**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 Chí Nguyên

Mã số sinh viên: 24521186

🙡🙢 Tp. Hồ Chí Minh, 09/2025 🙠🙣

**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. Cài đặt các hàm thực hiện các chức năng theo yêu cầu 5](#_Toc209903244)

[2. Viết chương trình nhập số nguyên dương n, tính tổng các số nguyên tố < n và xuất kết quả ra màn hình. 7](#_Toc209903245)

[3. Viết chương trình nhập ngày tháng năm, cho biết ngày tháng năm đó có hợp lệ không? 9](#_Toc209903246)

[4. Viết chương trình nhập tháng và năm. In ra số ngày của tháng đó. 11](#_Toc209903247)

[5. Viết chương trình nhập vào ngày tháng năm. Cho biết thứ trong tuần. 13](#_Toc209903248)

[6. Viết chương trình cho phép tạo ma trận chứa các số nguyên ngẫu nhiên gồm n dòng, m cột. Cài đặt các hàm thực hiện các chức năng theo yêu cầu 16](#_Toc209903249)

**DANH MỤC BẢNG**

[Bảng 1 Nội dung code của chương trình bài 1 5](#_Toc210891988)

[Bảng 2 Các testcase của chương trình bài 1 7](#_Toc210891989)

[Bảng 3 Nội dung code của chương trình bài 2 8](#_Toc210891990)

[Bảng 4 Các testcase của chương trình bài 2 8](#_Toc210891991)

[Bảng 5 Nội dung code của chương trình bài 3 9](#_Toc210891992)

[Bảng 6 Các testcase của chương trình bài 3 10](#_Toc210891993)

[Bảng 7 Nội dung code của chương trình bài 4 12](#_Toc210891994)

[Bảng 8 Các testcase của chương trình bài 4 12](#_Toc210891995)

[Bảng 9 Nội dung code của chương trình bài 5 13](#_Toc210891996)

[Bảng 10 Các testcase của chương trình bài 5 14](#_Toc210891997)

[Bảng 11 Nội dung code của chương trình bài 6 16](#_Toc210891998)

[Bảng 12 Các testcase của chương trình bài 6 20](#_Toc210891999)

**NỘI DUNG BÀI LÀM**

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

**a. Tính tổng các số lẻ trong mảng.**

**b. Đếm số nguyên tố trong mảng.**

**c. Tìm số chính phương nhỏ nhất (nếu không có hàm trả về -1).**

Mô tả đầu vào và đầu ra của bài toán:

* Input: Một mảng một chiều gồm n số nguyên ngẫu nhiên.
* Output: 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 của chương trình bài 1

