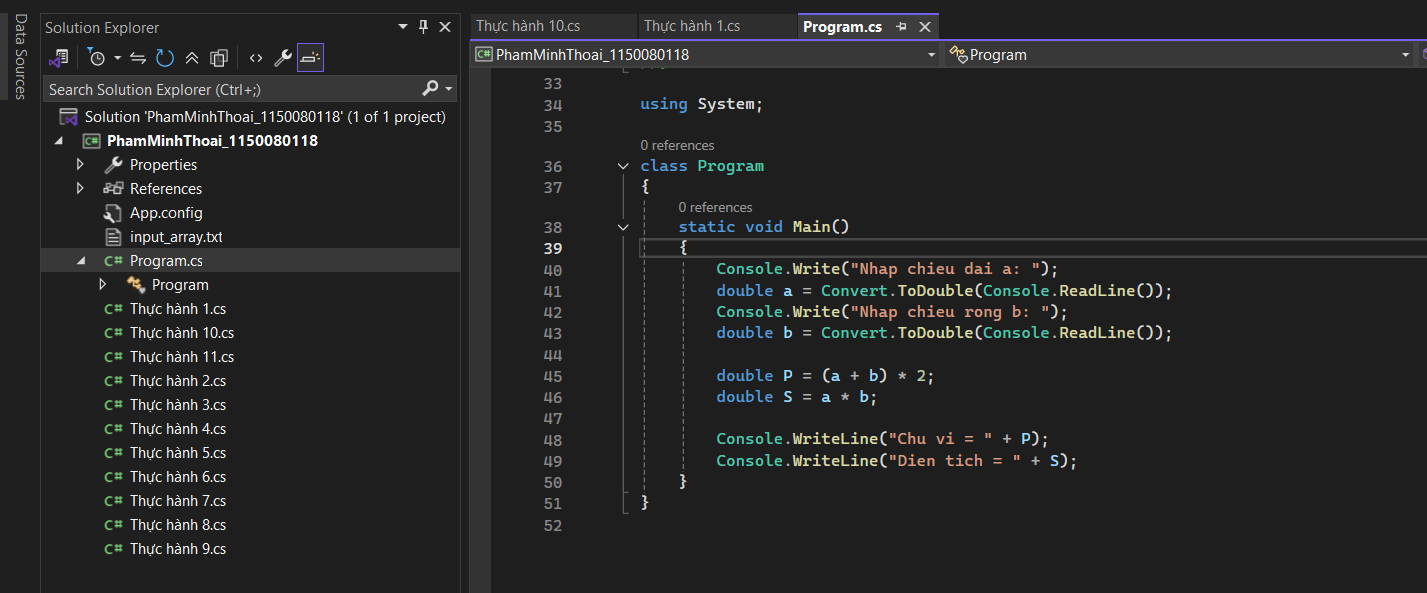
Họ và tên: **Phạm Minh Thoại**

MSSV:**1150080101**

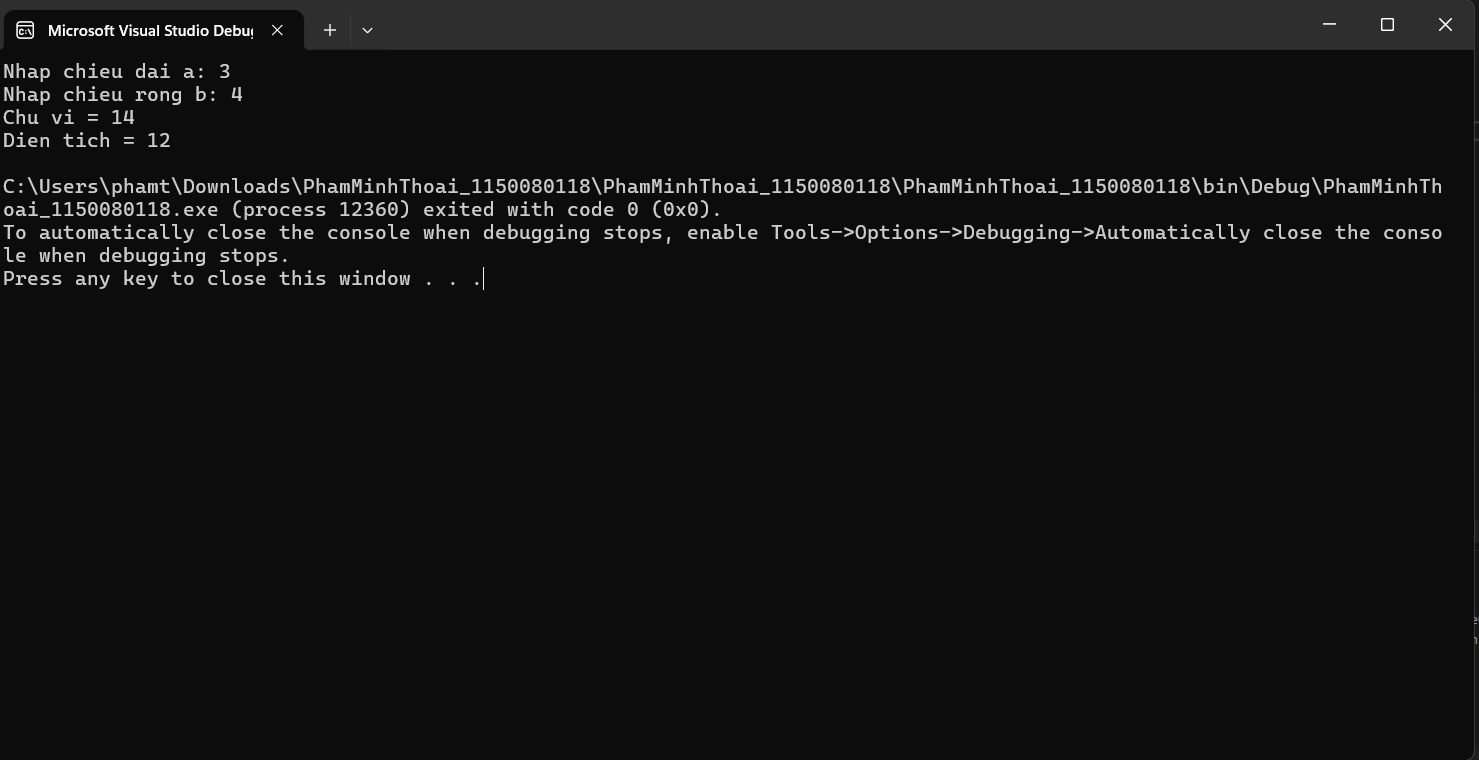
Lớp: **11\_ĐH\_CNPM2**

**LAB 1: NỀN TẢNG NGÔN NGỮ LẬP TRÌNH C#**

**Thực hành 1:** Tính chu vi và diện tích hình chữ nhật



**Kết quả:**



**Thực hành 2:** Tìm số lớn hơn trong 2 số nguyên

// Tìm max trong 2 số nguyên a, b

// Nhập dữ liệu

Console.Write(" hap vao so nguyen a: ");

int a = Convert.ToInt32(Console.ReadLine());

Console.Write(" hap vao so nguyen b: ");

int b = Convert.ToInt32(Console.ReadLine());

// Tìm số max

int max = 0;

if (a > b)

{

max = a;

}

if (b > a)

{

max = b;

}

// Hiển thị kết quả

Console.WriteLine("So lon hon trong 2 so la: " + max);

**7.1.2. Lệnh điều kiện if – đủ**

Cú pháp như sau:

if(<biểu thức điều kiện>)

{

}

else

{

}

Tập các câu lệnh của if;

Tập các câu lệnh của else;

Cách thức hoạt động:

− Đầu tiên, ta xác định biểu thức điều kiện nhận giá trị True hoặc False.

− Nếu biểu thức điều kiện có giá trị True thì thực hiện Tập các câu lệnh của

if.

