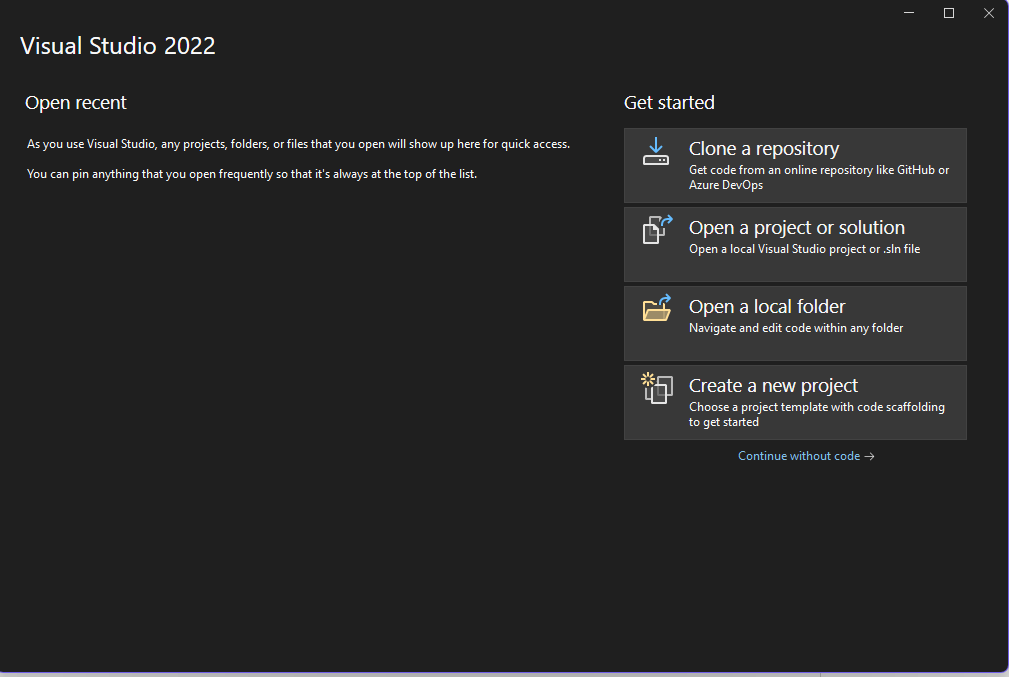
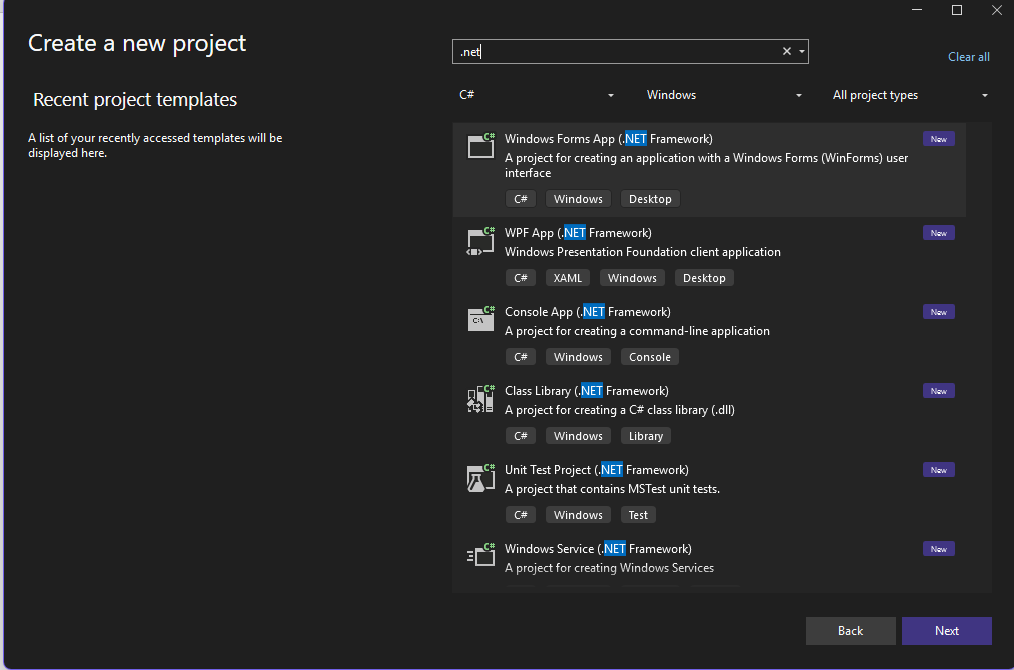
**Bài buổi 5**

**Bước 1:** Khởi động chương trình visual studio 2022. Hình ảnh dưới đây là màn  hình khởi động của ứng dụng:

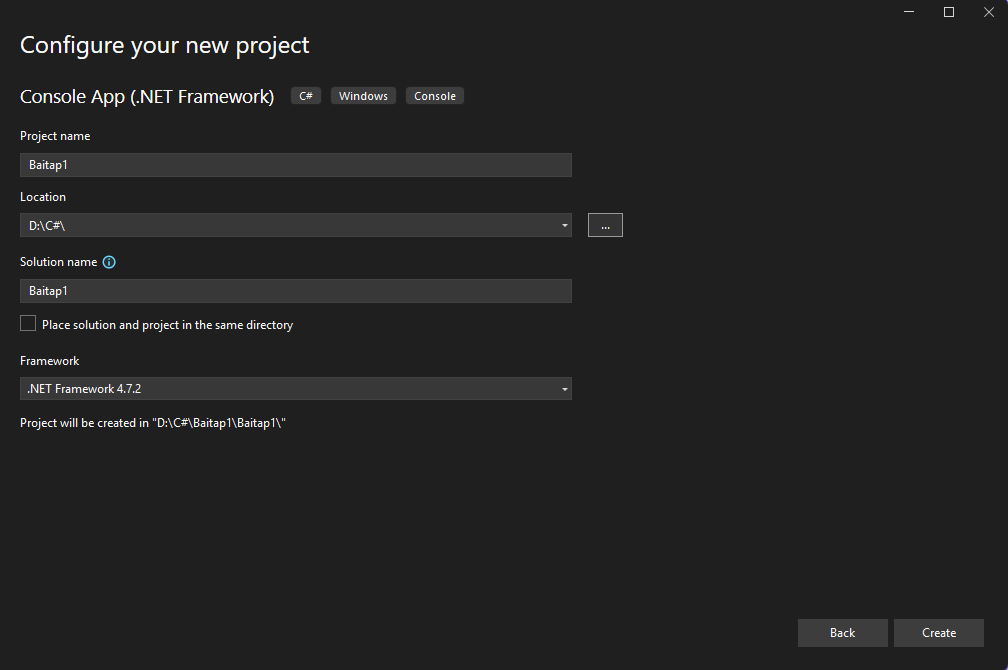
**1. Tạo Project Console Application trên Microsoft visual studio 2022**

