**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 2**

GVHD: Nguyễn Ngọc Quí

Sinh viên thực hiện: Nguyễn Gia Hưng – 24520604

🙡🙢 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

[Bài 1. Viết chương trình nhập vào tháng và năm, bạn in ra lịch của tháng đó như hình 6](#_Toc211937036)

[Bài 2. Viết chương trình yêu cầu người dùng nhập đường dẫn thư mục, xuất tất cả tên tập tin và thư mục con (tương tự lệnh DIR trong MS DOS) 7](#_Toc211937037)

[Bài 3. Bài 3 7](#_Toc211937038)

[Bài 4. Bài 3 8](#_Toc211937039)

[Bài 5. Bài 3 8](#_Toc211937040)

DANH MỤC BẢNG

[Bảng 1.1. Nội dung code bài 1 6](#_Toc211936665)

[Bảng 1.2. Các testcase bài 1 7](#_Toc211936666)

[Bảng 2.1. Nội dung code bài 2 7](#_Toc211936667)

[Bảng 2.2. Các testcase bài 2 7](#_Toc211936668)

[Bảng 3.1. Nội dung code bài 3 7](#_Toc211936669)

[Bảng 3.2. Các testcase bài 3 8](#_Toc211936670)

[Bảng 4.1. Nội dung code bài 4 8](#_Toc211936671)

[Bảng 4.2. Các testcase bài 4 8](#_Toc211936672)

[Bảng 5.1. Nội dung code bài 5 8](#_Toc211936673)

[Bảng 5.2. Các testcase bài 5 8](#_Toc211936674)

DANH MỤC HÌNH ẢNH

[Hình 1. 1 Class diagram của lớp NgayThangNam 6](#_Toc160803425)

[Hình 1. 2 xây dựng lớp, thuộc tính, phương thức. 6](#_Toc160803426)

NỘI DUNG BÀI LÀM

# Viết chương trình nhập vào tháng và năm, bạn in ra lịch của tháng đó như hình

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Hai số nguyên là tháng và năm. Nhập lại nếu không hợp lệ
* Output: Các ngày trong tháng đó sắp xếp theo lịch, từ chủ nhật đến thứ bảy

## Nội dung code của Bài 1

Bảng .. Nội dung code bài 1

|  |
| --- |
| using System;  namespace Bai01  {  internal class Program  {  static void Main(string[] args)  {  //1. Nhập tháng và năm  int month;  do  {  month = NhapSoNguyen("Nhap thang: ");  if (month < 1 || month > 12)  {  Console.Write("Thang khong hop le, nhap lai thang: ");  }  else break;  }  while (true);  int year;  do  {  year = NhapSoNguyen("Nhap nam: ");  if (year < 1 || year > 9999)  {  Console.Write("Nam khong hop le, vui long nhap lai: ");  }  else break;  } while (true);  //2. In ra lịch  Console.WriteLine("====================================================");  Console.WriteLine("Sun\tMon\tTue\tWed\tThu\tFri\tSat");  Console.WriteLine("====================================================");  //Lấy số ngày trong tháng  int daysInMonth = DateTime.DaysInMonth(year, month);    //Xử lý ngày 1 của tháng  DateTime firstDate = new DateTime(year, month, 1);  int firstDayOfWeek = (int)firstDate.DayOfWeek;  for(int i = 0; i < firstDayOfWeek; i++)  {  Console.Write("\t");  }  Console.Write(1);  int currentDayOfWeek = firstDayOfWeek;  for(int day = 2; day <= daysInMonth; day++)  {  if (currentDayOfWeek == 6)  {  Console.Write("\n" + day);  currentDayOfWeek = 0;  }  else  {  Console.Write("\t" + day);  currentDayOfWeek++;  }  }  }  //Hàm nhập số nguyên  public static int NhapSoNguyen(string thongBao)  {  int value;  bool ok;  do  {  Console.Write(thongBao);  ok = int.TryParse(Console.ReadLine(), out value);  if (!ok)  {  Console.WriteLine("Gia tri khong hop le, vui long nhap lai!");  }  } while (!ok);  return value;  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 1

|  |  |
| --- | --- |
| 1  2025 |  |
| 2  2020 |  |
|  |  |
|  |  |
|  |  |

# Viết chương trình yêu cầu người dùng nhập đường dẫn thư mục, xuất tất cả tên tập tin và thư mục con (tương tự lệnh DIR trong MS DOS)

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Nhập vào một chuỗi là đường dẫn thư mục. Chọn 1 để tiếp tục nhập đường dẫn tư vị trí trước đó hoặc 0 để thoát chương trình.
* Output: In ra tất cả thư mục con, file trong thư mục, nếu không tìm thấy hoặc đường dẫn không hợp lệ thì in ra “Khong thay thu muc”