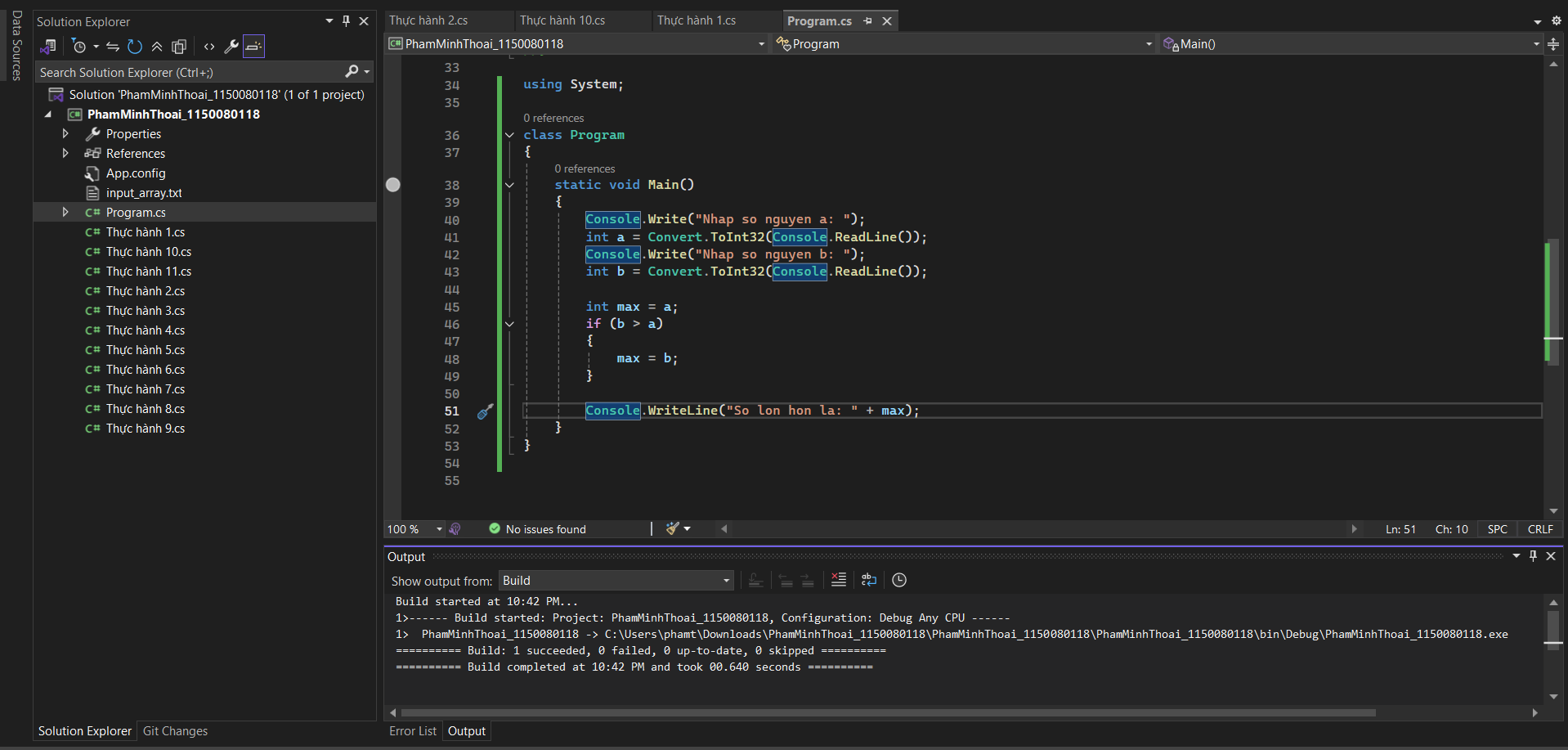
Học để chiến – Học để làm - Học để nuôi ta & nuôi bồ 9

− Còn nếu biểu thức điều kiện có giá trị là False thì thực thi Tập các câu lệnh

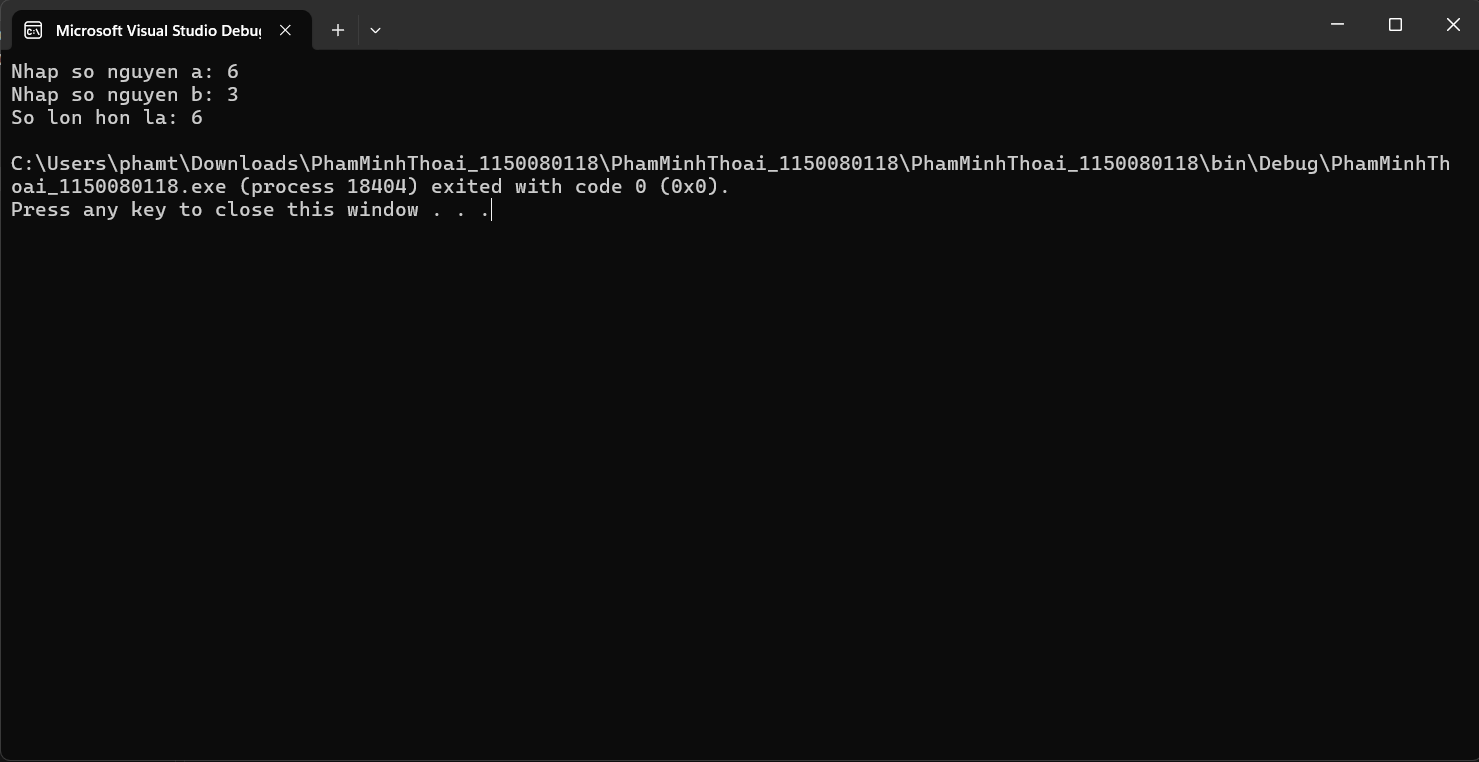
của else.

Tức là ở dạng if – đủ, ta sẽ kiểm tra cả 2 trường hợp: ĐÚNG thì làm những gì và

SAI thì làm những gì.



**Kết quả:**



**Thực hành 3:** Tìm số lớn nhất trong 3 số nguyên

// Tìm max trong 3 số nguyên a, b, c

// Nhập dữ liệu

Console.Write(" hap vao so nguyen a: ");

int a = Convert.ToInt32(Console.ReadLine());

Console.Write(" hap vao so nguyen b: ");

int b = Convert.ToInt32(Console.ReadLine());

Console.Write(" hap vao so nguyen c: ");

int c = Convert.ToInt32(Console.ReadLine());

// Tìm số max

int max = 0;

if ((a > b) fifi (a > c))

{

max = a;

}

else if ((b > a) fifi (b > c))

{

}

else

{

}

max = b;

max = c;

// iển thị kết quả

Console.WriteLine("So lon nhat trong 3 so la: {0}", max);

**7.1.3. Lệnh điều kiện if – liệt kê**

int diem = 8;

if (diem > 10 || diem < 0)

{

Console.WriteLine("Diem khong hop le!");

}

else if (diem >= 9)

{

Console.WriteLine("Xuat sac!");

}

else if (diem >= 8)

{

Console.WriteLine("Gioi!");

}

else if (diem >= 6.5)

{

Console.WriteLine("Kha!");

}

else if (diem >= 5)

{

Console.WriteLine("Trung binh!");

}

else if (diem >= 3.5)

{

}

else

{

}

Console.WriteLine("Yeu!");

Console.WriteLine("Kem!");

**7.2. Lệnh lựa chọn (switch ... case)**

Nếu để ý kỹ, bạn sẽ thấy if có những lúc hoạt động trên một tập các giá trị liên

tục. Tuy nhiên, lệnh lựa chọn switch ... case lại chỉ hoạt động với những giá trị

rời rạc mà thôi.

Cú pháp:

case(biến\_số\_nguyên)

{

case 1:

lệnh của case 1;

break;

case 2:

lệnh của case 2;

break;