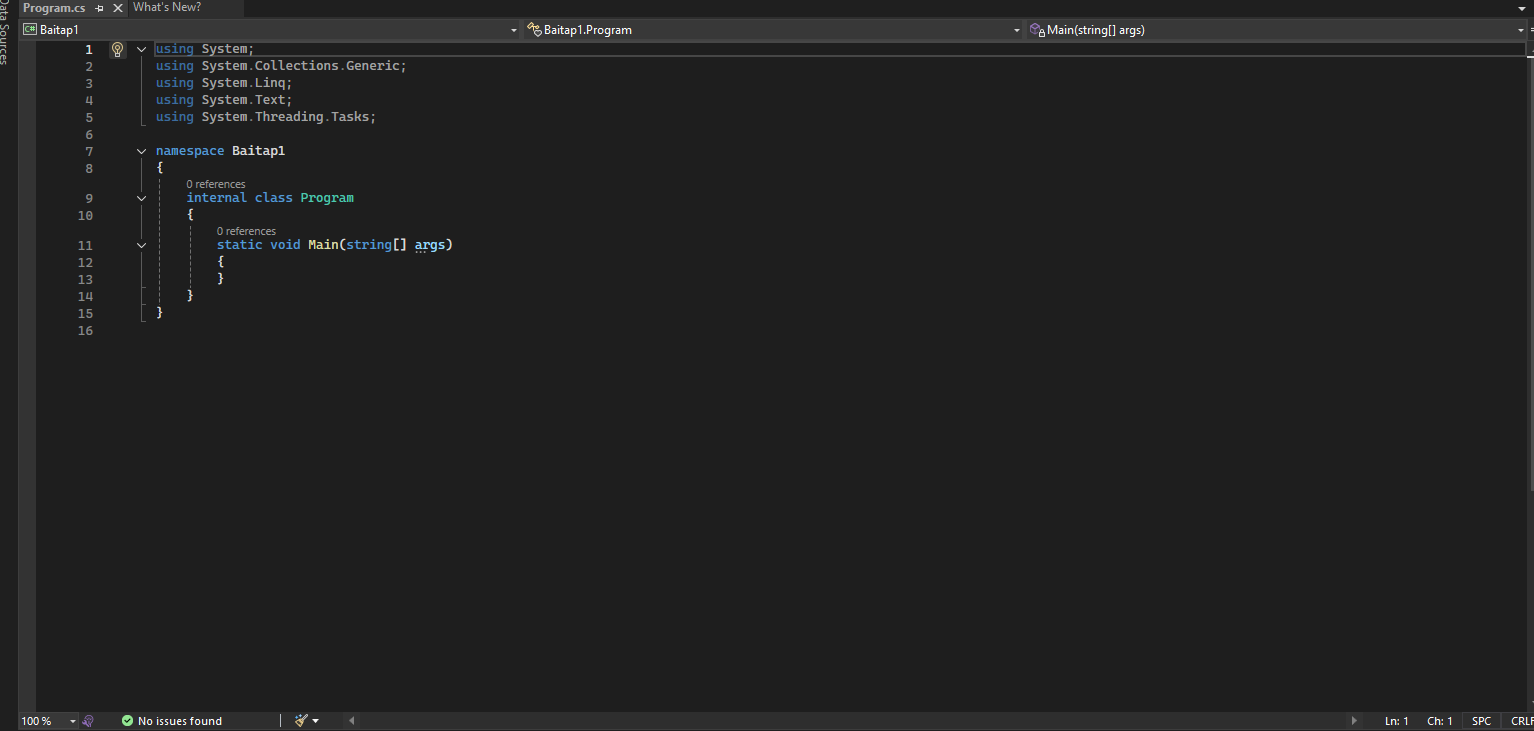


**Bước 2:** Chọn Create a new project trên màn hình khởi động. Trong màn hình  tiếp theo, ở mục All languages → chọn C# trong danh sách ngôn ngữ lập trình.  Mục All platforms → chọn Windows. Ở mục All project types → chọn Console.



**Bước 3:** Trong cửa sổ tiếp theo, bạn điền đầy đủ các thông tin về Project.





\* Phân tích đề bài:

− Vào: chiều dài a, chiều rộng b.

− Ra: Chu vi P, diện tích S.

{Chương trình này thực hiện nhập chiều dài a và chiều rộng b từ bàn phím. Tính  và hiển thị chu vi và diện tích của hình chữ nhật}

**Program** Tinh\_CV\_DT

1. Nhập dữ liệu

Read(a, b);

2. Tính chu vi và diện tích

P := (a + b) \* 2;

S:= a \* b;

3. Hiển thị kết quả

Write(P, S);

**End**.

**Source Code**:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Baitap1

{

internal class Program

{

static void Main(string[] args)

{

// Chương trình tính chu vi và diện tích hình chữ nhật

// Nhập dữ liệu

Console.Write(" Nhap vao chieu dai: ");

double a = Convert.ToDouble(Console.ReadLine());

Console.Write(" Nhap vao chieu rong: ");

double b = Convert.ToDouble(Console.ReadLine());

// Tính chu vi và diện tích

double fl = (a + b) \* 2;

double S = a \* b;

// iển thị kết quả

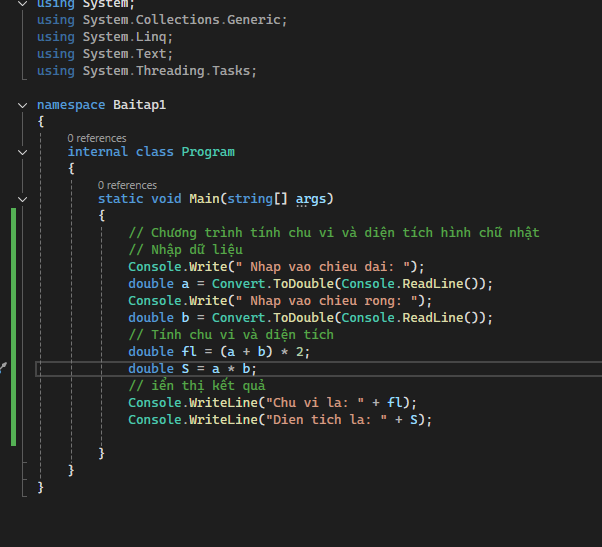
Console.WriteLine("Chu vi la: " + fl);

