**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 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 cho phép tạo mảng một chiều gồm n số nguyên ngẫu nhiên. 6](#_Toc210890087)

[Bài 2. Viết chương trình nhập số nguyên dương n, tính tổng các số nguyên tố < n và xuất kết quả ra màn hình. 10](#_Toc210890088)

[Bài 3. Viết chương trình nhập ngày tháng năm, cho biết ngày tháng năm đó có hợp lệ không? 12](#_Toc210890089)

[Bài 4. Viết chương trình nhập vào tháng và năm. In ra số ngày của tháng đó. 16](#_Toc210890090)

[Bài 5. Viết chương trình nhập vào ngày tháng năm, cho biết thứ trong tuần. 18](#_Toc210890091)

[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: 21](#_Toc210890092)

**DANH MỤC BẢNG**

[Bảng 1. Nội dung code bài 1 6](#_Toc210681592)

[Bảng 2. Testcase bài 1 8](#_Toc210681593)

[Bảng 3. Nội cung code bài 2 10](#_Toc210681594)

[Bảng 4. Testcase bài 2 11](#_Toc210681595)

[Bảng 5. Nội dung code bài 3 12](#_Toc210681596)

[Bảng 6. Testcase bài 3 13](#_Toc210681597)

[Bảng 7. Nội dung code bài 4 16](#_Toc210681598)

[Bảng 8. Testcase bài 4 16](#_Toc210681599)

[Bảng 9. Nội dung code bài 5 18](#_Toc210681600)

[Bảng 10. Testcase bài 5 20](#_Toc210681601)

[Bảng 11. Nội dung code bài 6 21](#_Toc210681602)

[Bảng 12. Testcase bài 6 26](#_Toc210681603)

**DANH MỤC HÌNH ẢNH**

**NỘI** **DUNG BÀI LÀM**

# Bài 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 . Nội dung code bài 1

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Bai01  {  internal class Program  {  public delegate bool DieuKien(int i);  public class MangSoNguyen  {  List<int> array;  public int Size  {  get  {  return array.Count;  }  }  //Khởi tạo n số ngẫu nhiên  public MangSoNguyen(int n = 0)  {  CreateRandom(n);  }  private void CreateRandom(int n, int minVal = -100, int maxVal = 100)  {  Random rnd = new Random();  array = new List<int>();  for (int i = 0; i < n; i++)  {  int r = rnd.Next(minVal, maxVal + 1);  array.Add(r);  }  }  //Tính tổng các số thỏa điều kiện dk  public int Sum(DieuKien dk)  {  int sum = 0;  foreach (int x in array)  {  if (dk(x)) sum += x;  }  return sum;  }  //Đếm các số thỏa điều kiện dk  public int Count(DieuKien dk)  {  int count = 0;  foreach (int x in array)  {  if (dk(x)) count++;  }  return count;  }  //Tìm số nhỏ nhất thỏa điều kiện dk  public int FindMin(DieuKien dk)  {  bool Found = false;  int Result = -1;  foreach (int x in array)  {  if (dk(x))  {  if (!Found)  {  Result = x;  Found = true;  }  else  {  if (x < Result)  {  Result = x;  }  }  }  }  return Result;  }  //In ra mảng  public void Xuat()  {  foreach (int x in array)  {  Console.Write(x + " ");  }  }  }  //Kiểm tra số nguyên tố  private static bool IsPrime(int x)  {  if (x <= 1) return false;  int root = (int)Math.Sqrt(x);  for (int i = 2; i <= root; i++)  {  if (x % i == 0) return false;  }  return true;  }  //Kiểm tra số chính phương  private static bool IsSoChinhPhuong(int x)  {  int root = (int)Math.Sqrt(x);  return root \* root == x;  }  static void Main(string[] args)  {  //1. Nhập n và tạo mảng  Console.Write("1. Nhap n > 0: ");  int n = Convert.ToInt32(Console.ReadLine());  MangSoNguyen mang = new MangSoNguyen(n);  //2. In ra mảng  Console.WriteLine("2. Da tao mang");  mang.Xuat();  //3. Tổng các số lẻ  Console.WriteLine("\n3. Tong so le: " +  mang.Sum(x => (x % 2 == 1)));    //4. Đếm số nguyên tố  Console.WriteLine("4. Dem so nguyen to: " +  mang.Count(x => IsPrime(x)));    //5. Tìm số chính phương nhỏ nhất  Console.WriteLine("5. So chinh phuong nho nhat: " +  mang.FindMin(x => IsSoChinhPhuong(x)));  }  }  } |

Các Testcase của chương trình:

Bảng . Testcase bài 1

|  |  |
| --- | --- |
| 10 |  |
| 20 |  |
| 0 |  |
| –5 |  |

# 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: Một số nguyên dương n
* Output: Kết quả tính tổng các số nguyên tố < n

Nội dung code của Bài 2:

Bảng . Nội cung code bài 2

|  |
| --- |
| using System;  namespace Bai02  {  internal class Program  {  static void Main(string[] args)  {  //1. Nhập số nguyên dương  Console.Write("Nhap so nguyen n > 0: ");  int n = Convert.ToInt32(Console.ReadLine());  //2. In ra tổng số nguyên tố nhỏ hơn n  Console.WriteLine("Tong so nguyen to nho hon n: "  + TongSoNguyenToNhoHon(n));  }  //Kiểm tra số nguyên tố  private static bool IsPrime(int x)  {  if (x <= 1) return false;  int root = (int)Math.Sqrt(x);  for (int i = 2; i <= root; i++)  {  if (x % i == 0) return false;  }  return true;  }  //Tính tổng số nguyên tố bé hơn n  static int TongSoNguyenToNhoHon(int n)  {  int sum = 0;  for (int i = 2; i < n; i++)  {  if (IsPrime(i)) sum += i;  }  return sum;  }  }  } |

Các testcase của chương trình:

Bảng . Testcase bài 2

|  |  |
| --- | --- |
| 20 |  |
| 100000 |  |
| 0 |  |
| –10 |  |

# Bài 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: Ngày tháng năm theo định dạng dd-MM-yyyy
* Output: In ra “Ngay hop le” nếu là ngày hợp lệ, in ra “Ngay khong hop le” nếu ngày không hợp lệ

Nội dung code của Bài 3:

Bảng . Nội dung code bài 3

|  |
| --- |
| using System;  namespace Bai03  {  internal class Program  {  static void Main(string[] args)  {  //1. Tạo và nhập ngày  Day ngay = new Day();  Console.WriteLine("Nhap ngay can kiem tra:");  ngay.Nhap();  //2. Kiểm tra ngày hợp lệ  if (ngay.NgayHopLe)  {  Console.Write("Ngay hop le: ");  ngay.Xuat();  }  else  {  Console.WriteLine("Ngay khong hop le");  }  }  class Day  {  private int ngay, thang, nam;  //Nhập ngày tháng năm  public void Nhap()  {  Console.Write("Ngay: ");  ngay = Convert.ToInt32(Console.ReadLine());  Console.Write("Thang: ");  thang = Convert.ToInt32(Console.ReadLine());  Console.Write("Nam: ");  nam = Convert.ToInt32(Console.ReadLine());  }  //In ra ngày/tháng/năm  public void Xuat()  {  if (NgayHopLe)  {  Console.Write("{0}/{1}/{2}", ngay, thang, nam);  }  }  //Kiểm tra năm nhuận  private bool IsNamNhuan  {  get  {  return (nam % 4 == 0 && nam % 100 != 0) || (nam % 400 == 0);  }  }  //Kiểm tra ngày hợp lệ  public bool NgayHopLe  {  get  {  if (thang < 1 || thang > 12) return false;  if (ngay < 1) return false;  int maxNgay;  switch (thang)  {  case 1:  case 3:  case 5:  case 7:  case 8:  case 10:  case 12:  maxNgay = 31; break;  case 2:  if (IsNamNhuan) maxNgay = 29;  else maxNgay = 28;  break;  default:  maxNgay = 30; break;  }  if (ngay > maxNgay) return false;  return true;  }  }  }  }  } |

Các testcase của chương trình:

Bảng . Testcase bài 3

|  |  |
| --- | --- |
| 15 8 2024 |  |
| 1 1 2022 |  |
| 30 4 2025 |  |
| 31 7 2028 |  |
| 29 2 2020 |  |
| 0 5 2021 |  |
| 31 4 2012 |  |
| 19 0 2123 |  |
| 13 13 1313 |  |
| 29 2 1234 |  |
| 29 2 0 |  |

# Bài 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: 2 số nguyên tháng và năm, nhập lại tháng nhiều lần đến khi hợp lệ
* Output: Thông báo số ngày trong tháng vừa nhập

Nội dung code của Bài 4:

Bảng . Nội dung code bài 4

|  |
| --- |
| using System;  namespace Bai04  {  internal class Program  {  static void Main(string[] args)  {  //1. Tạo và nhập tháng, kiểm tra tháng  int month;  Console.Write("Nhap thang: ");  do  {  month = int.Parse(Console.ReadLine());  if (month < 1 || month > 12)  {  Console.Write("Thang khong hop le, nhap lai thang: ");  }  else break;  }  while (true);  //2. Tạo và nhập năm  Console.Write("Nhap nam: ");  int year = int.Parse(Console.ReadLine());  //3. In ra số ngày trong tháng  Console.WriteLine("So ngay trong thang {0} năm {1} là: {2}",  month, year, DateTime.DaysInMonth(year, month));  }  }  } |