|  |
| --- |
| using System;  namespace Bai01  {  class Program  {  static void Main(string[] args)  {  // Tao mang n so nguyen ngau nhien tu 1 den 100  int n;  while (true)  {  Console.Write("Nhap so luong phan tu cua mang (> 0): ");  string input = Console.ReadLine();  if (int.TryParse(input, out n) && n > 0)  break;  Console.WriteLine("Loi: Vui long nhap mot so nguyen duong hop le!");  }  int[] arr = CreateRandomArray(n, 1, 100);  // In mang ra man hinh  Console.WriteLine("Mang da duoc khoi tao ngau nhien la: ");  PrintArray(arr, n);  // a. tinh tong cac so le trong mang  Console.WriteLine("\na. Tong cac so le: " + CalculateOddSum(arr, n));  // b. dem cac so nguyen to trong mang  Console.WriteLine("b. So luong so nguyen to: " + CountPrime(arr, n));  // c. tim so chinh phuong nho nhat  Console.WriteLine("c. So chinh phuong nho nhat la: " + FindMinPerfectSquare(arr, n));  }  // Tao mang ngau nhien co n phan tu gioi han tu minVal -> maxVal  static int[] CreateRandomArray(int n, int minVal, int maxVal)  {  int[] arr = new int[n];  Random rnd = new Random();  for (int i = 0; i < n; i++)  {  arr[i] = rnd.Next(minVal, maxVal);  }  return arr;  }  // In mang  static void PrintArray(int[] arr, int n)  {  foreach (int a in arr)  Console.Write(a + " ");  }  // Tinh tong cac so le  static int CalculateOddSum(int[] arr, int n)  {  int sum = 0;  foreach (int a in arr)  {  if (a % 2 != 0)  {  sum += a;  }  }  return sum;  }  // Dem so luong so nguyen to  static int CountPrime(int[] arr, int n)  {  int count = 0;  foreach (int a in arr)  {  if (IsPrime(a))  {  count++;  }  }  return count;  }  // Tim so chinh phuong nho nhat  static int FindMinPerfectSquare(int[] arr, int n)  {  int ans = int.MaxValue;  foreach (int a in arr)  {  if (IsPerfectSquare(a) && a < ans)  {  ans = a;  }  }  if (ans == int.MaxValue)  ans = -1;  return ans;  }  // Kiem tra so chinh phuon  static bool IsPerfectSquare(int a)  {  int i = (int)Math.Sqrt(a);  if (i \* i == a) return true;  return false;  }  // Kiem tra so nguyen to  static bool IsPrime(int a)  {  if (a < 2) return false;  for (int i = 2; i \* i <= a; i++)  {  if (a % i == 0)  return false;  }  return true;  }  }  } |

Bảng Các testcase của chương trình bài 1

|  |  |
| --- | --- |
| Input:  abc  -5  10 |  |
| Input:  100 |  |

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

Mô tả đầu vào và đầu ra của bài toán:

* Input: Số nguyên dương n.
* Output: Tổng các số nguyên tố nhỏ hơn n.

Nội dung code của Bài 2

Bảng Nội dung code của chương trình bài 2

|  |
| --- |
| using System;  namespace BTH1.Bai02  {  class Program  {  static void Main(string[] args)  {  int n;  // Nhap n  while (true)  {  Console.Write("Nhap so nguyen duong n (n > 0): ");  string input = Console.ReadLine();  if (int.TryParse(input, out n) && n > 0)  break;  Console.WriteLine("Loi: Vui long nhap mot so nguyen duong hop le!");  }  // Tinh tong cac so nguyen to nho hon n  int tongSNT = 0;  for (int i = 2; i < n; i++)  {  if(IsPrime(i))  {  tongSNT += i;  }  }  Console.WriteLine("Tong cac so nguyen to < n: " + tongSNT);  }  // Ham kiem tra so nguyen to  static bool IsPrime(int a)  {  if (a < 2) return false;  for (int i = 2; i \* i <= a; i++)  {  if (a % i == 0) return false;  }  return true;  }  }  } |

Bảng Các testcase của chương trình bài 2

|  |  |
| --- | --- |
| Input:  abc  -5  10 |  |
| Input:  /@~$  100 |  |

# 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: Ba số nguyên lần lượt là ngày, tháng, năm.
* Output: Ngày tháng năm vừa nhập hợp lệ hay không hợp lệ.

Nội dung code của Bài 3

Bảng Nội dung code của chương trình bài 3

|  |
| --- |
| using System;  namespace Bai03  {  class Program  {  static void Main(string[] args)  {  int ngay, thang, nam;  // nhap ngay thang nam tu ban phim  while(true)  {  string d, m, y;  Console.Write("Nhap ngay (1 -> 31): ");  d = Console.ReadLine();  Console.Write("Nhap thang (1 -> 12): ");  m = Console.ReadLine();  Console.Write("Nhap nam (> 0): ");  y = Console.ReadLine();  if ((int.TryParse(d, out ngay) && ngay > 0) &&  (int.TryParse(m, out thang) && thang > 0) &&  (int.TryParse(y, out nam) && nam > 0))  break;  Console.WriteLine("Dinh dang khong hop le! Vui long nhap so:");  }    // Xuat ket qua ngay/thang/nam co hop le hay khong  if (IsValidDate(ngay, thang, nam))  Console.WriteLine("Ngay thang nam hop le!");  else  Console.WriteLine("Ngay thang nam khong hop le!");  }  // Kiem tra ngay/thang/nam co hop le hay khong  static bool IsValidDate(int ngay, int thang, int nam)  {  if (nam < 0) return false;  if (thang < 1 || thang > 12) return false;  if (ngay < 1) return false;  int[] ngayTrongThangs = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  bool isLeapYear = (nam % 4 == 0 && nam % 100 != 0) || (nam % 400 == 0);  if (isLeapYear && thang == 2)  {  if (ngay > 29) return false;  }  else  {  if (ngay > ngayTrongThangs[thang - 1]) return false;  }  return true;  }  }  } |