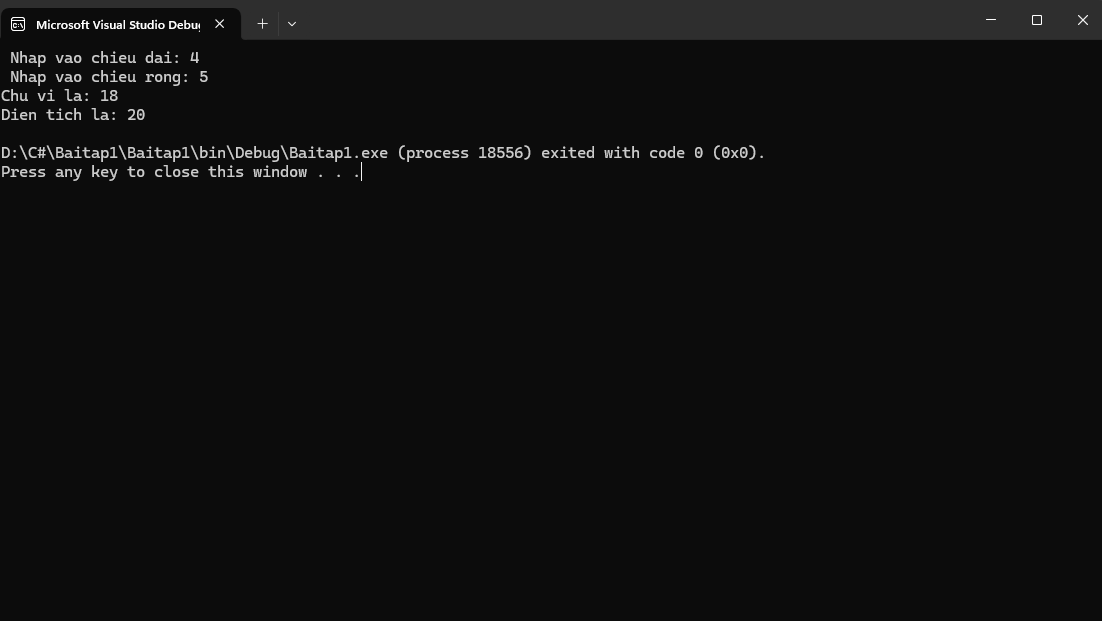
Console.WriteLine("Dien tich la: " + S);

}

}

}





**2. Biến là gì?**

− Biến là tên một ô nhớ trong bộ nhớ, dùng để chứa dữ liệu, giá trị của biến  có thể thay đổi trong quá trình thực thi chương trình.

− Dữ liệu gồm 3 loại:

+ Dữ liệu đầu vào.

+ Dữ liệu đầu ra.

+ Dữ liệu trung gian trong quá trình xử lý dữ liệu.

− Quy tắc đặt tên biến:

+ Là một dãy các ký tự chữ cái (A-Z, a-z), chữ số (0-9), dấu gạch dưới. + Không được bắt đầu bằng chữ số, được phép bắt đầu bằng dấu gạch  dưới.

+ Không được trùng với từ khóa trong C#.

+ C# PHÂN BIỆT KÝ TỰ HOA VÀ THƯỜNG.

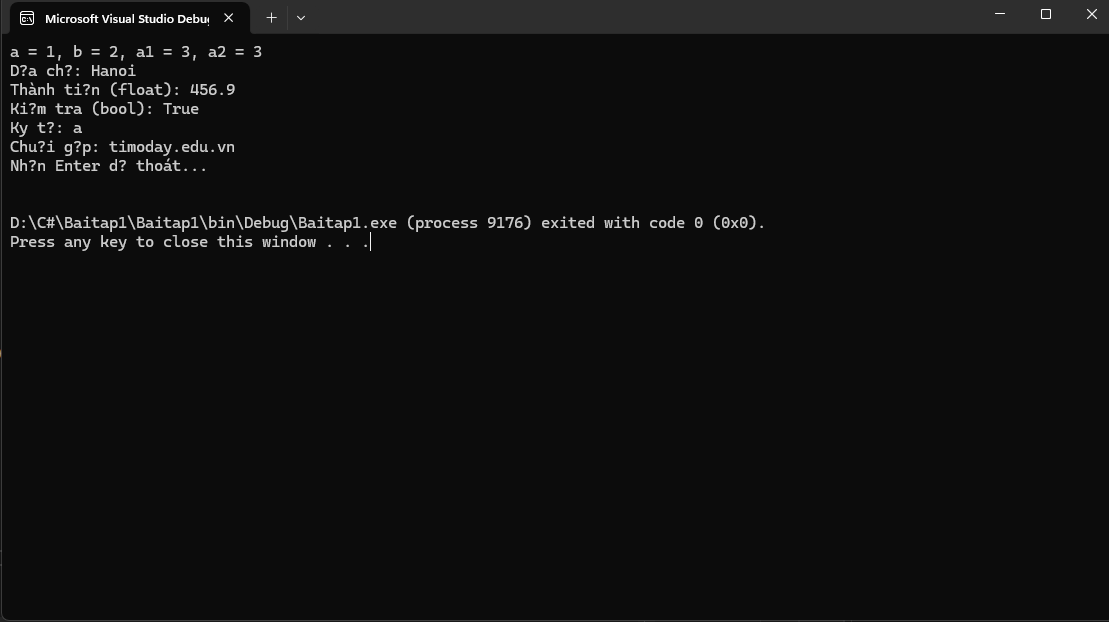
**3. Kiểu dữ liệu**

C# là một ngôn ngữ lập trình kiểm soát chặt chẽ về kiểu dữ liệu. Nghĩa là bạn phải khai báo kiểu dữ liệu với mỗi một biến.

Ví dụ về khai báo biến (declaration):

Kiểu\_dữ\_liệu tên\_biến\_1[, tên\_biến\_2, …] [=giá\_trị\_khởi\_tạo];

**SOURCE CODE:**

****

using System;

namespace Baitap1

{

internal class Program

{

static void Main(string[] args)

{

// Khai báo biến

int a = 1, b = 2;

int a1, a2 = 3; // a1 chưa gán

a1 = a + b; // gán để tránh warning

string diaChi = "Hanoi";

float thanhTien = 456.9F; // F cho kiểu float

bool kt = true;

char ch = 'a';

string s1 = "timoday";

string s2 = ".edu.vn";

string s3 = s1 + s2;

// In ra để thấy kết quả và tránh biến “không dùng”

Console.WriteLine($"a = {a}, b = {b}, a1 = {a1}, a2 = {a2}");

Console.WriteLine($"Địa chỉ: {diaChi}");

Console.WriteLine($"Thành tiền (float): {thanhTien}");

Console.WriteLine($"Kiểm tra (bool): {kt}");