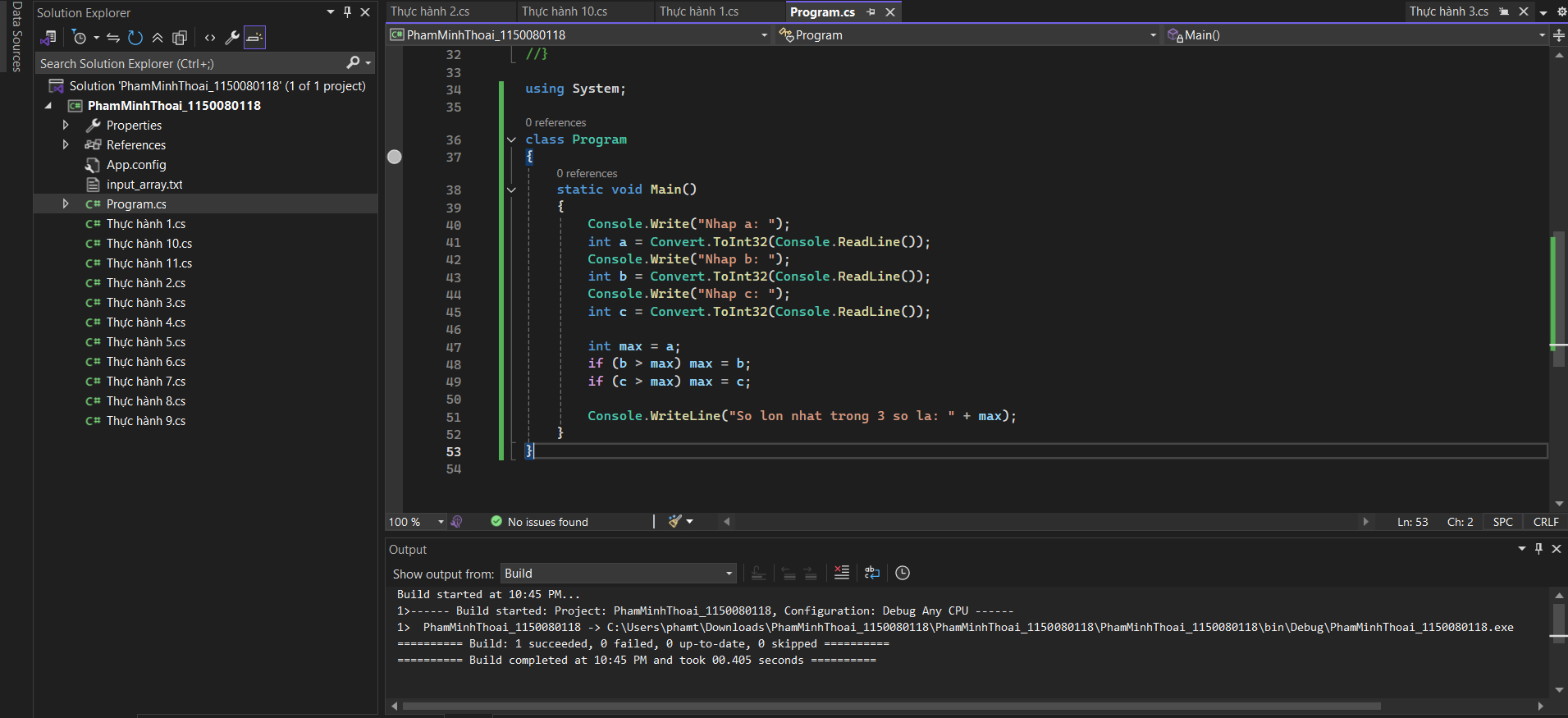
...

default:

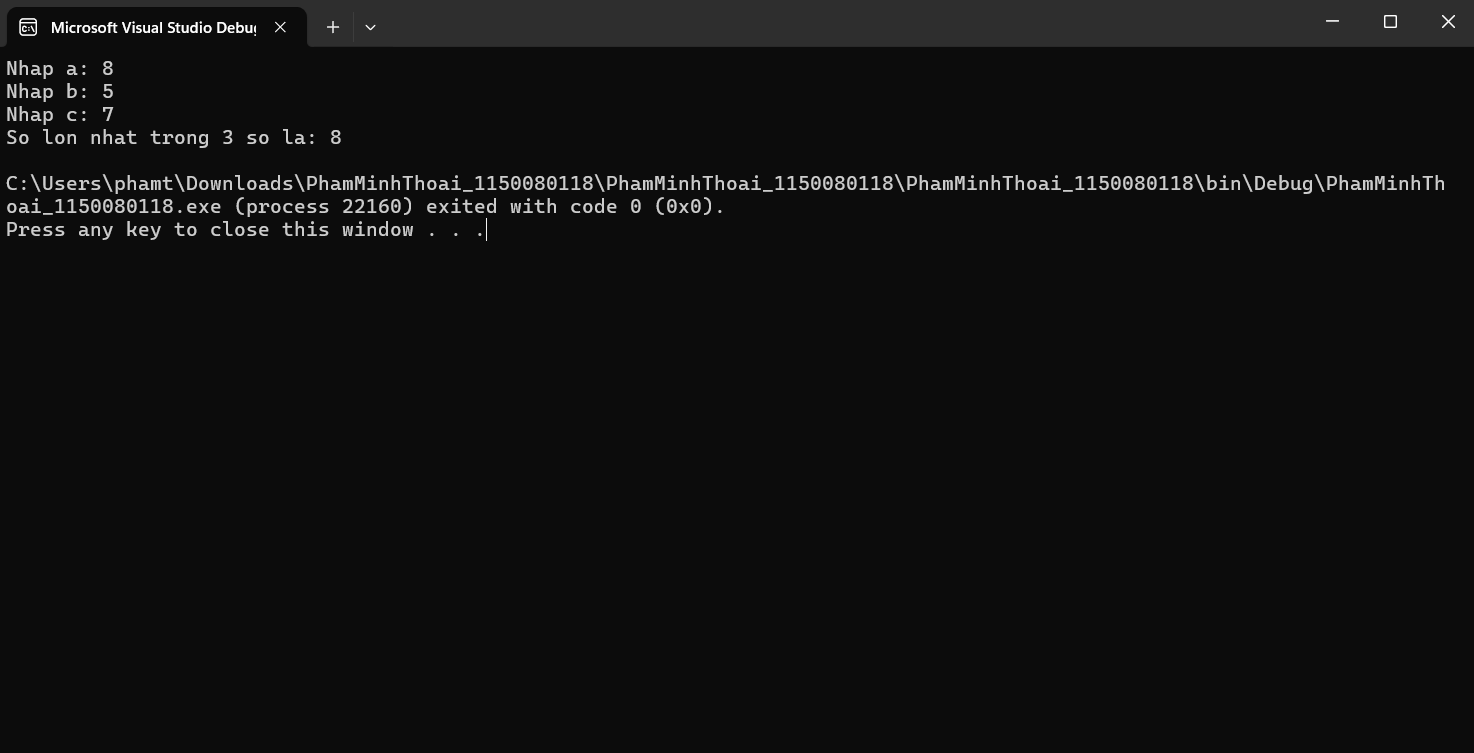
lệnh của default;

break;

}



**Kết quả:**

****

**Thực hành 4:** Cho biết một tháng có bao nhiêu ngày

// Tháng có bao nhiêu ngày

// hập dữ liệu

Console.Write(" hap vao nam: ");

int nam = Convert.ToInt32(Console.ReadLine());

Console.Write(" hap vao thang: ");

int thang = Convert.ToInt32(Console.ReadLine());

// Tìm số ngày trong tháng

switch(thang)

{

case 1:

case 3:

case 5:

case 7:

case 8:

case 10:

case 12:

Console.WriteLine("Thang co 31 ngay!");

break;

case 4:

case 6:

case 9:

case 11:

Console.WriteLine("Thang co 30 ngay!");

break;

case 2:

if ((nam % 400 == 0) || ((nam % 4 == 0) fifi (nam % 100 != 0)))

{

}

else

{

}

Console.WriteLine("Thang co 29 ngay!");

Console.WriteLine("Thang co 28 ngay!");

break;

}

**7.3. Lệnh lặp**

**7.3.1. Lệnh lặp for**

Cú pháp:

for(biểu thức 1; biểu thức 2; biểu thức 3)

Các câu lệnh;

Cách thức hoạt động:

− Đầu tiên, biểu thức 1 sẽ thực hiện khởi tạo các giá trị ban đầu.

− Tiếp đến kiểm tra điều kiện trong biểu thức 2.

− Nếu biểu thức điều kiện đúng thì thực hiện các câu lệnh. Ngược lại, dừng

vòng lặp for.

− Cuối cùng, các giá trị cần thay đổi cho vòng lặp mới sẽ được thực hiện

trong biểu thức 3. Và lại kiểm tra điều kiện ở biểu thức 2.

Ví dụ: tính tổng các số từ 1 đến 100.

int s = 0;

for(int i=1; i<=100; i++)

{

s += 1;

}

**7.3.2. Lệnh lặp while**

Cú pháp:

while(biểu thức điều kiện)

Các câu lệnh;

Cách thức hoạt động:

Kiểm tra biểu thức điều kiện có nhận giá trị True hay không. Các câu lệnh được

thực hiện lặp đi lặp lại cho đến khi nào biểu thức điều kiện mang giá trị False thì

dừng lại.

Ví dụ:

int i = 1, s = 0;

while (i <= 100)

{

s += i;

i++;

}

**7.3.3. Lệnh lặp do ... while**

Cú pháp:

do

{

Các câu lệnh;

}

while(biểu thức điều kiện);

Cách thức hoạt động:

Đầu tiên, thực hiện các câu lệnh.

Tiếp đến, kiểm tra điều kiện. Nếu điều kiện có giá trị True thì tiếp tục thực thi các

lệnh lần tiếp theo. Ngược lại, dừng vòng lặp do ... while.

Ví dụ:

int i = 1, s = 0;