Bảng Các testcase của chương trình bài 3

|  |  |
| --- | --- |
| Input:  4 4 2006 |  |
| Input:  abc 3 2006  10 xyz 2006  10 3 qwerty  10 3 2006 |  |
| Input:  3.14 6 2025  3 6.96 2025  7 7 99.99  7 7 2025 |  |
| Input:  -5 6 2020  5 -6 2020  5 6 -2020  5 6 2020 |  |
| Input:  32 12 2025 |  |
| Input:  15 13 2025 |  |
| Input:  29 2 2024 |  |
| Input:  29 2 2025 |  |

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

Mô tả đầu vào và đầu ra của bài toán:

* Input: Hai số nguyên lần lượt là tháng và năm.
* Output: Số ngày của tháng/năm vừa nhập.

Nội dung code của Bài 4

Bảng Nội dung code của chương trình bài 4

|  |
| --- |
| using System;  namespace Bai04  {  class Program  {  static void Main(string[] args)  {  int thang, nam;  // Xu li nhap thang/nam  while (true)  {  string m, y;  Console.Write("Nhap thang (1 -> 12): ");  m = Console.ReadLine();  Console.Write("Nhap nam (> 0): ");  y = Console.ReadLine();  if ((int.TryParse(m, out thang) && thang > 0 && thang < 13) &&  (int.TryParse(y, out nam) && nam > 0))  break;  Console.WriteLine("Du lieu khong hop le! Vui long nhap lai:");  }  // Tao mang chua ngay cua tung thang trong nam  int[] ngayTrongThangs = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  // Kiem tra nam nhuan  bool isLeapYear = (nam % 4 == 0 && nam % 100 != 0) || (nam % 400 == 0);  // Xuat ket qua  Console.Write($"So ngay cua thang {thang} nam {nam} la: ");  if (isLeapYear && thang == 2)  {  Console.Write(29);  }  else  {  Console.Write(ngayTrongThangs[thang - 1]);  }  }  }  } |

Bảng Các testcase của chương trình bài 4

|  |  |
| --- | --- |
| Input:  abc 2025  10 xyzw  10 2025 |  |
| Input:  0 2025  13 2025  2 2025 |  |
| Input:  12 2000000000 |  |
| Input:  -10 2006  10 -2006  10 2006 |  |
| Input:  2 2024 |  |

# 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: Ba số nguyên lần lượt là ngày, tháng và năm.
* Output: Thứ trong tuần của ngày/tháng/năm vừa nhập.