Console.WriteLine($"Ký tự: {ch}");

Console.WriteLine($"Chuỗi gộp: {s3}");

Console.WriteLine("Nhấn Enter để thoát...");

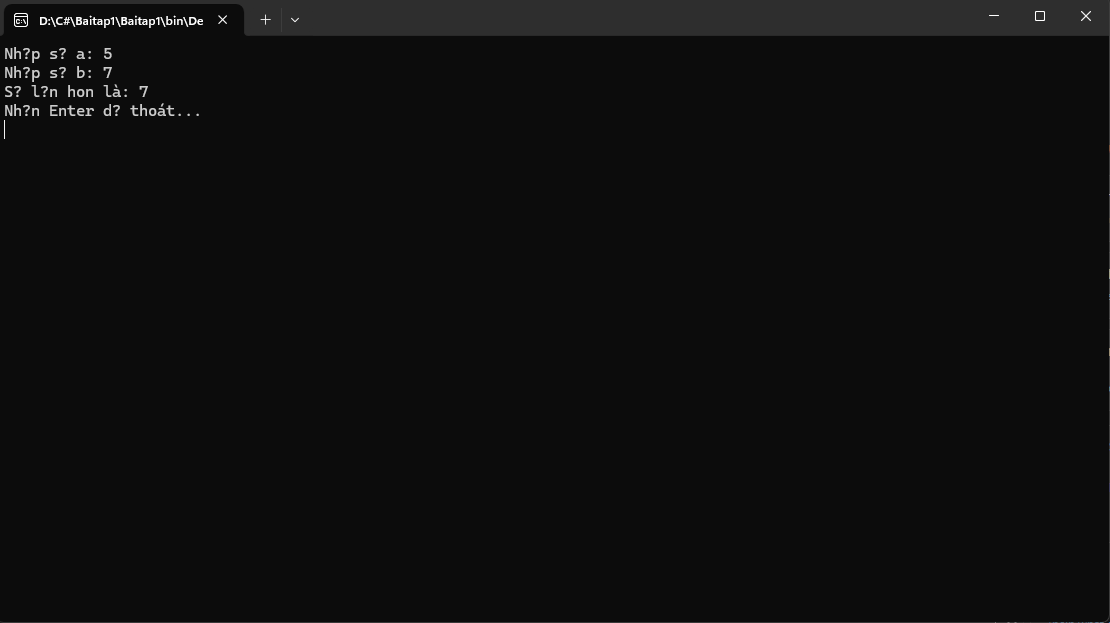
Console.ReadLine();

}

}

}

**Thực hành 2:** Tìm số lớn hơn trong 2 số nguyên a và b được nhập vào từ bàn phím.

****

using System;

namespace ThucHanh2

{

internal class Program

{

static void Main(string[] args)

{

// Nhập số nguyên a

Console.Write("Nhập số a: ");

int a = int.Parse(Console.ReadLine());

// Nhập số nguyên b

Console.Write("Nhập số b: ");

int b = int.Parse(Console.ReadLine());

// So sánh và in ra kết quả

if (a > b)

{

Console.WriteLine($"Số lớn hơn là: {a}");

}

else if (b > a)

{

Console.WriteLine($"Số lớn hơn là: {b}");

}

else

{

Console.WriteLine("Hai số bằng nhau.");

}

Console.WriteLine("Nhấn Enter để thoát...");

Console.ReadLine();

}

}

}

**Thực hành 3:** Tìm số có giá trị lớn nhất trong 3 số nguyên a, b, c được nhập vào  từ bàn phím.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_1150080131\_TranDuc

{

internal class TH3

{

static void Main(string[] args)

{

Console.Write("Nhập số a: ");

int a = int.Parse(Console.ReadLine());

Console.Write("Nhập số b: ");

int b = int.Parse(Console.ReadLine());

Console.Write("Nhập số c: ");

int c = int.Parse(Console.ReadLine());

if (a == b && b == c)

{

Console.WriteLine("Ba số bằng nhau.");

}

else

{

int max = Math.Max(a, Math.Max(b, c));

Console.Write("Số lớn nhất là: " + max);

// Nếu muốn báo các số bằng nhau với max

if ((a == max ? 1 : 0) + (b == max ? 1 : 0) + (c == max ? 1 : 0) > 1)

{

Console.Write(" (có nhiều số cùng lớn nhất)");

}

Console.WriteLine();

}

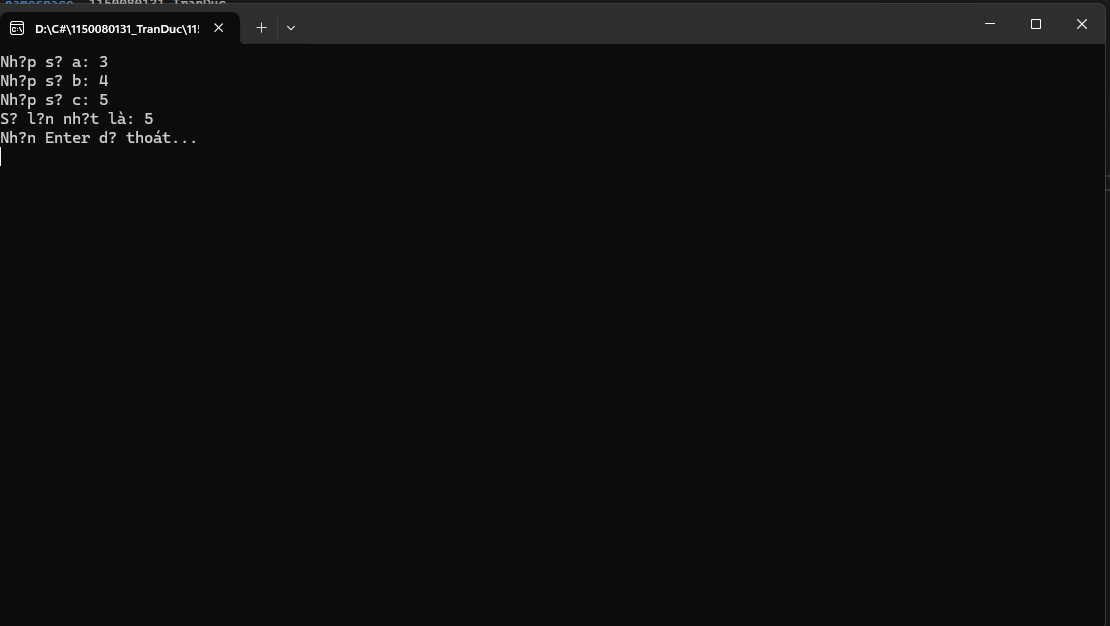
Console.WriteLine("Nhấn Enter để thoát...");