## Nội dung code của Bài 2:

Bảng .. Nội dung code bài 2

|  |
| --- |
| using System;  using System.IO;  namespace Bai02  {  internal class Program  {  static void Main(string[] args)  {  int isContinue;  string oldDir = "";  string mainDir = "";  do  {  Console.Write("Nhap duong dan thu muc: " + mainDir);  //Đường dẫn tương đối từ vị trí trước đó  string dir = Console.ReadLine();  //Cập nhật đường dẫn chính  if (!string.IsNullOrWhiteSpace(dir))  {  oldDir = mainDir;  if (dir[dir.Length - 1] == '/')  {  mainDir += dir;  }  else  {  mainDir += dir + "/";  }  }    //Nếu tìm thấy thư mục  if (Directory.Exists(mainDir))  {  DirectoryInfo di = new DirectoryInfo(mainDir);  DirectoryInfo[] subDirs = di.GetDirectories();  FileInfo[] files = di.GetFiles();  foreach (DirectoryInfo dirInfo in subDirs)  {  Console.WriteLine("{0}\t<DIR>\t{1}",  dirInfo.LastWriteTime.ToString("dd/MM/yyyy HH:mm"),  dirInfo.Name);  }  foreach (FileInfo file in files)  {  Console.WriteLine("{0}\t{1}\t{2}",  file.LastWriteTime.ToString("dd/MM/yyyy HH:mm"),  file.Length,  file.Name);  }  }  else  {  Console.WriteLine("Khong thay thu muc");  mainDir = oldDir;  }  //Tiếp tục tìm hoặc thoát chương trình  Console.WriteLine("Tiep tuc?\n" +  "[1]: Yes\t[0]: No");  isContinue = int.Parse(Console.ReadLine());  }  while (isContinue > 0);  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 2

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

# Xây dựng chương trình có chức năng

**a. Nhập / xuất ma trận hai chiều các số nguyên**

**b. Tìm kiếm một phần tử trong ma trận**

**c. Xuất các phần tử là số nguyên tố**

**d. Cho biết dòng nào có nhiều số nguyên tố nhất**

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Nhập vào hai số nguyên là số dòng m và số cột n của ma trận. Chọn 1 để tạo tự động ma trận ngẫu nhiên hoặc chọn 0 và nhập mn số nguyên để tạo thành ma trận. Nhập số k cần tìm trong ma trận.
* Output: In ra ma trận đã tạo, in ra vị trí phần tử k cần tìm, in ra các phân tử là số nguyên tố, in ra dòng chứa nhiều số nguyên tố nhất và in ra dòng đó