do

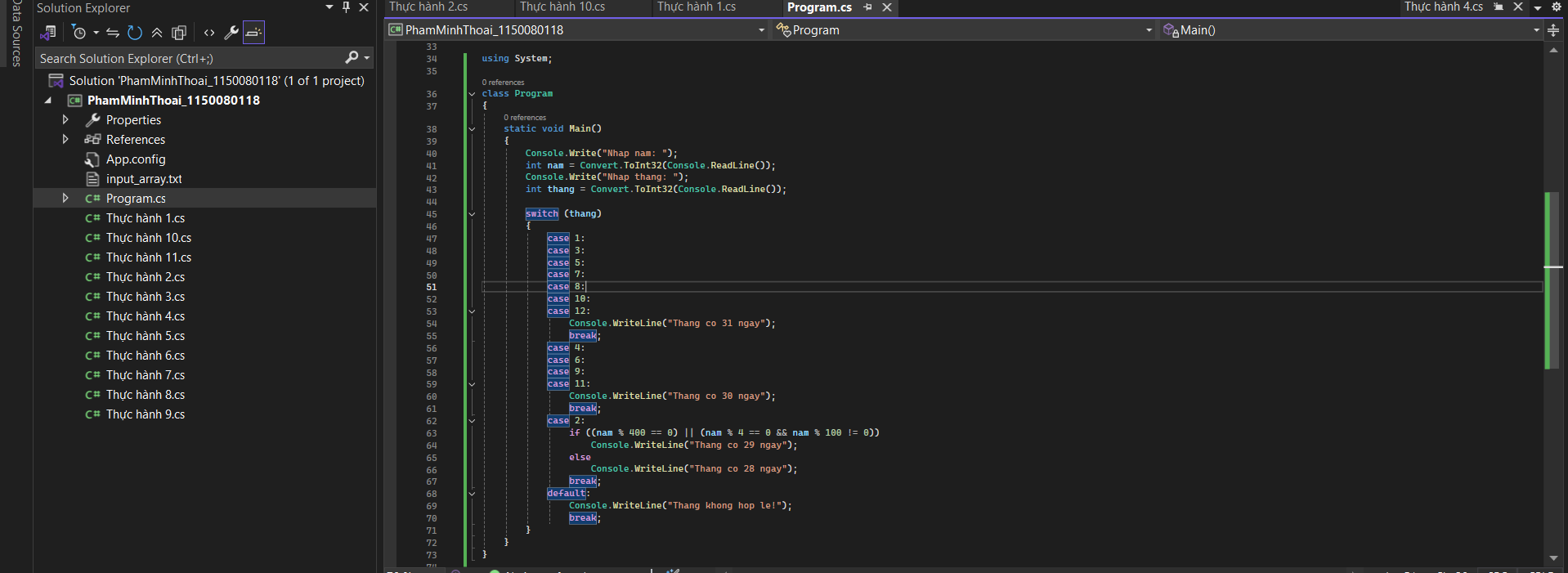
{

s += i;

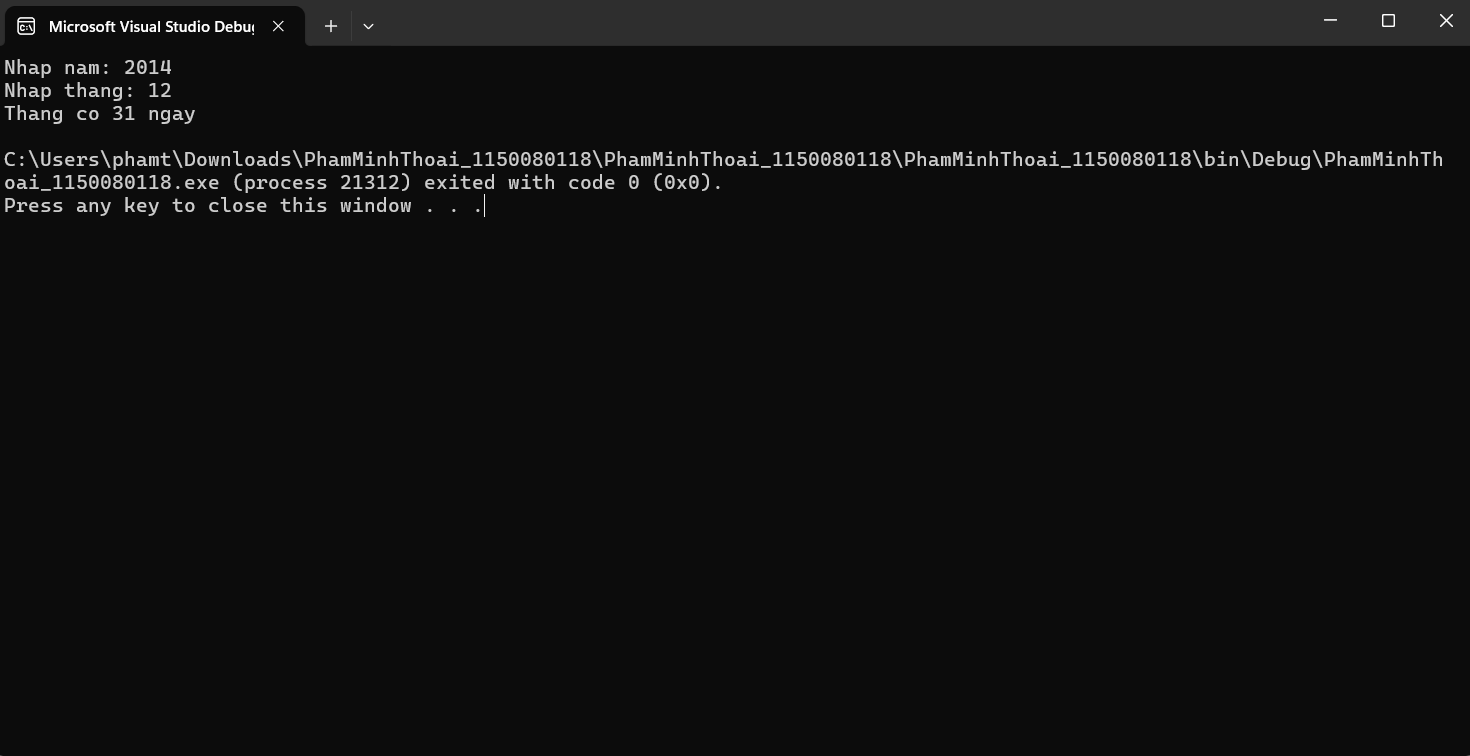
i++;

}

while (i <= 100);



