**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 Quí

Sinh viên thực hiện: Nguyễn Huỳnh Anh Trung

MSSV: 24521886

🙡🙢 Tp. Hồ Chí Minh, 10/2025 🙠🙣

**NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

*……., ngày……...tháng……năm 2025*

**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](#_Toc211202196)

[2. Viết chương trình nhập số nguyên dương n, tính tổng các số nguyên tố < n và xuất kết quả ra màn hình. 12](#_Toc211202197)

[3. Viết chương trình nhập ngày tháng năm, cho biết ngày tháng năm đó có hợp lệ không? 15](#_Toc211202198)

[4. Viết chương trình nhập vào tháng và năm. In ra số ngày của tháng đó…… 19](#_Toc211202199)

[5. Viết chương trình nhập vào ngày tháng năm, cho biết thứ trong tuần… 23](#_Toc211202200)

[6. Viết chương trình cho phép tạo ma trận chứa các số nguyên ngẫu nhiên gồm n dòng, m cột. 28](#_Toc211202201)

**DANH MỤC BẢNG**

[Bảng 1.1 Nội dung code của chương trình Bài 1 5](#_Toc211202178)

[Bảng 1.2 Giải thích code của chương trình Bài 1 8](#_Toc211202179)

[Bảng 1.3 Kết quả chương trình Bài 1 9](#_Toc211202180)

[Bảng 2.1 Nội dung code của chương trình Bài 2 12](#_Toc211202181)

[Bảng 2.2 Giải thích code của chương trình Bài 2 13](#_Toc211202182)

[Bảng 2.3 Kết quả chương trình Bài 2 14](#_Toc211202183)

[Bảng 3.1 Nội dung code của chương trình Bài 3 15](#_Toc211202184)

[Bảng 3.2 Giải thích code của chương trình Bài 3 16](#_Toc211202185)

[Bảng 3.3 Kết quả chương trình Bài 3 17](#_Toc211202186)

[Bảng 4.1 Nội dung code của chương trình Bài 4 19](#_Toc211202187)

[Bảng 4.2 Giải thích code của chương trình Bài 4 20](#_Toc211202188)

[Bảng 4.3 Kết quả chương trình Bài 4 21](#_Toc211202189)

[Bảng 5.1 Nội dung code của chương trình Bài 5 23](#_Toc211202190)

[Bảng 5.2 Giải thích code của chương trình Bài 5 25](#_Toc211202191)

[Bảng 5.3 Kết quả chương trình Bài 5 26](#_Toc211202192)

[Bảng 6.1 Nội dung code của chương trình Bài 6 28](#_Toc211202193)

[Bảng 6.2 Giải thích code của chương trình Bài 6 33](#_Toc211202194)

[Bảng 6.3 Kết quả chương trình Bài 6 35](#_Toc211202195)

**NỘI DUNG BÀI LÀM**

# 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; }  }  //Contructor co tham so dau vao la so phan tu mang  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);  }  }  //Tong cac so le trong mang  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 {  //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 (n > 0): ");  int n = read\_int();  RandomArray rArray = new RandomArray(n);  int choice;  do {  //In menu  Console.WriteLine("\n===== MENU =====");  Console.WriteLine("1. Thoat");  Console.WriteLine("2. In mang");  Console.WriteLine("3. Tinh tong cac so le");  Console.WriteLine("4. Dem so nguyen to");  Console.WriteLine("5. Tim so chinh phuong nho nhat");  Console.Write("Chon chuc nang: ");  choice = read\_int();  switch (choice) {  case 1:  Console.WriteLine("Ket thuc chuong trinh");  break;  case 2:  Console.WriteLine("Mang cac so nguyen:");  rArray.Print();  break;  case 3:  Console.WriteLine("Tong cac so le la: " + rArray.sum\_Odds());  break;  case 4:  Console.WriteLine("So nguyen to trong mang la: " + rArray.count\_Prime());  break;  case 5:  Console.WriteLine("So chinh phuong nho nhat la: " + rArray.find\_min\_Square());  break;  default:  Console.WriteLine("Khong co chuc nang nam ngoai 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 | 10  2 |
| Tính tổng các số lẻ | 3 |
| Đếm các số nguyên tố | 4 |
| Tìm số chính phương nhỏ nhất và thoát | 5  1 |
| Trường hợp gây lỗi | * Tạo mảng có 0 phần tử -> throw exception:Dừng chương trình * Chọn chức năng ngoài MENU -> Dừng chương trình |

# 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 {  //Kiem tra so dau vao  public static int check\_input() {  int x;  try {  x = int.Parse(Console.ReadLine());  if (x != -1 && x < 0)  throw new ArgumentOutOfRangeException("Phai la so duong");  }catch(Exception e) {  Console.WriteLine(e.Message);  throw new Exception(e.Message);  }  return x;  }  //Kiem tra so nguyen to  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;  }  //Tong cac so nguyen to nho hon 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() {  //Nhap so nguyen duong n  int n;  do {  Console.WriteLine("Nhap mot so nguyen duong:");  n = check\_input();  if (n == -1)  break;  Console.WriteLine("Tong cac so nguyen to nho hon n la: " + sum\_Prime(n));  Console.WriteLine("Nhap -1 de thoat.");  } 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. |

