
total_day(int ngay, int thang, int nam)	Phương thức tính tổng số ngày bắt đầu từ 1/1/1 đến ngày nhập vào, nếu không hợp lệ dừng chương trình.
retrun_day(int ngay, int thang, int nam)	Phương thức trả về số ngày trong tuần.

Các testcase của chương trình:

Bảng 5.3 Kết quả chương trình Bài 5

Chức năng	Input Output
Tìm thứ trong tuần	12 10 2025 
Tìm thứ trong tuần	29 2 2024 
Tìm thứ trong tuần	28 2 2025

	<pre>Nhap ngay, thang, nam: 28 2 2025 28/2/2025 la thu: Thu sau Nhap Enter de thoat chuong trinh</pre>
Thoát	<pre>Nhap ngay, thang, nam: Thoat chuong trinh</pre>
Trường hợp gây lỗi	<p>❖ Nhập ngày tháng năm không hợp lệ -> Dừng chương trình</p> <pre>29 2 2025</pre> <pre>Nhap ngay, thang, nam: 29 2 2025 Khong hop le</pre> <p>❖ Nhập chuỗi kí tự -> throw exception: Dừng chương trình</p> <pre>Abc</pre> <pre>Nhap ngay, thang, nam: Abc The input string 'Abc' was not in a correct format.</pre>

6. Viết chương trình cho phép tạo ma trận chứa các số nguyên ngẫu nhiên gồm n dòng, m cột.

- a. Xuất ma trận
- b. Tìm phần tử lớn nhất/nhỏ nhất
- c. Tìm dòng có tổng lớn nhất
- d. Tính tổng các số không phải là số nguyên tố
- e. Xóa dòng thứ k trong ma trận
- f. Xóa cột chứa phần tử lớn nhất trong ma trận

Mô tả đầu vào và đầu ra của bài toán:

- Input: Một mảng 2 chiều gồm M x N số nguyên ngẫu nhiên.
- Output: Trả về ma trận, phần tử lớn nhất/nhỏ nhất, dòng có tổng lớn nhất, tổng các số không phải là số nguyên tố, xóa dòng thứ k của ma trận, xóa cột chứa phần tử lớn nhất của ma trận.

➤ Nội dung code của Bài 6

Bảng 6.1 Nội dung code của chương trình Bài 6

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Linq;
using System.Numerics;
using System.Reflection.Metadata;
using System.Text;
using System.Threading.Tasks;

namespace BTTH1_BT6 {
    public class Matrix {
        const int MAX = 101;
        const int MIN = -100;
        private int rows, cols;
        public int Size_Row {
            get { return rows; }
        }
        public int Size_Col {
            get { return cols; }
        }
    }

    private int[][] matrix;

    //Constructor có tham số tạo ma trận ngẫu nhiên
    public Matrix(int rows, int cols) {
        this.rows = rows;
        this.cols = cols;
    }
}
```

```

Random rand = new Random();
matrix = new int[rows][];

for (int i = 0; i < rows; ++i) {
    matrix[i] = new int[cols];
    for (int j = 0; j < cols; ++j) {
        matrix[i][j] = rand.Next(MIN, MAX);
    }
}

//In ma tran
public void Print() {
    const int cellWidth = 5;
    Console.WriteLine("\t===== MATRIX =====");
    Console.WriteLine("+");
    for (int j = 0; j < cols; j++) {
        Console.Write(new string('-', cellWidth));
    }
    Console.WriteLine("+");

    for (int i = 0; i < rows; i++) {
        Console.WriteLine("|");
        for (int j = 0; j < cols; j++) {
            Console.Write($"{matrix[i][j],cellWidth}");
        }
        Console.WriteLine("|");
    }

    Console.WriteLine("+");
    for (int j = 0; j < cols; j++) {
        Console.Write(new string('-', cellWidth));
    }
    Console.WriteLine("+");
}

//Tim vi tri cua phan tu lon/be
private void Find(string op, out int row, out int col) {
    int tmp = matrix[0][0];
    row = col = 0;
    switch (op) {
        case "MAX":
            for (int i = 0; i < Size_Row; ++i) {
                for (int j = 0; j < Size_Col; ++j) {
                    if (matrix[i][j] > tmp) {
                        tmp = matrix[i][j];
                        row = i;
                        col = j;
                    }
                }
            }
            break;
        case "MIN":
            for (int i = 0; i < Size_Row; ++i) {
                for (int j = 0; j < Size_Col; ++j) {
                    if (matrix[i][j] < tmp) {
                        tmp = matrix[i][j];
                        row = i;
                        col = j;
                    }
                }
            }
            break;
    }
}

```

```

        }
    }
    }
    break;
}

//Tim phan tu lon nhat
public int find_Max() {
    int row, col;
    this.Find("MAX", out row, out col);
    return matrix[row][col];
}

//Tim phan tu be nhat
public int find_Min() {
    int row, col;
    this.Find("MIN", out row, out col);
    return matrix[row][col];
}