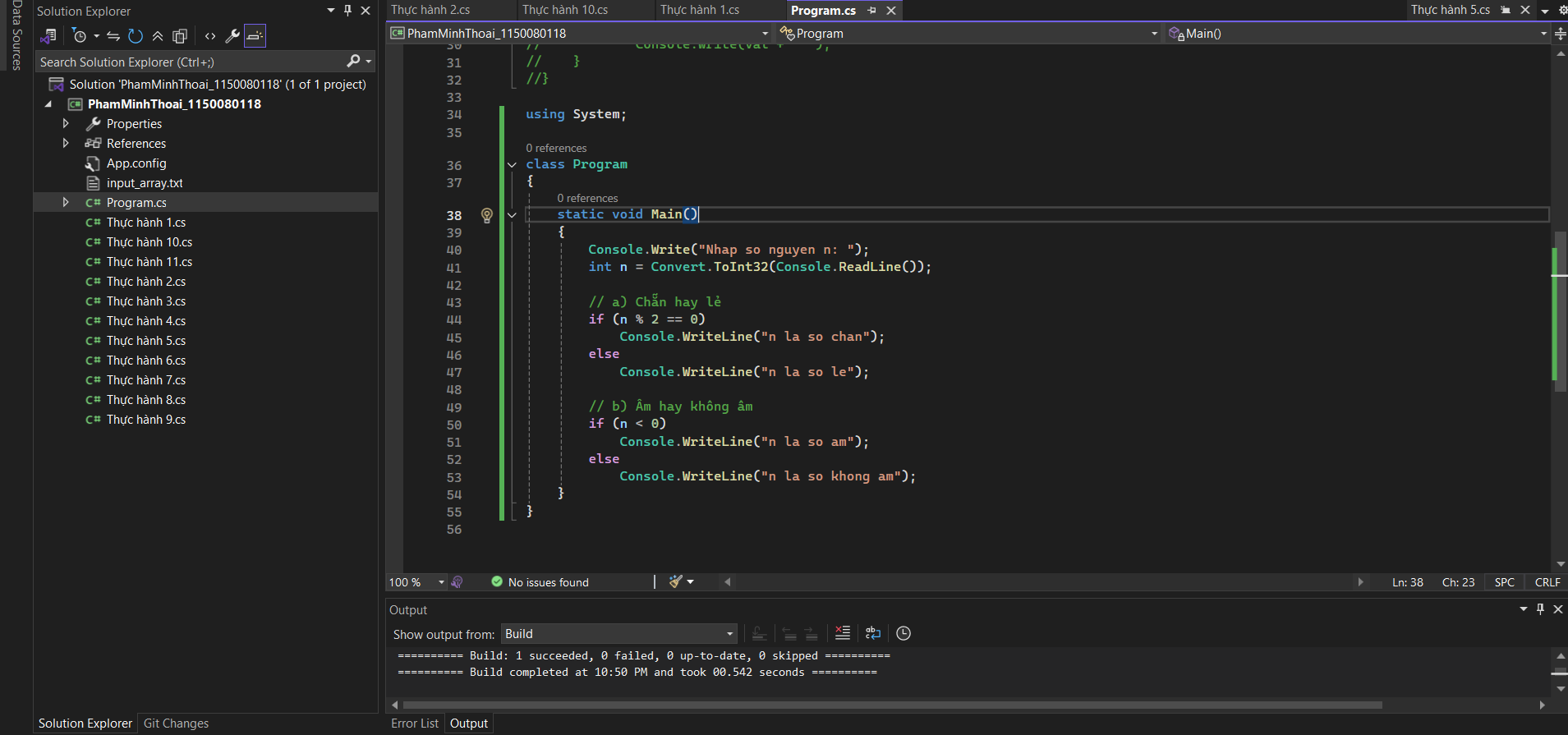
**Kết quả:**

****

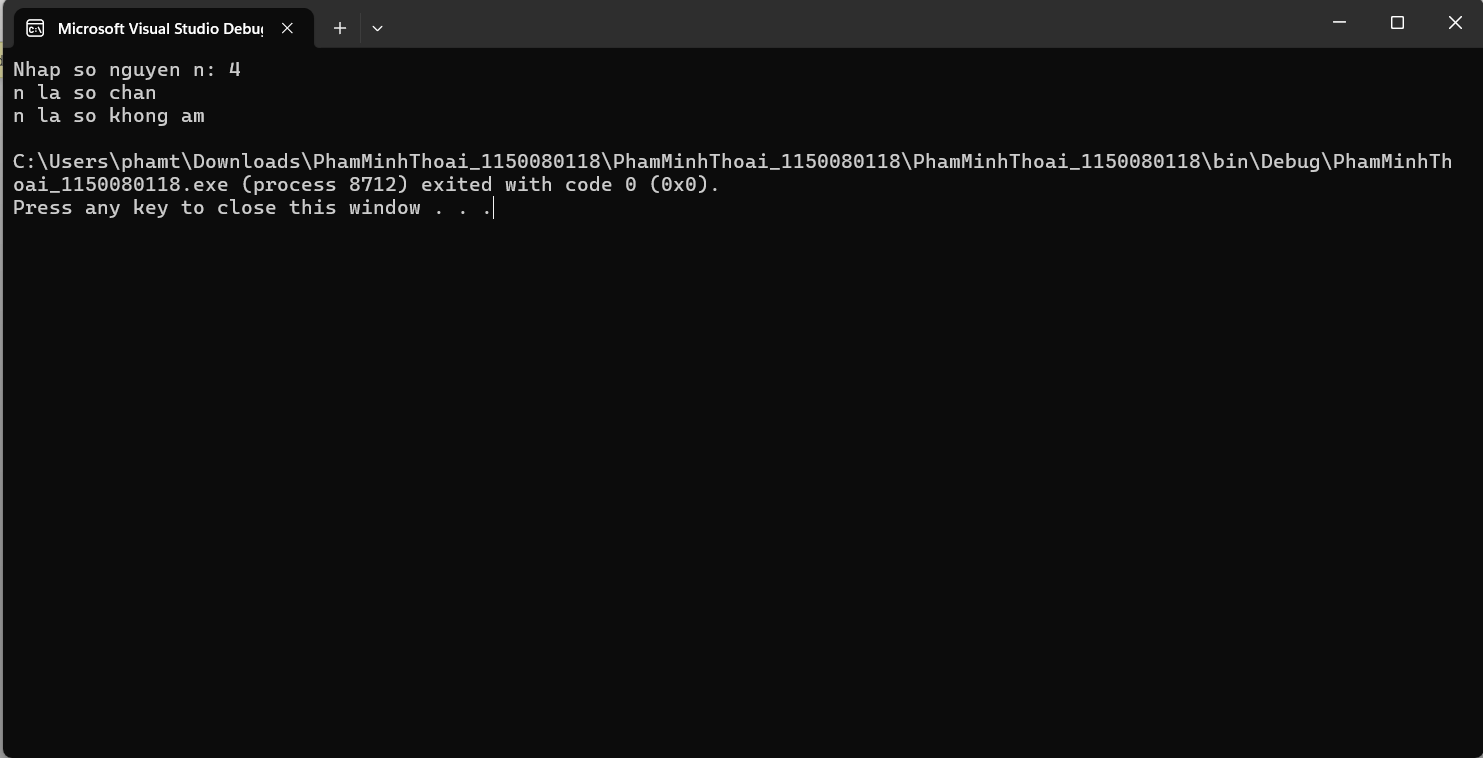
**Thực hành 5:** Viết chương trình nhập vào một số nguyên n. Cho biết:

a) n là số chẵn hay số lẻ?

b) n là số âm hay số không âm?