## Nội dung code của Bài 3:

Bảng .. Nội dung code bài 3

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Bai03  {  internal class Program  {  static void Main(string[] args)  {  //1. Tạo, nhập và xuất ma trận  MaTran mat = new MaTran();  mat.Nhap();  mat.Print();  //2. Tìm số nguyên trong ma trận  int iToFind = NhapSoNguyen("\n2. Nhap so can tim: ");  mat.FindFirstIf((x) => (x == iToFind)).Print();  //3. Tìm các phần tử là số nguyên tố  Console.WriteLine("\n3. Cac phan tu la so nguyen to: ");  mat.Print((x) => IsPrime(x));  //4. Tìm các dòng có nhiều số nguyên tố nhất  List<int> rows = mat.MaxRowCountIf((x) => IsPrime(x));  Console.WriteLine("\n4. Dong co so nguyen to nhieu nhat:");  foreach (int r in rows)  {  Console.Write("- Dong " + (r + 1) + ":\t");  mat.PrintRow(r);  }  }    public delegate bool DieuKien(int i);  public class MaTran  {  private List<List<int>> mat;  public int Row  {  get  {  return mat.Count;  }  }  public int Col  {  get  {  if (Row != 0)  {  return mat[0].Count;  }  else return 0;  }  }  public MaTran(int r = 1, int c = 1, bool random = true, int maxVal = 100, int minVal = -100)  {  CreateMatrix(r, c, random, maxVal, minVal);  }  //Tạo ma trận  private void CreateMatrix(int r = 1, int c = 1, bool random = true, int maxVal = 100, int minVal = -100)  {  mat = new List<List<int>>();  if (r <= 0 || c < 1) return;  Random rand = new Random();  for (int i = 0; i < r; i++)  {  List<int> row = new List<int>();  for (int j = 0; j < c; j++)  {  row.Add(random ? rand.Next(minVal, maxVal + 1) : 0);  }  mat.Add(row);  }  }  public void Nhap()  {  Console.WriteLine("Tao ma tran:");  int dong = NhapSoNguyen("Dong: ");  int cot = NhapSoNguyen("Cot: ");  int choice = NhapSoNguyen("[1]: Tao tu dong\t"  + "[0]: Tao thu cong\n");  if (choice > 0)  {  CreateMatrix(dong, cot);  }  else  {  CreateMatrix(dong, cot, false);  for (int r = 0; r < dong; r++)  {  for (int c = 0; c < cot; c++)  {  Console.Write("Nhap [{0}][{1}]: ", r, c);  mat[r][c] = NhapSoNguyen("");  }  }  }  }  //In ma trận  public void Print()  {  for (int r = 0; r < Row; r++)  {  PrintRow(r);  }  }  //In dòng r trong ma trận  public void PrintRow(int r)  {  if (r < 0 || r >= Row) return;  foreach (int i in mat[r])  {  Console.Write(i + "\t");  }  Console.WriteLine();  }  //In ra các số thỏa điều kiện dk  public void Print(DieuKien dk)  {  foreach (var row in mat)  {  foreach (int i in row)  {  if (dk(i))  {  Console.Write(i + " ");  }  }  }  Console.WriteLine();  }  //Vị trị trong ma trận  public struct Position  {  public int row { get; private set; }  public int col { get; private set; }  public Position(int r, int c)  {  row = r; col = c;  }  public void Print()  {  Console.WriteLine("Dong {0} - cot {1}", row + 1, col + 1);  }  }  //Giá giá trị một số trong vị trí  public int GetValue(Position pos)  {  if (pos.row < 0 || pos.col < 0  || pos.row >= Row || pos.col >= Col)  {  return -1;  }  return mat[pos.row][pos.col];  }  //Vị trí đầu tiên nếu tìm thấy số thỏa điều kiện dk  public Position FindFirstIf(DieuKien dk)  {  for (int r = 0; r < Row; r++)  {  for (int c = 0; c < Col; c++)  {  if (dk(mat[r][c]))  {  return new Position(r, c);  }  }  }  Console.Write("Khong tim thay: ");  return new Position(-1, -1);  }  //Những dòng có nhiều số nhất thỏa điều kiện dk  public List<int> MaxRowCountIf(DieuKien dk)  {  List<int> result = new List<int>();  int maxCount = 0;  for (int r = 0; r < Row; r++)  {  int countRow = 0;  for (int c = 0; c < Col; c++)  {  if (dk(mat[r][c])) countRow++;  }  if (countRow > maxCount)  {  maxCount = countRow;  result.Clear();  result.Add(r);  }  else if (countRow == maxCount && countRow > 0)  {  result.Add(r);  }  }  return result;  }  }  //Nhập số nguyên  public static int NhapSoNguyen(string thongBao)  {  int value;  bool ok;  do  {  Console.Write(thongBao);  ok = int.TryParse(Console.ReadLine(), out value);  if (!ok)  {  Console.WriteLine("Gia tri khong hop le, vui long nhap lai!");  }  } while (!ok);  return value;  }  //Kiểm tra số nguyên tố  public static bool IsPrime(int x)  {  if (x < 2) return false;  int root = (int)Math.Sqrt(x);  for(int i = 2; i <= root; i++)  {  if (x % i == 0) return false;  }  return true;  }  }  } |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 3

|  |  |
| --- | --- |
| 2  2  1  abc  1 |  |
| 2  3  0  2  4  5  7  8  9  5 |  |
| 2  2  0  4  6  8  9  8 |  |
| 2  4  0  -3  1  2  3  5  10  11  12  11 |  |
| 1  1  0  7  7 |  |
| 3  3  1  5 |  |
| 3  4  0  2  3  4  5  6  7  8  9  10  11  13  17  13 |  |
| 2  3  0  2  3  5  7  11  13  9 |  |

# Xây dựng lớp phân số với toán tử +, -, \*, /. Viết chương trình nhập vào hai phân số, xuất ra các giá trị là tổng, hiệu, tích và thương của hai phân số này.

**Viết chương trình cho phép nhập vào một dãy các phân số:**

**- Tìm phần số lớn nhất**

**- Sắp xếp các phân số trong dãy tăng dần**

## Mô tả đầu vào và đầu ra của bài toán:

* Input: Nhập tử và mẫu hai phân số đầu tiên. Nhập số n và lần lượt nhập tử và mẫu của n phân số tiếp theo
* Output: In ra tổng, hiệu, tích, thương của hai phân số đầu tiên

## Nội dung code của Bài 4:

Bảng .. Nội dung code bài 4

|  |
| --- |
|  |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 4

|  |  |
| --- | --- |
| 1  2  -3  4  7  0  5  2  4  -3  -6  3  2  -5  7  5  -7  4  8 |  |
| 0  5  1  1  5  0  1  1  1  -1  1  2  1  -3  1 |  |
| 1  0  3  2  3  4  1  0  2  3  4  5  0  6  7  8 |  |
| 1  2  0  5  3  1  2  3  4  5  6 |  |

# Bài 3

## Mô tả đầu vào và đầu ra của bài toán:

* Input:
* Output:

## Nội dung code của Bài 5:

Bảng .. Nội dung code bài 5

|  |
| --- |
|  |

## Các Testcase của chương trình:

Bảng .. Các testcase bài 5

|  |  |
| --- | --- |
|  |  |

LinkCode: Github, drive