Console.ReadLine();

}

}

}

****

**Thực hành 4:** Cho biết một tháng có bao nhiêu ngày. Biết tháng, năm được nhập  vào từ bàn phím.

using System;

namespace \_1150080131\_TranDuc

{

public static class TH4

{

public static void Run()

{

Console.Write("Nhập tháng (1-12): ");

int month = int.Parse(Console.ReadLine());

Console.Write("Nhập năm: ");

int year = int.Parse(Console.ReadLine());

int days;

switch (month)

{

case 1: case 3: case 5: case 7: case 8: case 10: case 12: days = 31; break;

case 4: case 6: case 9: case 11: days = 30; break;

case 2:

days = (year % 400 == 0 || (year % 4 == 0 && year % 100 != 0)) ? 29 : 28;

break;

default:

Console.WriteLine("Tháng không hợp lệ!"); return;

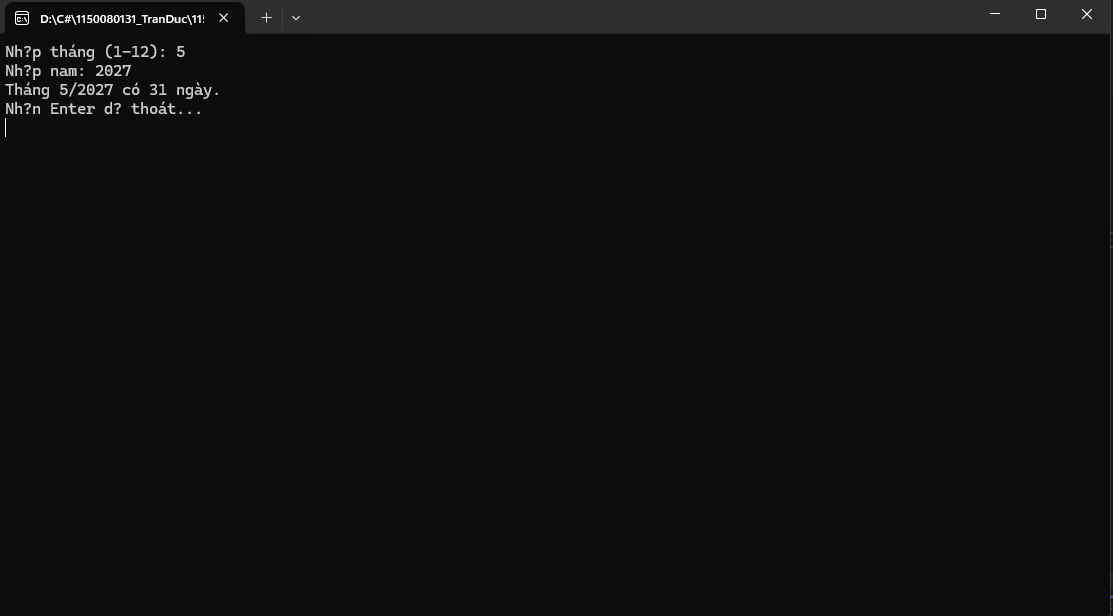
}

Console.WriteLine($"Tháng {month}/{year} có {days} ngày.");

}

}

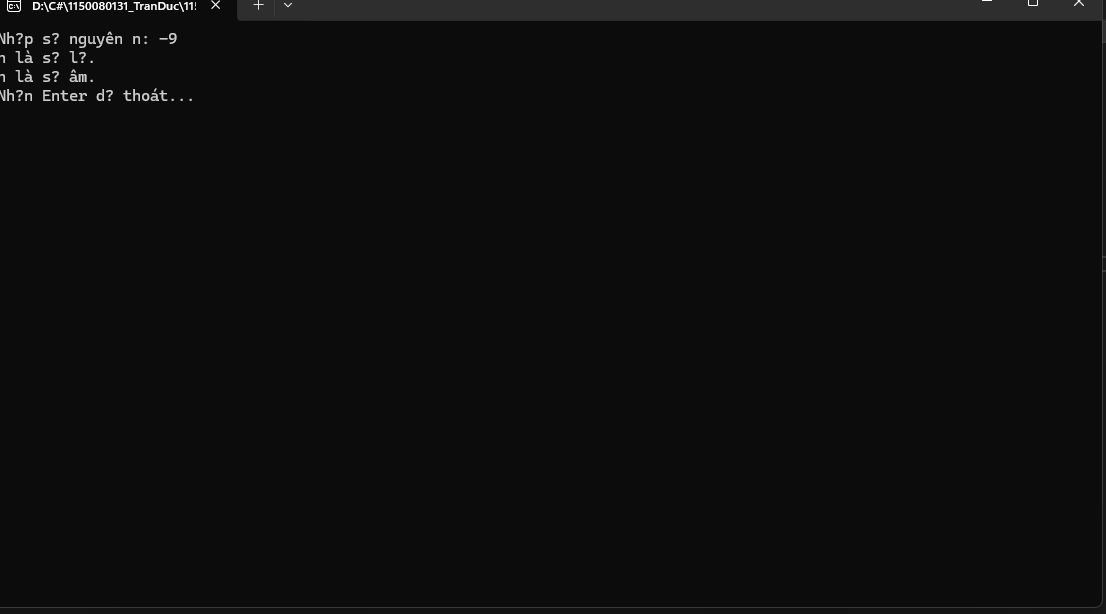
}

****

**Thực hành 5:** Viết chương trình nhập vào một số nguyên n. Cho biết:12

a) n là số chẵn hay số lẻ?

b) n là số âm hay số không âm?

****

using System;

namespace \_1150080131\_TranDuc

{

public static class TH5

{

public static void Run()

{

Console.Write("Nhập số nguyên n: ");

int n = int.Parse(Console.ReadLine());

// a) Kiểm tra chẵn lẻ

if (n % 2 == 0)

Console.WriteLine("n là số chẵn.");

else

Console.WriteLine("n là số lẻ.");

// b) Kiểm tra âm hay không âm

if (n < 0)