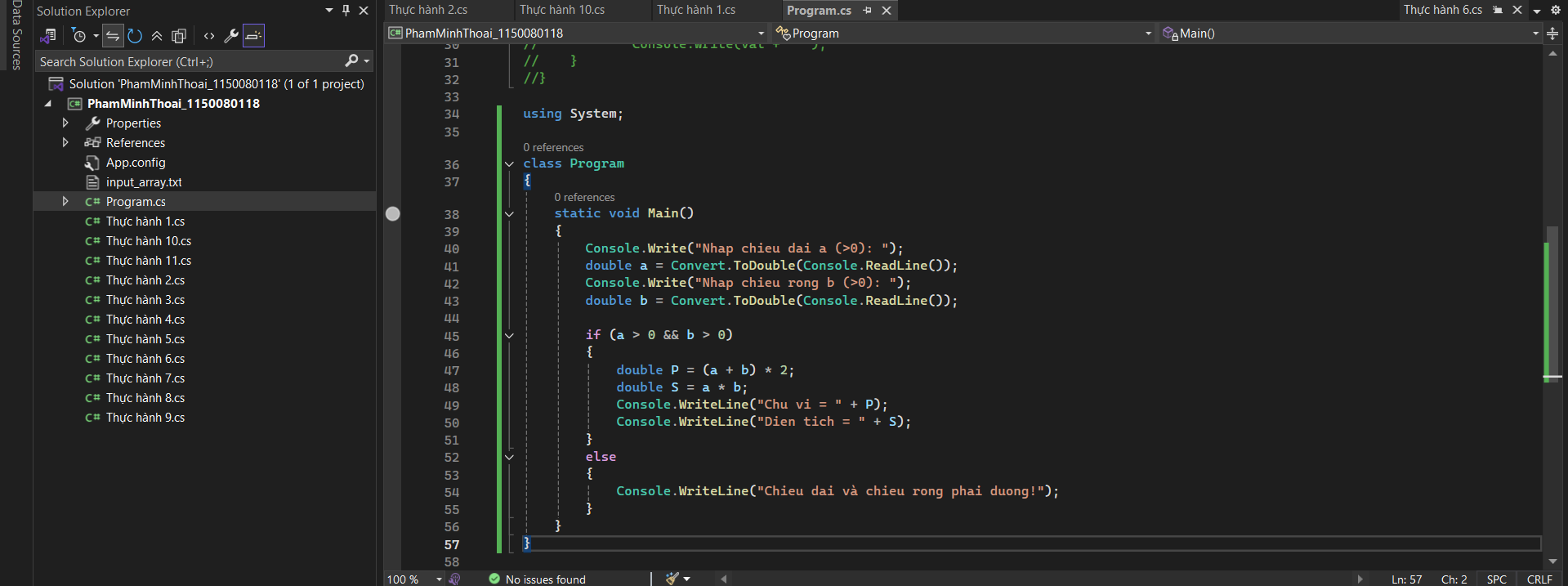


**Kết quả:**

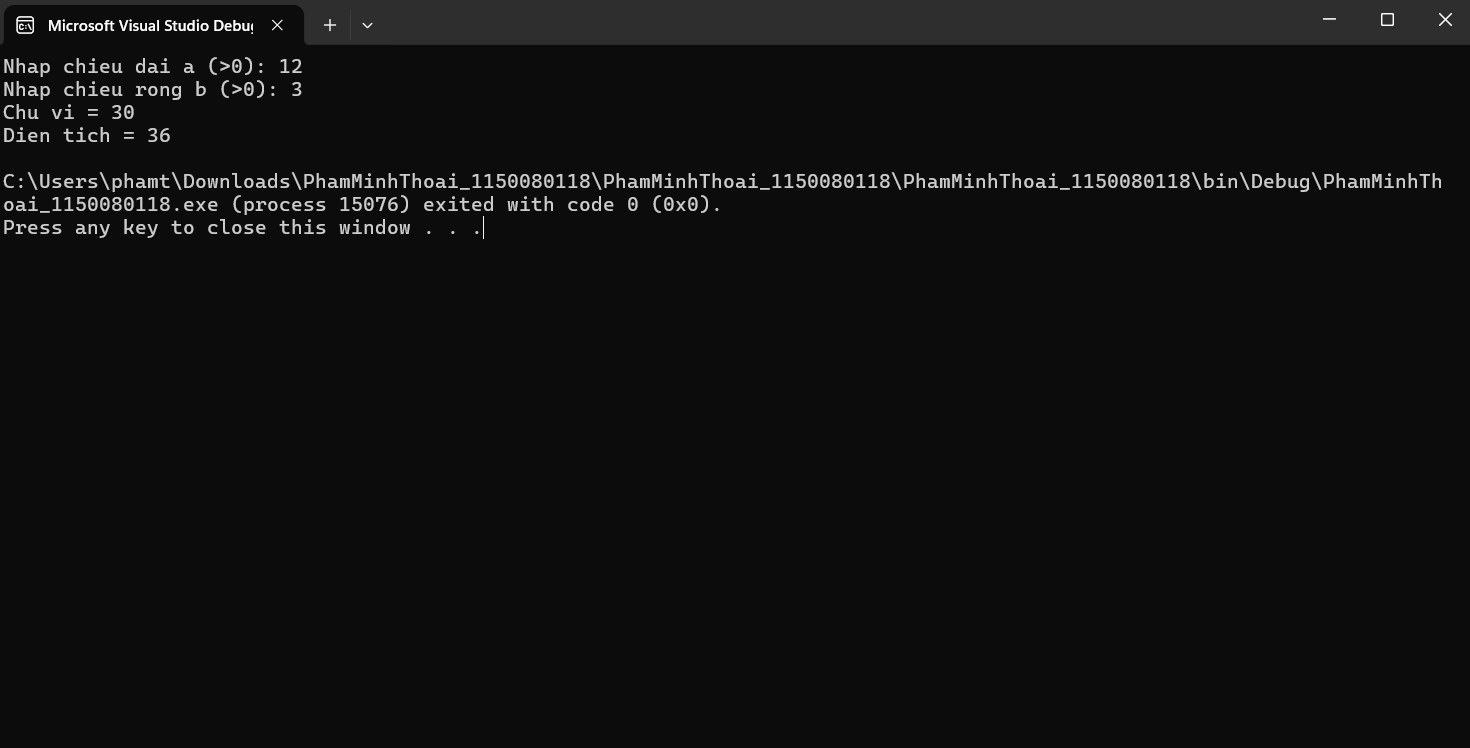
****

**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.



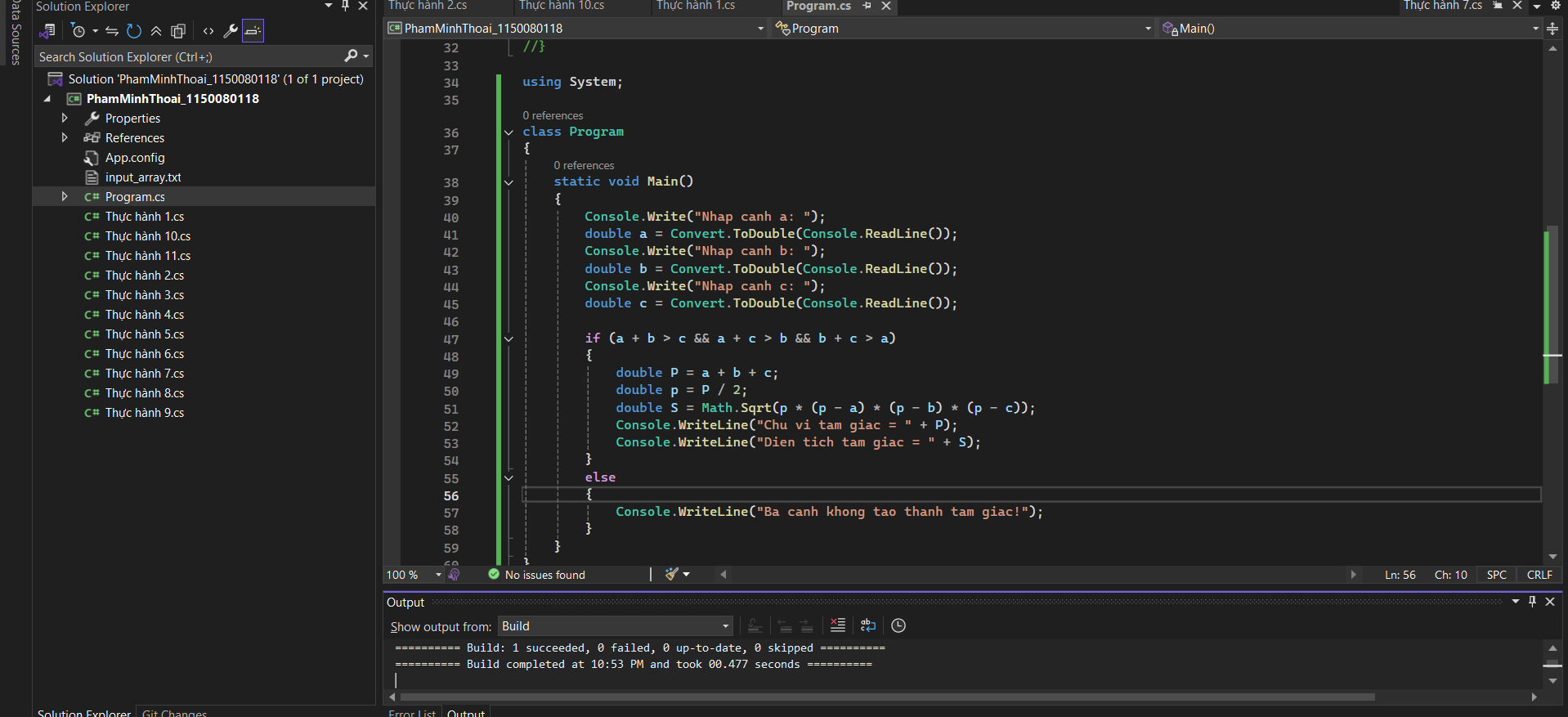
**Kết quả:**

****

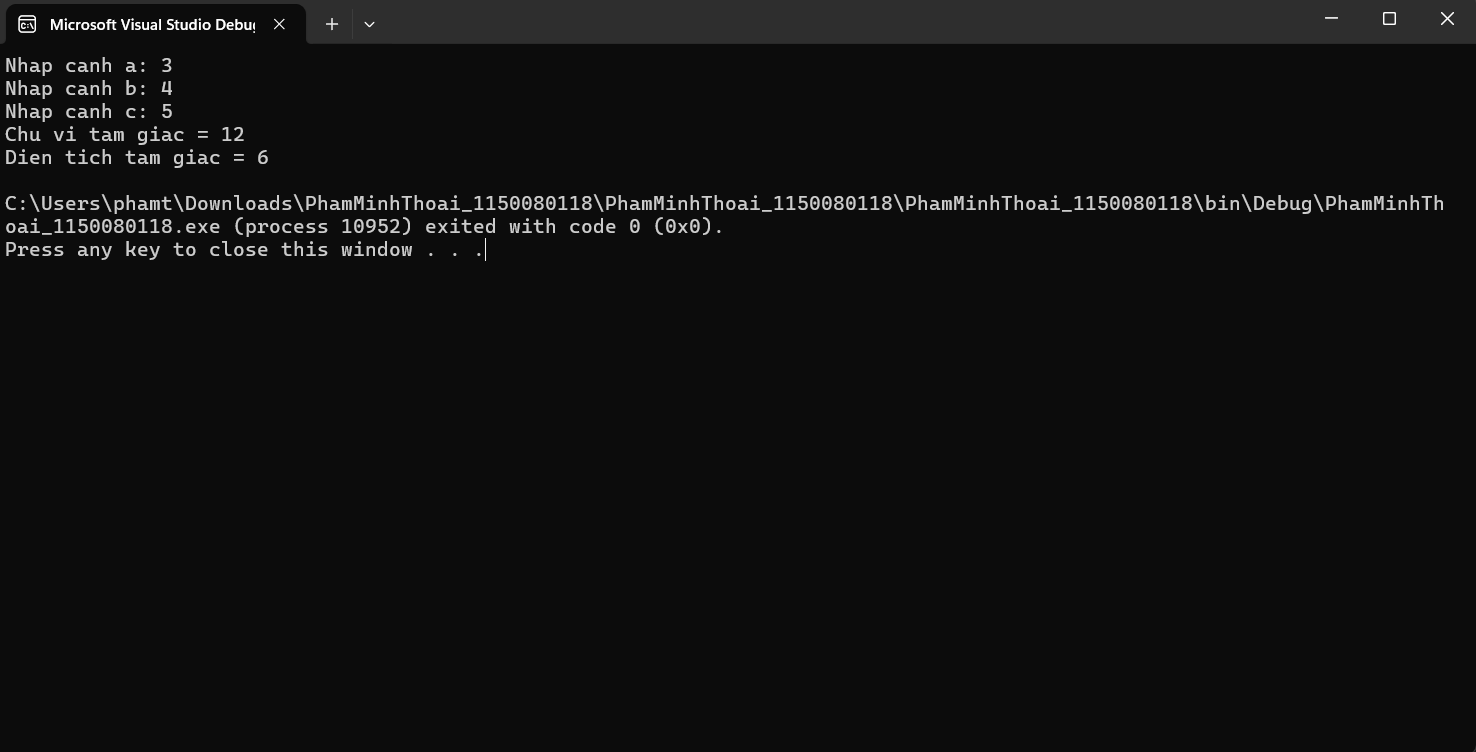
**Thực hành 7:** Viết chương trình nhập vào ba số thực chỉ độ dài của ba đoạn

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.



**Kết quả:**

****

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

**8. Kiểu mảng (array) trong C#**

**8.1. Mảng trong C# là gì?**

− Mảng (array) trong C# là một đối tượng cho phép chứa các phần tử có kiểu

dữ liệu giống nhau, kích thước của mảng là cố định.

− Đặc trưng của mảng là cặp dấu ngoặc vuông [ và ]

− Về cách thức sử dụng. Các phần tử trong mảng được truy cập trực tiếp

thông qua chỉ số. Mỗi một phần tử đều có một chỉ số riêng biệt. Chỉ số này

có miền chạy từ 0 đến (n - 1), trong đó n là kích thước của mảng.

**8.2. Khai báo và khởi tạo mảng trong C#**

Ví dụ: để khai báo mảng arr có số lượng phần tử là 5, các phần tử có kiểu dữ liệu

là int.

int[] arr = new int[5];

− Câu lệnh trên đã tạo ra mảng arr có 5 phần tử là số 0.

− Vừa khai báo và vừa khởi tạo mảng trong C#, ta lập trình:

int[] arr = { 3, 5, 1, 4, 9 };

int[] arr2 = new int[5] { 3, 5, 1, 4, 9 };