Nội dung code của Bài 5

Bảng Nội dung code của chương trình bài 5

|  |
| --- |
| using System;  namespace Bai05  {  class Program  {  static void Main(string[] args)  {  int ngay, thang, nam;  // Nhap ngay thang nam tu ban phim  while(true)  {  string d, m, y;  Console.Write("Enter the day: ");  d = Console.ReadLine();  Console.Write("Enter the month: ");  m = Console.ReadLine();  Console.Write("Enter the year: ");  y = Console.ReadLine();  if(int.TryParse(d, out ngay) && int.TryParse(m, out thang) && int.TryParse(y, out nam)  && IsValidDate(ngay, thang, nam))  break;  Console.WriteLine("Invalid data. Please try again!");  }  // Khai bao doi tuong kieu DateTime va truyen doi so ngay/thang/nam de tan dung ham DayOfWeek  DateTime date = new DateTime(nam, thang, ngay);  Console.WriteLine("{0}/{1}/{2} is: " + date.DayOfWeek, ngay, thang, nam);  }  static bool IsValidDate(int ngay, int thang, int nam)  {  if (nam < 0) return false;  if (thang < 1 || thang > 12) return false;  if (ngay < 1) return false;  int[] ngayTrongThangs = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  bool isLeapYear = (nam % 4 == 0 && nam % 100 != 0) || (nam % 400 == 0);  if (isLeapYear && thang == 2)  {  if (ngay > 29) return false;  }  else  {  if (ngay > ngayTrongThangs[thang - 1]) return false;  }  return true;  }  }  } |

Bảng Các testcase của chương trình bài 5

|  |  |
| --- | --- |
| Input:  10 3 2006 |  |
| Input:  a 10 2025  10 b 2025  10 10 c  10 10 2025 |  |
| Input:  -1 10 2025  10 -1 2025  10 10 -2025  10 10 2025 |  |
| Input:  3.14 10 2025  10 7.77 2025  10 10 2025.1  10 10 2025 |  |

# 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: Ma trận (mảng 2 chiều) các số nguyên ngẫu nhiên gồm n dòng và m cột. Số nguyên k.
* Output: Xuất ma trận ra màn hình; phần tử lớn nhất/nhỏ nhất; dòng có tổng lớn nhất; tổng các số không phải là số nguyên tố; xuất ma trận sau khi xóa dòng thứ k; xuất ma trận sau khi xóa cột chứa phần tử lớn nhất.