Console.WriteLine("n là số âm.");

else

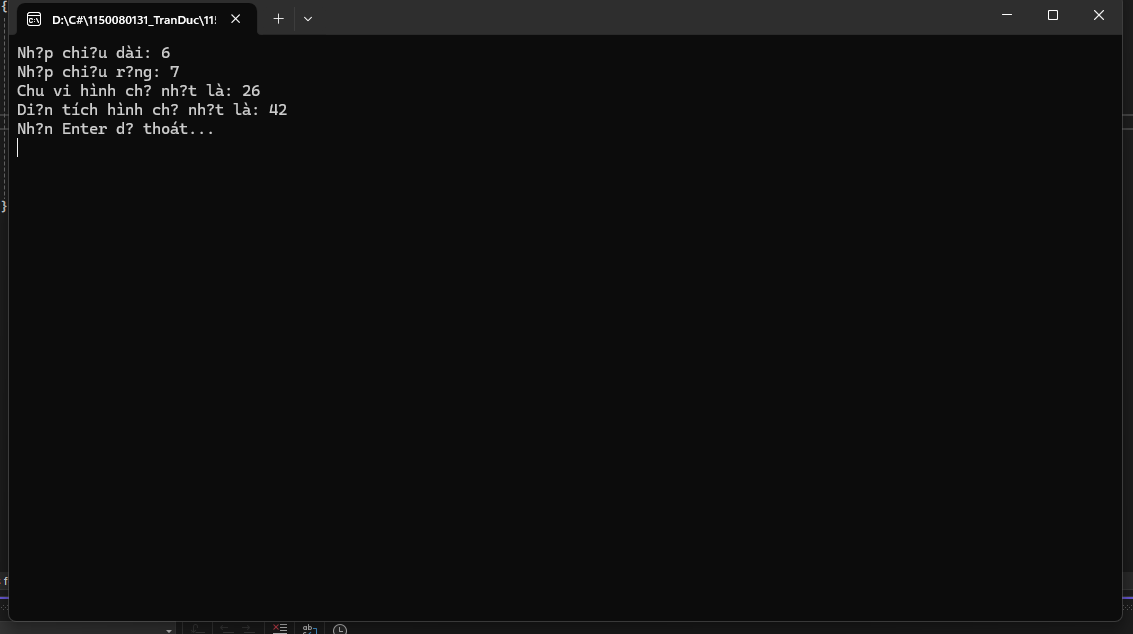
Console.WriteLine("n là số không âm.");

}

}

}

**Thực hành 6:** Viết chương trình nhập vào 2 số thực dương chỉ chiều dài và chiều  rộng của hình chữ nhật. In ra màn hình chu vi và diện tích của hình chữ nhật đó.

****

using System;

namespace \_1150080131\_TranDuc

{

public static class TH6

{

public static void Run()

{

double dai = ReadPositiveDouble("Nhập chiều dài: ");

double rong = ReadPositiveDouble("Nhập chiều rộng: ");

double chuVi = 2 \* (dai + rong);

double dienTich = dai \* rong;

Console.WriteLine($"Chu vi hình chữ nhật là: {chuVi}");

Console.WriteLine($"Diện tích hình chữ nhật là: {dienTich}");

}

// Hàm nhập số thực dương

private static double ReadPositiveDouble(string prompt)

{

while (true)

{

Console.Write(prompt);

if (double.TryParse(Console.ReadLine(), out double value) && value > 0)

{

return value;

}

Console.WriteLine("Giá trị không hợp lệ. Vui lòng nhập số thực dương.");

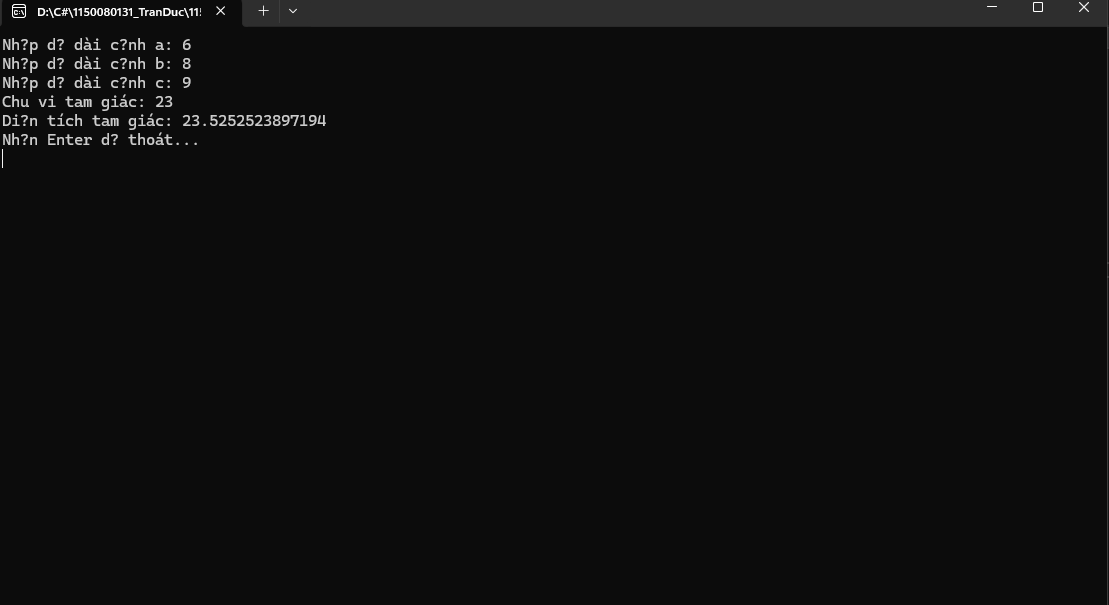
}

}

}

}

**Thực hành 7:** Viết chương trình nhập vào ba số thực chỉ độ dài của ba đoạn  thẳng. Kiểm tra nếu ba đoạn thẳng này lập thành được một tam giá thì hiển thị  chu vi và diện tích của tam giác đó.

****

using System;

namespace \_1150080131\_TranDuc

{

public static class TH7

{

public static void Run()

{

double a = ReadPositiveDouble("Nhập độ dài cạnh a: ");

double b = ReadPositiveDouble("Nhập độ dài cạnh b: ");

double c = ReadPositiveDouble("Nhập độ dài cạnh c: ");

// Kiểm tra điều kiện tạo thành tam giác: a, b, c > 0 và

// tổng hai cạnh bất kỳ > cạnh còn lại

if (IsTriangle(a, b, c))

{

double p = a + b + c; // chu vi

double s = p / 2.0; // nửa chu vi

double area = Math.Sqrt(s \* (s - a) \* (s - b) \* (s - c)); // Heron

Console.WriteLine($"Chu vi tam giác: {p}");

Console.WriteLine($"Diện tích tam giác: {area}");

}

else

{

Console.WriteLine("Ba đoạn thẳng KHÔNG lập thành tam giác (vi phạm bất đẳng thức tam giác).");

}

}

private static bool IsTriangle(double a, double b, double c)

{

return a > 0 && b > 0 && c > 0

&& a + b > c

&& a + c > b

&& b + c > a;

}

private static double ReadPositiveDouble(string prompt)

{

while (true)

{

Console.Write(prompt);

if (double.TryParse(Console.ReadLine(), out double value) && value > 0)

return value;