Các testcase của chương trình:

Bảng . Testcase bài 4

|  |  |
| --- | --- |
| 1 2024 |  |
| 4 2023 |  |
| 2 2012 |  |
| 2 2123 |  |
| 0 5 2022 |  |
| 0 13 2 2000 |  |

# 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:

Nội dung code của Bài 5:

Bảng . Nội dung code bài 5

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Bai05  {  internal class Program  {  static void Main(string[] args)  {  Console.WriteLine("Nhap ngay: ");  Day ngay = new Day();  ngay.Nhap();  Console.WriteLine("Thu trong tuan la " + ngay.DayInWeek);  }  }  class Day  {  private int ngay, thang, nam;  //Nhập ngày tháng năm  public void Nhap()  {  thang = 1;  nam = 4;  do  {  do  {  do  {  Console.Write("Ngay: ");  ngay = Convert.ToInt32(Console.ReadLine());  if (!NgayHopLe) Console.WriteLine("Ngay khong hop le!");  else break;  }  while (true);  Console.Write("Thang: ");  thang = Convert.ToInt32(Console.ReadLine());  if (!NgayHopLe)  {  thang = 1;  Console.WriteLine("Thang khong hop le!");  }  else break;  }  while (true);  Console.Write("Nam: ");  nam = Convert.ToInt32(Console.ReadLine());  if (!NgayHopLe)  {  thang = 1;  nam = 4;  Console.WriteLine("Nam khong hop le!");  }  else break;  }  while (true);  }  //Kiểm tra năm nhuận  private bool IsNamNhuan  {  get  {  return (nam % 4 == 0 && nam % 100 != 0) || (nam % 400 == 0);  }  }  //Kiểm tra ngày hợp lệ  private bool NgayHopLe  {  get  {  if (thang < 1 || thang > 12) return false;  if (ngay < 1) return false;  int maxNgay;  switch (thang)  {  case 1:  case 3:  case 5:  case 7:  case 8:  case 10:  case 12:  maxNgay = 31; break;  case 2:  if (IsNamNhuan) maxNgay = 29;  else maxNgay = 28;  break;  default:  maxNgay = 30; break;  }  if (ngay > maxNgay) return false;  return true;  }  }  //Danh sách ngày trong tuần  private Dictionary<int, string> dayInWeek = new Dictionary<int, string>()  {  {0, "Chu nhat" },  {1, "Thu hai" },  {2, "Thu ba" },  {3, "Thu tu" },  {4, "Thu nam" },  {5, "Thu sau" },  {6, "Thu bay" }  };    //Trả về ngày trong tuần  public string DayInWeek  {  get  {  DateTime date = new DateTime(nam, thang, ngay);  int thu = (int)date.DayOfWeek;  return dayInWeek[thu];  }  }  }  } |

Các testcase của chương trình:

Bảng . Testcase bài 5

|  |  |
| --- | --- |
| 15 8 2024 |  |
| 0 31 12 2021 |  |
| 10 13 10 11 2021 |  |
| 29 2 2023 28 2 2023 |  |

# 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:

**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: 3 số nguyên gồm số dòng, số cột của ma trận, và dòng k cần xóa
* Output: Kết quả 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.

Nội dung code của Bài 6:

Bảng . Nội dung code bài 6

|  |
| --- |
| using System;  using System.Collections.Generic;  namespace Bai06  {  internal class Program  {  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;  }  static void Main(string[] args)  {  //1. Nhập n dòng và m cột  Console.WriteLine("1. Tao ma tran: ");  Console.Write("Dong: ");  int n = int.Parse(Console.ReadLine());  Console.Write("Cot: ");  int m = int.Parse(Console.ReadLine());  //2. Tạo và in ra ma trận  Console.WriteLine("\n2. Ma tran da tao:");  MaTran mat = new MaTran(n, m);  mat.Xuat();  //3. Phần tử lớn nhất/nhỏ nhất  Console.WriteLine("\n3a. Phan tu lon nhat: " + mat.MaxValue);  Console.WriteLine("3b. Phan tu nho nhat: " + mat.MinValue);  //4. Dòng có tổng lớn nhất  Console.WriteLine("\n4. Dong co tong lon nhat: " + mat.MaxSumRowIndex);  //5. Tổng các số không phải số nguyên tố  Console.WriteLine("\n5. Tong so khong phai so nguyen to: " +  mat.Sum(x => !IsPrime(x)));  //6. Xóa hàng thứ k  Console.Write("\n6. Xoa dong thu: ");  int k = int.Parse(Console.ReadLine());  mat.DeleteRow(k);  Console.WriteLine("Ma tran da xoa dong " + k);  mat.Xuat();  //7. Xóa cột có số lớn nhất  int colToDelete = mat.FindMaxValue().col;  Console.WriteLine("\n7. Xoa cot co so lon nhat ({0}, cot {1}):",  mat.MaxValue, colToDelete);  mat.DeleteCol(colToDelete);  Console.WriteLine("Ma tran da xoa cot {0}:", colToDelete);  mat.Xuat();  }  }  public delegate bool DieuKien(int i);  class MaTran  {  //Mảng 2 chiều  private List<List<int>> mat;  //Số hàng của ma trận  public int Row  {  get  {  return mat.Count;  }  }  //Số cột của ma trận  public int Col  {  get  {  return Row > 0 ? mat[0].Count : 0;  }  }  //Khởi tạo ma trận r hàng và c cột, giá trị ngẫu nhiên  public MaTran(int r, int c, int minVal = -100, int maxVal = 100)  {  mat = new List<List<int>>();  if (r <= 0 || c <= 0)  {  return;  }  Random rnd = new Random();  for (int i = 0; i < r; i++)  {  List<int> \_row = new List<int>();  for (int j = 0; j < c; j++)  {  int randomNum = rnd.Next(minVal, maxVal + 1);  \_row.Add(randomNum);  }  mat.Add(\_row);  }  }  //In ma trận  public void Xuat()  {  for (int i = 0; i < Row; i++)  {  for (int j = 0; j < Col; j++)  {  Console.Write(mat[i][j] + "\t");  }  Console.WriteLine();  }  }  //Giá trị lớn nhất trong ma trận  public int MaxValue  {  get  {  if (Row == 0 || Col == 0) return -1;  int max = mat[0][0];  for (int i = 0; i < Row; i++)  {  for (int j = 0; j < Col; j++)  {  if (mat[i][j] > max)  {  max = mat[i][j];  }  }  }  return max;  }  }  //Giá trị nhỏ nhất trong ma trận  public int MinValue  {  get  {  if (Row == 0 || Col == 0) return -1;  int min = mat[0][0];  for (int i = 0; i < Row; i++)  {  for (int j = 0; j < Col; j++)  {  if (mat[i][j] < min)  {  min = mat[i][j];  }  }  }  return min;  }  }  //Tổng hàng thứ row  private int SumRow(int row)  {  if (row < 0 || row > Row) return 0;  int sum = 0;  foreach (int i in mat[row])  {  sum += i;  }  return sum;  }  //Hàng có tổng lớn nhất  public int MaxSumRowIndex  {  get  {  if (Row == 0 || Col == 0) return -1;  int index = 0;  int maxSum = SumRow(0);  for (int i = 1; i < Row; i++)  {  int sum = SumRow(i);  if (sum > maxSum)  {  index = i;  maxSum = sum;  }  }  return index;  }  }  //Tính tổng các số thỏa điều kiện trong ma trận  public int Sum(DieuKien dieukien)  {  int sum = 0;  for (int i = 0; i < Row; i++)  {  for (int j = 0; j < Col; j++)  {  if (dieukien(mat[i][j]))  sum += mat[i][j];  }  }  return sum;  }  //Xóa dòng  public void DeleteRow(int k)  {  if (k < 0 || k >= Row)  {  Console.WriteLine("Ma tran khong thay doi!");  return;  }  mat.RemoveAt(k);  }  //Xóa cột  public void DeleteCol(int k)  {  if (k < 0 || k >= Col)  {  Console.WriteLine("Ma tran khong thay doi!");  return;  }  foreach (var row in mat)  {  row.RemoveAt(k);  }  }  //Struct vị trí 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;  }  }  //Tìm vị trí giá trị lớn nhất của ma trận  public Position FindMaxValue()  {  if (Row == 0 || Col == 0) return new Position(-1, -1);  int max = mat[0][0];  Position maxPosition = new Position(0, 0);  for (int i = 0; i < Row; i++)  {  for (int j = 0; j < Col; j++)  {  if (mat[i][j] > max)  {  max = mat[i][j];  maxPosition = new Position(i, j);  }  }  }  return maxPosition;  }  }  } |

Các testcase của chương trình:

Bảng . Testcase bài 6

|  |  |
| --- | --- |
| 4 3 2 |  |
| 1 1 1 |  |
| 0 0 3 |  |
| 10 9 20 |  |

LinkCode: <https://github.com/nguyenhun11/-IT008.Q14-BTH1_NguyenGiaHung_24520604>