//Tong cac phan tu tren dong
private int sum_Row(int[] array) {
    int sum = 0;
    foreach(int x in array) {
        sum += x;
    }
    return sum;
}

//Tim dong co tong lon nhat
public int find_Row() {
    int res = 0;
    int sum = 0;
    for(int i = 0; i < rows; ++i) {
        int tmp = sum_Row(matrix[i]);
        if(sum < tmp) {
            sum = tmp;
            res = i;
        }
    }
    return res;
}

//Kiem tra so nguyen to
public bool check_Prime(int x) {
    if (x < 1)
        return false;
    if (x == 2)
        return true;
    for(int i = 2; i <= (int)Math.Sqrt(x); ++i) {
        if(x % i == 0) {
            return false;
        }
    }
    return true;
}

//Tong cac so khong phai la so nguyen to
public int sum_not_Prime() {
    int sum = 0;

```

```

        for (int i = 0; i < rows; ++i) {
            for (int j = 0; j < cols; ++j) {
                if (!check_Prime(matrix[i][j])) {
                    sum += matrix[i][j];
                }
            }
        }
        return sum;
    }

    //Xoa dong thu k trong ma tran
    public void delete_Row(int k) {
        if (k < 0 || k >= rows) {
            Console.WriteLine("Index out of range");
            return;
        }
        this.Delete(k, -1);
    }

    //Xoa cot chua phan tu lon nhat trong ma tran
    public void delete_Col_containt_MAX() {
        int row, col;
        this.Find("MAX", out row, out col);
        this.Delete(-1, col);
    }

    //Xoa dong/cot
    public void Delete(int l_row, int l_col) {
        int newRowCount = rows - (l_row >= 0 ? 1 : 0);
        int newColCount = cols - (l_col >= 0 ? 1 : 0);

        int[][] newMatrix = new int[newRowCount][];
        int newRow = 0;

        for (int i = 0; i < rows; i++) {
            if (i == l_row)
                continue;
            newMatrix[newRow] = new int[newColCount];
            int newCol = 0;
            for (int j = 0; j < cols; j++) {
                if (j == l_col)
                    continue;
                newMatrix[newRow][newCol++] = matrix[i][j];
            }
            newRow++;
        }

        rows = newRowCount;
        cols = newColCount;
        matrix = newMatrix;
    }

}

public class Program {
    //Kiem tra so dau vao
    public static int read_int() {
        int n;
        try {
            n = int.Parse(Console.ReadLine());
            if (n <= 0) {

```

```

        throw new ArgumentOutOfRangeException();
    }
}
catch (Exception e) {
    Console.WriteLine(e.Message);
    throw new Exception(e.Message);
}
return n;
}
public static void Main() {
    //Nhap so phan tu cua mang va tao mang
    Console.WriteLine("Nhap so phan tu mang MxN (M, N > 0): ");
    int m = read_int();
    int n = read_int();
    Matrix matrix = new Matrix(m, n);

    int choice;
    do {
        //In menu
        Console.WriteLine("\n===== MENU =====");
        Console.WriteLine("1. Thoat");
        Console.WriteLine("2. In ma tran");
        Console.WriteLine("3. Tim phan tu lon nhat");
        Console.WriteLine("4. Tim phan tu nho nhat");
        Console.WriteLine("5. Dong co tong lon nhat");
        Console.WriteLine("6. Tinh tong cac so khong phai la so
nguyen to");

        Console.WriteLine("7. Xoa dong thu k trong ma tran");
        Console.WriteLine("8. Xoa cot chua phan tu lon nhat");
        Console.WriteLine("Chon chuc nang: ");

        choice = read_int();

        switch (choice) {
            case 1:
                Console.WriteLine("Ket thuc chuong trinh");
                break;
            case 2:
                Console.WriteLine("In ma tran:");
                matrix.Print();
                break;
            case 3:
                Console.WriteLine("Phan tu lon nhat la: " +
matrix.find_Max());
                break;
            case 4:
                Console.WriteLine("Phan tu nho nhat la: " +
matrix.find_Min());
                break;
            case 5:
                Console.WriteLine("Dong co tong lon nhat la: " +
matrix.find_Row());
                break;
            case 6:
                Console.WriteLine("Tong cac so khong la so nguyen
to: " + matrix.sum_not_Prime());
                break;
            case 7:
                Console.WriteLine("Xoa dong thu: ");

```