Console.WriteLine("Giá trị không hợp lệ. Vui lòng nhập số thực dương.");

}

}

}

}

**Thực hành 8:** Viết chương trình giải phương trình bậc 2: ax2 + bx + c = 0.

****

using System;

namespace \_1150080131\_TranDuc

{

public static class TH8

{

public static void Run()

{

Console.Write("Nhập a: ");

double a = double.Parse(Console.ReadLine());

Console.Write("Nhập b: ");

double b = double.Parse(Console.ReadLine());

Console.Write("Nhập c: ");

double c = double.Parse(Console.ReadLine());

if (a == 0)

{

// Khi a = 0 thì phương trình trở thành bx + c = 0

if (b == 0)

{

if (c == 0)

Console.WriteLine("Phương trình có vô số nghiệm.");

else

Console.WriteLine("Phương trình vô nghiệm.");

}

else

{

double x = -c / b;

Console.WriteLine($"Phương trình có một nghiệm: x = {x}");

}

}

else

{

double delta = b \* b - 4 \* a \* c;

if (delta < 0)

{

Console.WriteLine("Phương trình vô nghiệm (Δ < 0).");

}

else if (delta == 0)

{

double x = -b / (2 \* a);

Console.WriteLine($"Phương trình có nghiệm kép: x1 = x2 = {x}");

}

else

{

double x1 = (-b + Math.Sqrt(delta)) / (2 \* a);

double x2 = (-b - Math.Sqrt(delta)) / (2 \* a);

Console.WriteLine($"Phương trình có 2 nghiệm phân biệt:");

Console.WriteLine($"x1 = {x1}");

Console.WriteLine($"x2 = {x2}");

}

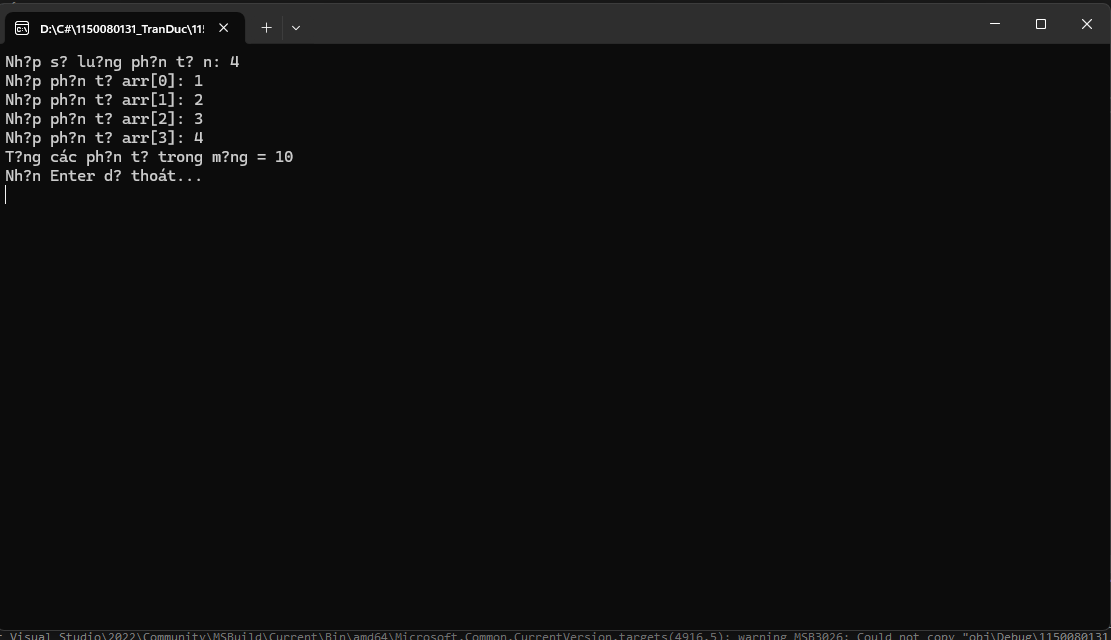
}

}

}

}

**Thực hành 9:** Tính tổng các phần tử trong mảng.

****

using System;

namespace \_1150080131\_TranDuc

{

public static class TH9

{

public static void Run()

{

Console.Write("Nhập số lượng phần tử n: ");

int n = ReadPositiveInt();

int[] arr = new int[n];

for (int i = 0; i < n; i++)

{

Console.Write($"Nhập phần tử arr[{i}]: ");

arr[i] = ReadInt();

}

int sum = 0;

foreach (int x in arr)

{

sum += x;

}

Console.WriteLine("Tổng các phần tử trong mảng = " + sum);

}

private static int ReadPositiveInt()

{

while (true)

{

if (int.TryParse(Console.ReadLine(), out int v) && v > 0)

return v;

Console.Write("Giá trị không hợp lệ. Nhập lại số nguyên dương: ");

}

}

private static int ReadInt()

{

while (true)

{

if (int.TryParse(Console.ReadLine(), out int v))

return v;

Console.Write("Giá trị không hợp lệ. Nhập lại số nguyên: ");

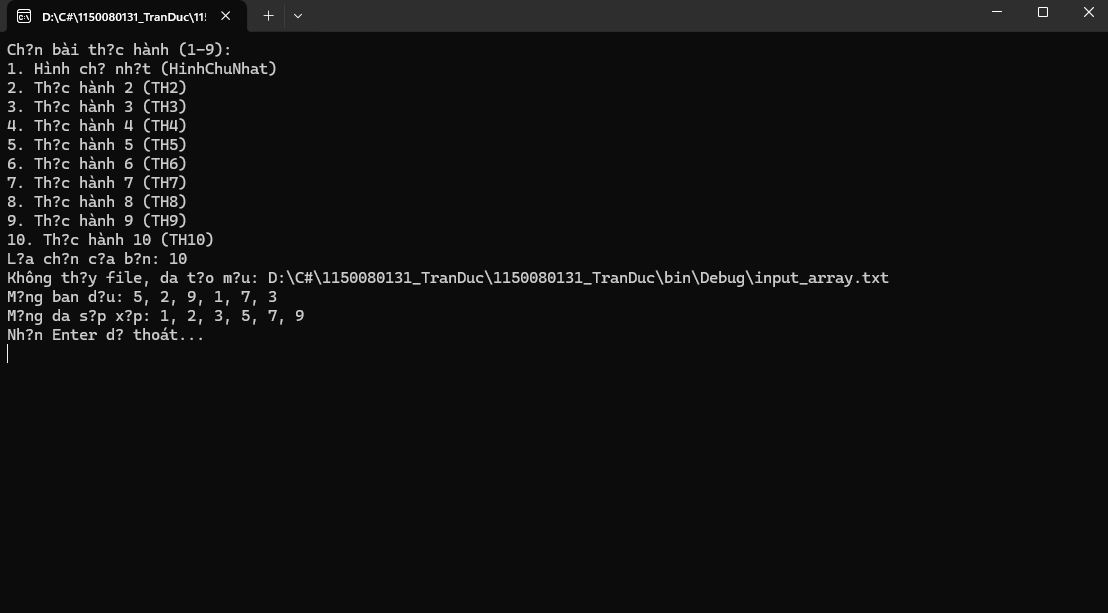
}

}

}

}

**Thực hành 10:** Dùng giải thuật sắp xếp chọn (Selection Sort) để sắp xếp tăng  dần mảng các số nguyên. Mảng các số nguyên được lưu trong tệp văn bản có tên  là “input\_array.txt”.