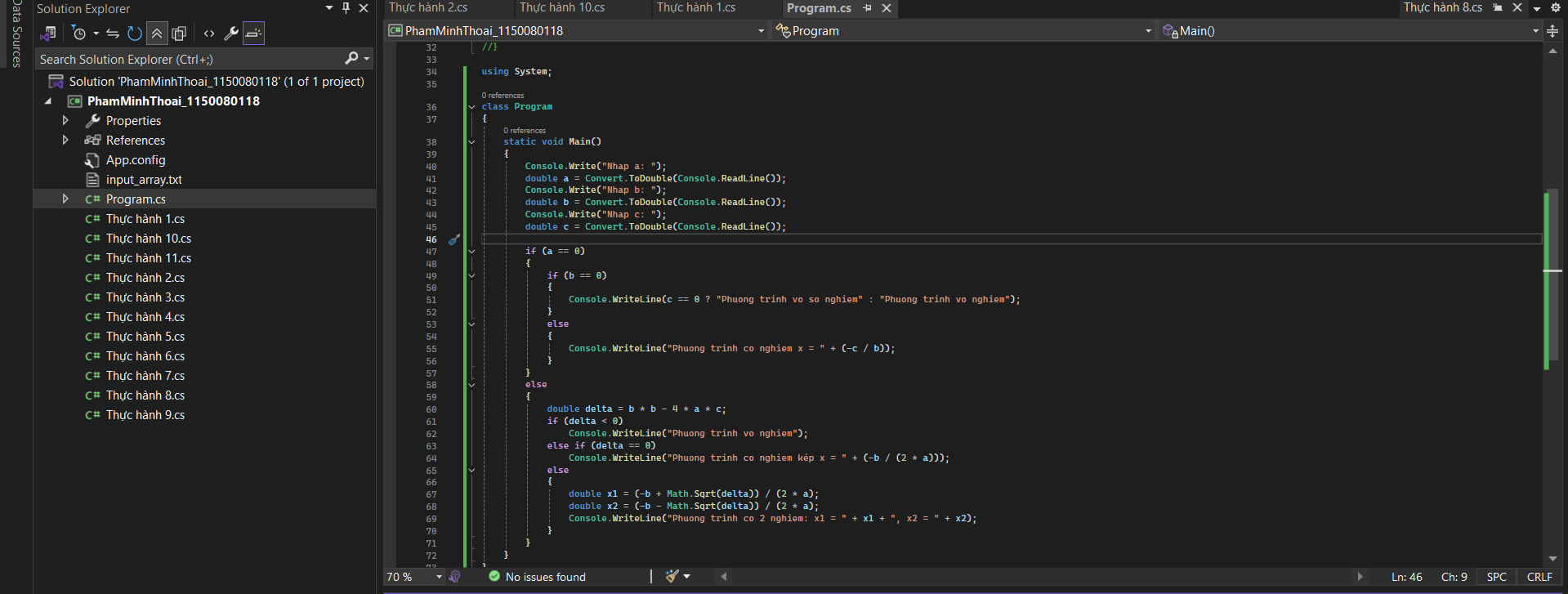
int[] arr3 = new int[] { 3, 5, 1, 4, 9 };

8.3. Một số tiện ích trong các thư viện C# hỗ trợ sử dụng mảng

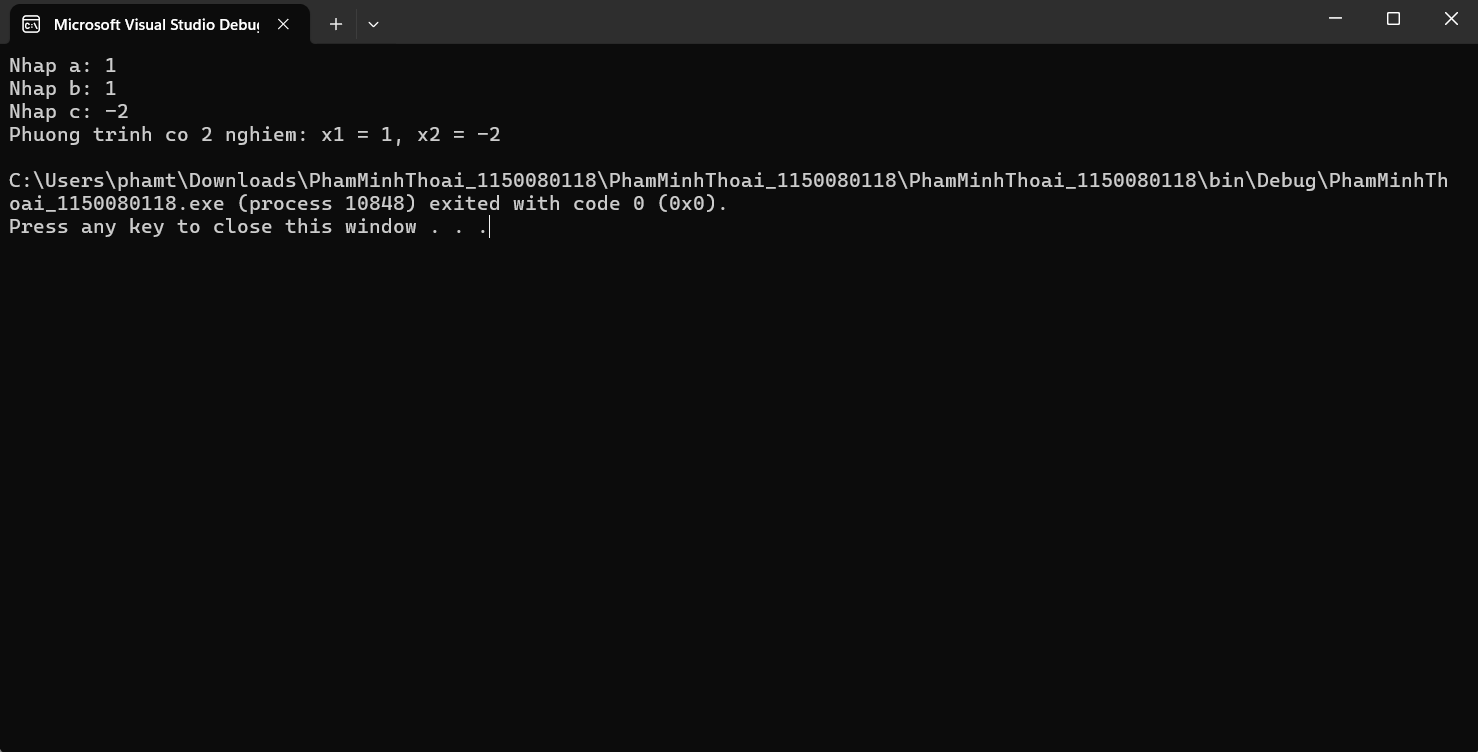
− Để xác định kích thước hay số lượng các phần tử của mảng arr. Ta dùng

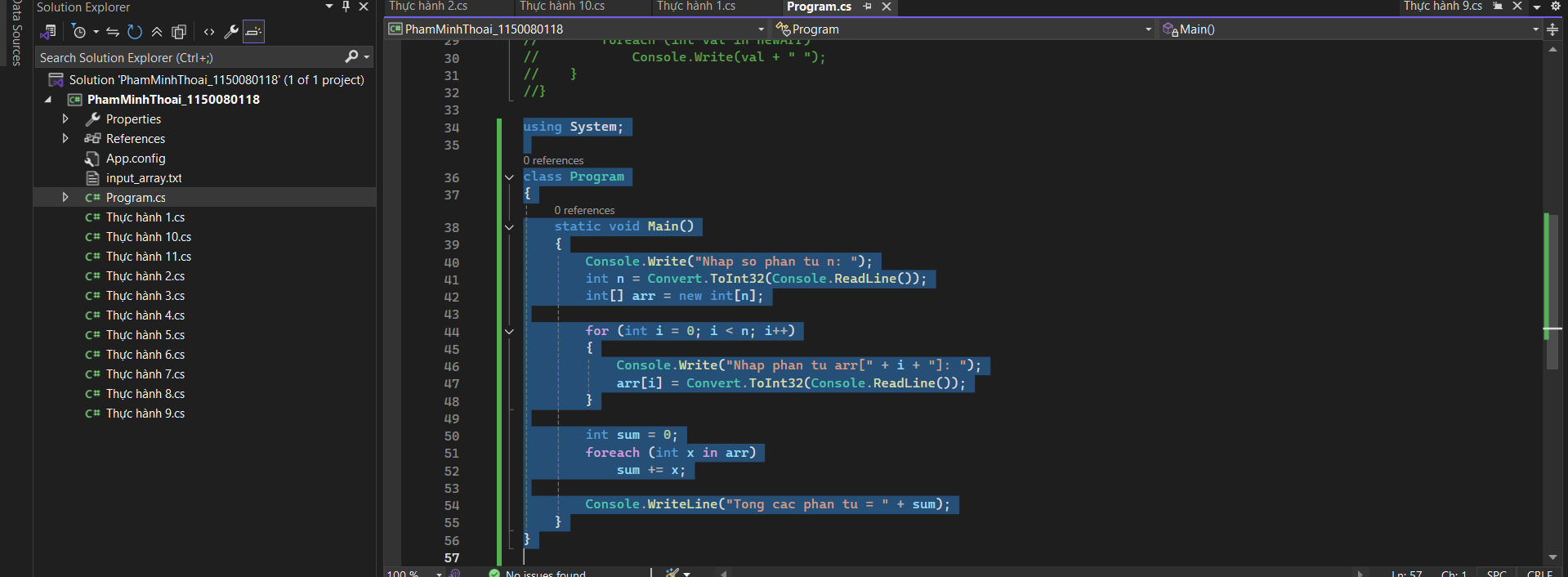
hàm Lengh.