* 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 |

# 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 {  //Ngay cua cac thang trong nam bat dau tu index 1  private static int[] valid\_day = { -1, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  //Kiem tra nam nhuan  public static bool check\_leap\_year(int year) {  return (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0);  }  //Kiem tra so dau vao  public static int check\_input() {  int x;  string tmp = Console.ReadLine();  if (String.IsNullOrWhiteSpace(tmp)) {  Console.WriteLine("Thoat chuong trinh");  Environment.Exit(0);  }  try {  x = int.Parse(tmp);  }  catch (Exception e) {  Console.WriteLine(e.Message);  throw new Exception(e.Message);  }  return x;  }  //Kiem tra thoi gian hop le  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("Ngay thang nam hop le");  } else {  Console.WriteLine("Ngay thang nam khong hop le");  }  Console.WriteLine("Nhap Enter de thoat chuong trinh");  }  }  }  } |

* 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 | 29  2  2000 |
| Kiểm tra năm hợp lệ | 29  2  2001 |
| Kiểm tra năm hợp lệ | 29  2  1900 |
| Kiểm tra năm hợp lệ | -10  2  2222 |
| Thoát |  |
| Trường hợp gây lỗi | Abc -> throw exception: Dừng chương trình |

# 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 {  //Ngay cua cac thang trong nam bat dau tu index 1  private static int[] valid\_day = { -1, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  //Kiem tra nam nhuan  public static bool check\_leap\_year(int year) {  return (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0);  }  //Kiem tra so dau vao  public static int check\_input() {  int x;  string tmp = Console.ReadLine();  if (String.IsNullOrWhiteSpace(tmp)) {  Console.WriteLine("Thoat chuong trinh");  Environment.Exit(0);  }  try {  x = int.Parse(tmp);  }  catch (Exception e) {  Console.WriteLine(e.Message);  throw new Exception(e.Message);  }  return x;  }  //Ngay trong thang  public static int return\_day(int thang, int nam) {  if (thang < 1 || thang > 12){  Console.WriteLine("Khong hop le");  throw new Exception("Thang khong hop le");  }  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. |

* 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 |  |
| Trường hợp gây lỗi | * Lỗi nhập tháng -> throw exception: Dừng chương trình   -2  2000     * Lỗi input -> throw exception: Dừng chương trình   Abc |

# 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 };  //Kiem tra nam nhuan  public static bool check\_leap\_year(int year) {  return (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0);  }  //Kiem tra so dau vao  public static int check\_input() {  int x;  string tmp = Console.ReadLine();  if (String.IsNullOrWhiteSpace(tmp)) {  Console.WriteLine("Thoat chuong trinh");  Environment.Exit(1);  }  try {  x = int.Parse(tmp);  }  catch (Exception e) {  Console.WriteLine(e.Message);  throw new Exception(e.Message);  }  return x;  }  //Kiem tra thoi gian hop le  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);  }  //Tinh tong so ngay  private static int total\_day(int ngay, int thang, int nam) {  if (check\_valid(ngay, thang, nam) == false) {  Console.WriteLine("Khong hop le");  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 ve ngay trong tuan  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} la thu: {3}", ngay, thang, nam, return\_ngay(ngay, thang, nam));  Console.WriteLine("Nhap Enter de thoat chuong trinh");  }  }  }  } |

* 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 |
| Thoát |  |
| Trường hợp gây lỗi | * Nhập ngày tháng năm không hợp lệ -> Dừng chương trình   29  2  2025     * Nhập chuỗi kí tự -> throw exception: Dừng chương trình   Abc |

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

1. **Xuất ma trận**
2. **Tìm phần tử lớn nhất/nhỏ nhất**
3. **Tìm dòng có tổng lớn nhất**
4. **Tính tổng các số không phải là số nguyên tố**
5. **Xóa dòng thứ k trong ma trận**
6. **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 co tham so tao ma tran ngau nhien  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.Write("+");  for (int j = 0; j < cols; j++) {  Console.Write(new string('-', cellWidth));  }  Console.WriteLine("+");  for (int i = 0; i < rows; i++) {  Console.Write("|");  for (int j = 0; j < cols; j++) {  Console.Write($"{matrix[i][j],cellWidth}");  }  Console.WriteLine("|");  }  Console.Write("+");  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;  }  }  //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.Write("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 |
| read\_int() | 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 | 5  5  2 |
| Tìm phần tử lớn nhất | 3 |
| Tìm phần tử nhỏ nhất | 4 |
| Tìm dòng có tổng lớn nhất | 5 |
| Tính tổng các số không phải là số nguyên tố | 6 |
| Xóa dòng thứ 2 trong ma trận và in ra | 7  2  2 |
| Tìm phần tử lớn nhất và xóa cột chứa nó rồi in ra | 3  8  2 |
| Thoát | 0 |
| Trường hợp gây lỗi | * Tạo ma trận 0 x 0 -> throw exception: Dừng chương trình   0     * Chọn chức năng không tồn tại -> throw Exception: Dừng chương trình   10     * Xóa dòng vượt quá số dòng hiện có -> throw Exception: Dừng chương trình   7  100     * Nhập chuỗi -> throw Exception: Dừng chương trình   Abc |

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