```

        int k = int.Parse(Console.ReadLine());
        matrix.delete_Row(k);
        break;
    case 8:
        Console.WriteLine("Xoa cot chua phan tu lon nhat
trong ma tran");
        matrix.delete_Col_containt_MAX();
        break;
    default:
        Console.WriteLine("Khong co chuc nang nam ngoai
MENU");
        throw new ArgumentOutOfRangeException();
    }
} while (choice != 1);
}
}
}

```

➤ Giải thích code của chương trình Bài 6

Bảng 6.2 Giải thích code của chương trình Bài 6

Constructor và phương thức	Giải thích
Matrix(int rows, int cols)	Constructor khởi tạo ma trận gồm M x N phần tử ngẫu nhiên.
Print()	In ma trận ra màn hình.
Find(string op, out int row, out int col)	Phương thức tìm vị trí hàng và cột của phần tử lớn nhất/nhỏ nhất dựa vào op (op = { “MAX”, “MIN” }).
find_Max()	Phương thức trả về phần tử lớn nhất.
find_Min()	Phương thức trả về phần tử nhỏ nhất.
sum_Row(int[] array)	Phương thức tính tổng các phần tử trên mảng 1 chiều.
find_Row()	Phương thức tìm dòng tổng lớn nhất.
check_Prime(int x)	Phương thức kiểm tra số đó có phải là số nguyên tố không.
sum_not_Prime()	Phương thức tính tổng các phần tử không phải là số nguyên tố trong mảng.
delete_Row(int k)	Phương thức xóa dòng thứ k trong ma trận dựa vào phương thức Delete (Tính từ 0).
delete_Col_containt_MAX()	Phương thức xóa cột chứa phần tử lớn nhất dựa vào phương thức Delete.
Delete(int l_row, int l_col)	Phương thức xóa dòng/cột

IT008 – LẬP TRÌNH TRỰC QUAN

<code>read_int()</code>	Phương thức đọc số nguyên từ bàn phím, đồng thời kiểm tra lớn hơn 0.
-------------------------	--

➤ Các testcase của chương trình:

Bảng 6.3 Kết quả chương trình Bài 6

Chức năng	Input Output
Tạo ma trận 5x5 và in ra	<p>5</p> <p>5</p> <p>2</p> <pre> Nhap so phan tu mang MxN (M, N > 0): 5 5 ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 2 In ma tran: ===== MATRIX ===== +-----+ 23 33 -94 -14 80 -86 32 48 84 -58 91 17 -50 -36 -87 -73 -84 -87 -8 51 -44 -64 4 -95 -53 +-----+ </pre>
Tìm phần tử lớn nhất	3

	<pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 3 Phan tu lon nhat la: 91 </pre>
<p>Tim phần tử nhỏ nhất</p>	<p>4</p> <pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 4 Phan tu nho nhat la: -95 </pre>
<p>Tim dòng có tổng lớn nhất</p>	<p>5</p> <pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 5 Dong co tong lon nhat la: 0 </pre>

Tính tổng các số không phải là số nguyên tố	6 <pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 6 Tong cac so khong la so nguyen to: -510 </pre>
Xóa dòng thứ 2 trong ma trận và in ra	7 2 2

	<pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 7 Xoa dong thu: 2 ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 2 In ma tran: ===== MATRIX ===== +-----+ 23 33 -94 -14 80 -86 32 48 84 -58 -73 -84 -87 -8 51 -44 -64 4 -95 -53 +-----+ </pre>
Tìm phần tử lớn nhất và xóa cột chứa nó rồi in ra	3 8 2

	<pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 3 Phan tu lon nhat la: 84 ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 8 Xoa cot chua phan tu lon nhat trong ma tran ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 2 In ma tran: ===== MATRIX ===== +-----+ 23 33 -94 80 -86 32 48 -58 -73 -84 -87 51 -44 -64 4 -53 +-----+ </pre>
Thoát	0

	<pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 1 Ket thuc chuong trinh </pre>
Trường hợp gây lỗi	<p>❖ Tạo ma trận 0 x 0 -> throw exception: Dừng chương trình</p> <p>0</p> <pre> System.Exception: 'Specified argument was out of the range of valid values.' </pre> <p>❖ Chọn chức năng không tồn tại -> throw Exception: Dừng chương trình</p> <p>10</p> <pre> ===== MENU ===== 1. Thoat 2. In ma tran 3. Tim phan tu lon nhat 4. Tim phan tu nho nhat 5. Dong co tong lon nhat 6. Tinh tong cac so khong phai la so nguyen to 7. Xoa dong thu k trong ma tran 8. Xoa cot chua phan tu lon nhat Chon chuc nang: 10 Khong co chuc nang nam ngoai MENU </pre> <p>❖ Xóa dòng vượt quá số dòng hiện có -> throw Exception: Dừng chương trình</p> <p>7</p> <p>100</p>


```
===== MENU =====
1. Thoat
2. In ma tran
3. Tim phan tu lon nhat
4. Tim phan tu nho nhat
5. Dong co tong lon nhat
6. Tinh tong cac so khong phai la so nguyen to
7. Xoa dong thu k trong ma tran
8. Xoa cot chua phan tu lon nhat
Chon chuc nang: 7
Xoa dong thu:
100
Index out of range
|
```

❖ Nhập chuỗi -> throw Exception: Dừng chương trình

Abc

```
Nhap so phan tu mang MxN (M, N > 0):
Abc
The input string 'Abc' was not in a correct format.
|
```

IT008 – LẬP TRÌNH TRỰC QUAN

LinkCode: <https://github.com/trungnha-uit/IT008.git>