****

using System;

using System.IO;

using System.Linq;

namespace \_1150080131\_TranDuc

{

public static class TH10

{

public static void Run()

{

string path = Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "input\_array.txt");

if (!File.Exists(path))

{

// Tạo file mẫu nếu chưa có

File.WriteAllText(path, "5 2 9 1 7 3");

Console.WriteLine($"Không thấy file, đã tạo mẫu: {path}");

}

int[] arr = File.ReadAllText(path)

.Split(new[] { ' ', '\n', '\r', '\t' }, StringSplitOptions.RemoveEmptyEntries)

.Select(int.Parse)

.ToArray();

Console.WriteLine("Mảng ban đầu: " + string.Join(", ", arr));

SelectionSort(arr);

Console.WriteLine("Mảng đã sắp xếp: " + string.Join(", ", arr));

}

private static void SelectionSort(int[] a)

{

for (int i = 0; i < a.Length - 1; i++)

{

int min = i;

for (int j = i + 1; j < a.Length; j++)

if (a[j] < a[min]) min = j;

if (min != i) (a[i], a[min]) = (a[min], a[i]);

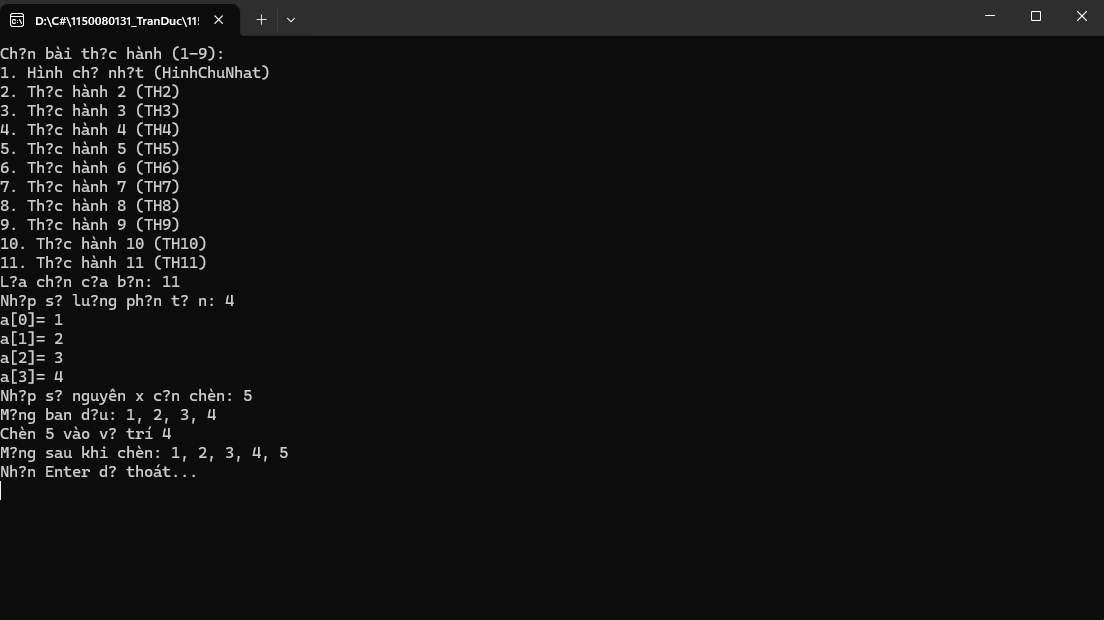
}

}

}

}

**Thực hành 11:** Chèn thêm một số nguyên được nhạp vào từ bàn phím vào mảng  đã sắp xếp tăng dần nhưng không làm mất tính tăng dần của mảng.

****

using System;

using System.Linq;

namespace \_1150080131\_TranDuc

{

public static class TH11

{

public static void Run()

{

int n = ReadPositiveInt("Nhập số lượng phần tử n: ");

// Nhập mảng tăng dần (nếu người dùng nhập chưa tăng dần, sẽ tự sắp xếp lại)

int[] a = new int[n];

for (int i = 0; i < n; i++)

{

a[i] = ReadInt($"a[{i}]= ");

}

// Đảm bảo mảng tăng dần (nếu đề yêu cầu chắc chắn tăng dần, có thể bỏ dòng này)

if (!IsNonDecreasing(a))

{

Array.Sort(a);

Console.WriteLine("Mảng nhập chưa tăng dần, đã tự sắp xếp lại.");

}

Console.Write("Nhập số nguyên x cần chèn: ");

int x = ReadInt();

// Tìm vị trí chèn bằng BinarySearch

int idx = Array.BinarySearch(a, x);

int insertPos = (idx >= 0) ? idx + 1 : ~idx; // chèn sau các phần tử bằng x

// Tạo mảng mới và chèn

int[] b = new int[n + 1];

Array.Copy(a, 0, b, 0, insertPos);

b[insertPos] = x;

Array.Copy(a, insertPos, b, insertPos + 1, n - insertPos);

Console.WriteLine("Mảng ban đầu: " + string.Join(", ", a));

Console.WriteLine($"Chèn {x} vào vị trí {insertPos}");

Console.WriteLine("Mảng sau khi chèn: " + string.Join(", ", b));

}

private static bool IsNonDecreasing(int[] a)

{

for (int i = 1; i < a.Length; i++)

if (a[i] < a[i - 1]) return false;

return true;

}

private static int ReadPositiveInt(string prompt)

{

while (true)

{

Console.Write(prompt);

if (int.TryParse(Console.ReadLine(), out int v) && v > 0) return v;

Console.WriteLine("Giá trị không hợp lệ. Nhập lại số nguyên dương.");

}

}

private static int ReadInt(string prompt = null)

{

while (true)

{

if (!string.IsNullOrEmpty(prompt)) Console.Write(prompt);

if (int.TryParse(Console.ReadLine(), out int v)) return v;

Console.WriteLine("Giá trị không hợp lệ. Nhập lại số nguyên.");

}

}

}

}