Nội dung code của Bài 6

Bảng Nội dung code của chương trình bài 6

|  |
| --- |
| using System;  namespace Bai06  {  class Program  {  static void Main(string[] args)  {  int numRows, numCols; // n: number of rows, m: number of column  // Nhap so dong va so cot  while (true)  {  string r, c;  Console.Write("Nhap so dong: ");  r = Console.ReadLine();  Console.Write("Nhap so cot: ");  c = Console.ReadLine();  if((int.TryParse(r, out numRows) && numRows > 0) && (int.TryParse(c, out numCols) && numCols > 0))  break;  Console.WriteLine("So dong va so cot phai la so nguyen duong. Vui long nhap lai!");  }  Matrix matrix = new Matrix(numRows, numCols);  // a. Xuat ma tran  Console.WriteLine("Ma tran da duoc khoi tao ngau nhien!");  Console.WriteLine("a. Xuat ma tran: ");  matrix.Print();  // b. Tim max/min  Console.WriteLine("b. Tim phan tu lon nhat/ nho nhat trong ma tran: ");  Console.WriteLine("\t-Phan tu lon nhat cua ma tran la: {0}", matrix.FindMax());  Console.WriteLine("\t-Phan tu nho nhat cua ma tran la: {0}", matrix.FindMin());  // c. Xuat dong co tong lon nhat  Console.WriteLine($"c. Dong co tong lon (tinh tu 0) nhat la dong thu {matrix.FindMaxSumRow()}");  // d. Tinh tong cac so khong phai la so nguyen to  Console.WriteLine($"d. Tong cac so khong phai la so nguyen to la {matrix.FindNonPrimeSum()}");  // e. Xoa dong thu k trong ma tran  Console.WriteLine("e. Xoa dong thu k trong ma tran:");  // Nhap dong can xoa  int k;  do  {  Console.Write("Nhap dong can xoa (tinh tu 0): ");  k = Convert.ToInt16(Console.ReadLine());  } while (k >= numRows || k < 0);  // Xoa  matrix.DeleteRow(k);  // In ma tran  Console.WriteLine($"Ma tran sau khi xoa dong thu {k}: ");  matrix.Print();  /// f. Xoa cot chua phan tu lon nhat  Console.WriteLine("f. Xoa cot chua phan tu lon nhat trong ma tran: ");  // Tim cot chua phan tu lon nhat  int colToDel = matrix.FindMaxEleColumnIndex();  // Xoa cot vua tim duoc  matrix.DeleteColumn(colToDel);  // In ma tran  Console.WriteLine($"Ma tran sau khi xoa cot thu {colToDel} (cot chua phan tu lon nhat) la: ");  matrix.Print();  }  private static bool isPrime(int n)  {  if (n < 2) return false;  for (int i = 2; i \* i <= n; i++)  {  if (n % i == 0) return false;  }  return true;  }  // lop Matrix de thuc hien cac yeu cau cua de bai  public class Matrix  {  int numRows, numCols;  int[][] matrix;  public Matrix(int numRows = 0, int numCols = 0)  {  this.numRows = numRows;  this.numCols = numCols;  matrix = new int[this.numRows][];  Random rnd = new Random();  for (int i = 0; i < this.numRows; i++)  {  matrix[i] = new int[numCols];  for (int j = 0; j < numCols; j++)  {  matrix[i][j] = rnd.Next(1, 20);  }  }  }  public void Print()  {  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  Console.Write(matrix[i][j] + " ");  }  Console.Write("\n");  }  }  public int FindMax()  {  int maxValue = int.MinValue;  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  if (matrix[i][j] > maxValue)  maxValue = matrix[i][j];  }  }  return maxValue;  }  public int FindMin()  {  int minValue = int.MaxValue;  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  if (matrix[i][j] < minValue)  minValue = matrix[i][j];  }  }  return minValue;  }  public int FindMaxSumRow()  {  int maxRowSum = 0;  int maxRowIdx = 0;  for (int i = 0; i < numRows; i++)  {  int tempSum = 0;  for (int j = 0; j < numCols; j++)  {  tempSum += matrix[i][j];  }  if (tempSum > maxRowSum)  {  maxRowSum = tempSum;  maxRowIdx = i;  }  }  return maxRowIdx;  }  public int FindNonPrimeSum()  {  int tongKhongPhaiSoNguyenTo = 0;  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  if (!isPrime(matrix[i][j]))  {  tongKhongPhaiSoNguyenTo += matrix[i][j];  }  }  }  return tongKhongPhaiSoNguyenTo;  }  public void DeleteRow(int rowToDelete)  {  if (rowToDelete == numRows - 1)  {  numRows--;  }  else  {  for(int r = rowToDelete; r < numRows - 1; r++)  {  for (int i = 0; i < numCols; i++)  {  matrix[r][i] = matrix[r + 1][i];  }  }  numRows--;  }  }  public int FindMaxEleColumnIndex()  {  int maxValue = FindMax();  for (int i = 0; i < numRows; i++)  {  for (int j = 0; j < numCols; j++)  {  if (matrix[i][j] == maxValue)  return j;  }  }  return -1;  }  public void DeleteColumn(int colToDelete)  {  if (colToDelete == numCols - 1)  numCols--;  else  {  for (int c = colToDelete; c < numCols - 1; c++)  {  for(int i = 0; i < numRows; i++)  {  matrix[i][c] = matrix[i][c + 1];  }  }  numCols--;  }    }  }  }  } |

Bảng Các testcase của chương trình bài 6

|  |  |
| --- | --- |
| Input:  3 4  0 |  |
| Input:  5 4  4 |  |
| Input:  -3 -4  -3 4  3 -4  3 4  1 |  |
| Input:  x y  x 3  3 3  2 |  |
| Input:  0 4  1 1  0 |  |
| Input:  3 3  10  -5  3  2 |  |
| Input:  12 12  5 | ­­ |

LinkCode: [Github\_24521186\_NguyenChiNguyen](https://github.com/Naathan404/IT008_LapTrinhTrucQuan/tree/master/ThucHanh/LAB01/BTH1_NguyenChiNguyen_245211886)