− Để sắp xếp mảng theo thứ tự tăng dần, ta dùng Array.Sort(arr)

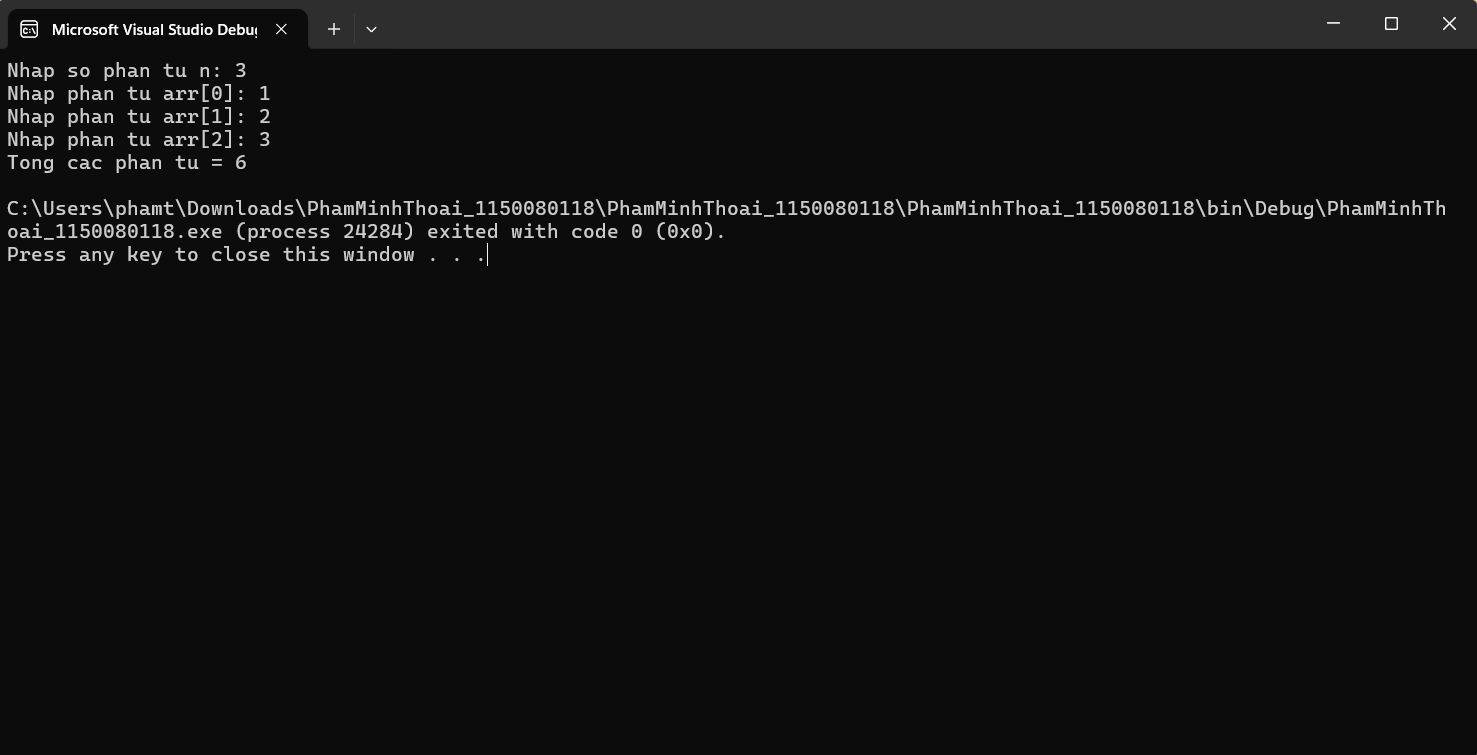
− Sử dụng System.Linq, ta có nhiều phương thức xử lý mảng hữu ích khác như Min, Max, Sum

**Kết quả:**

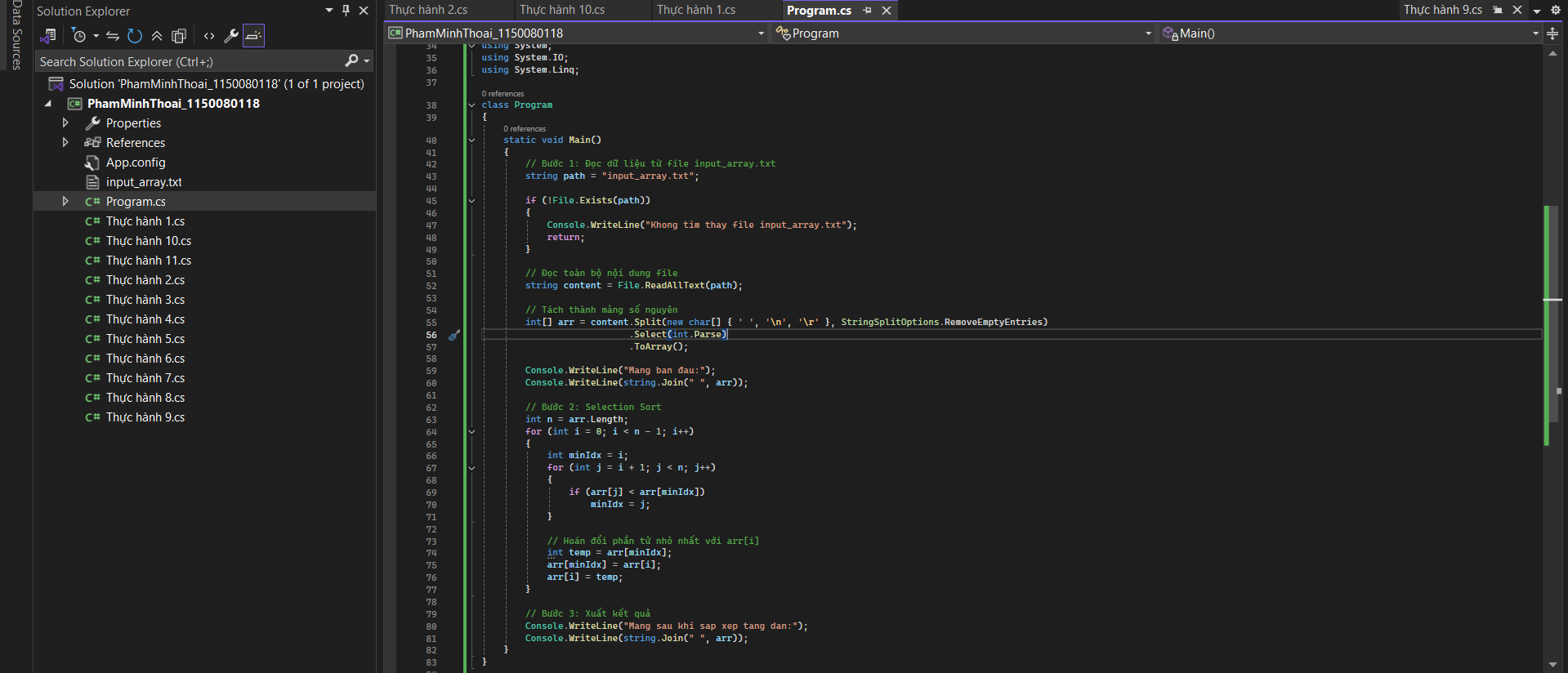
****

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

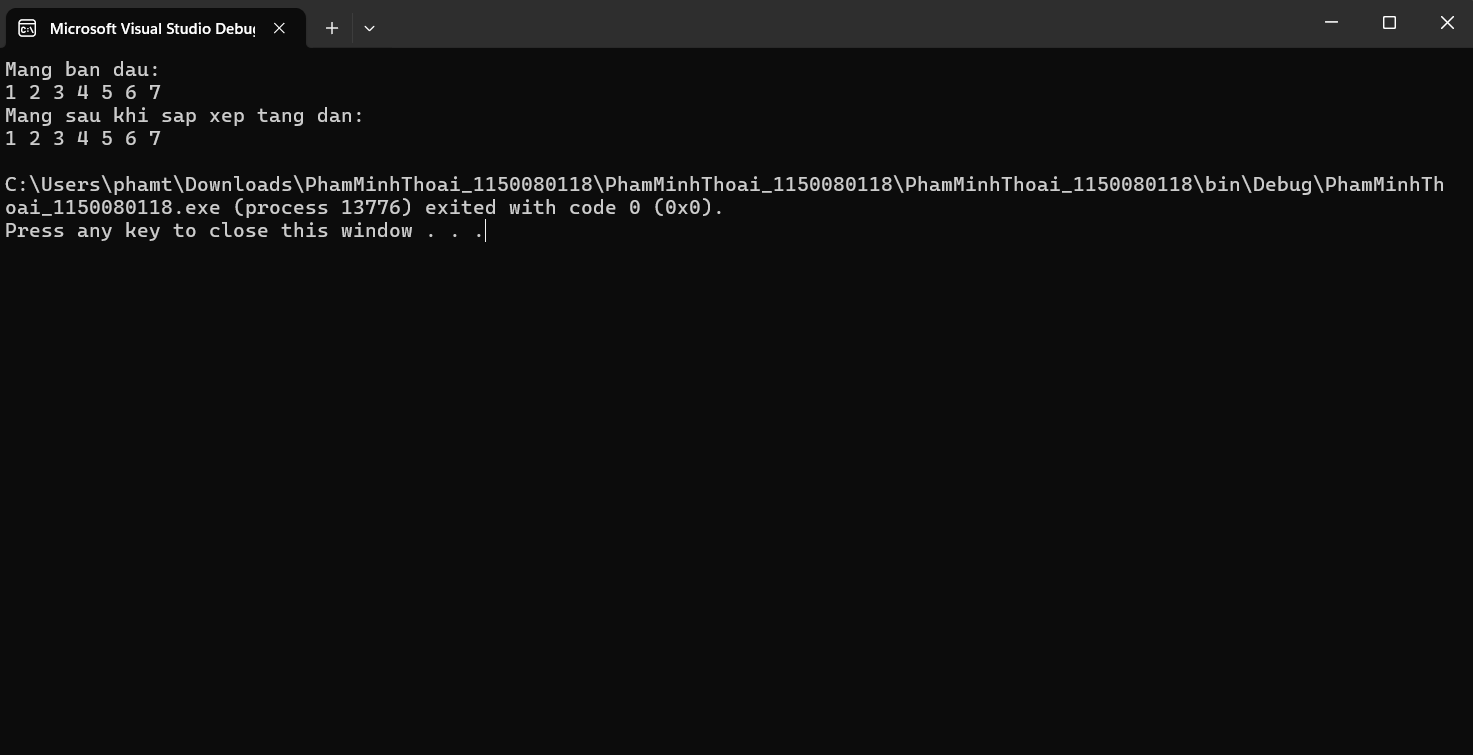
**Kết quả:**

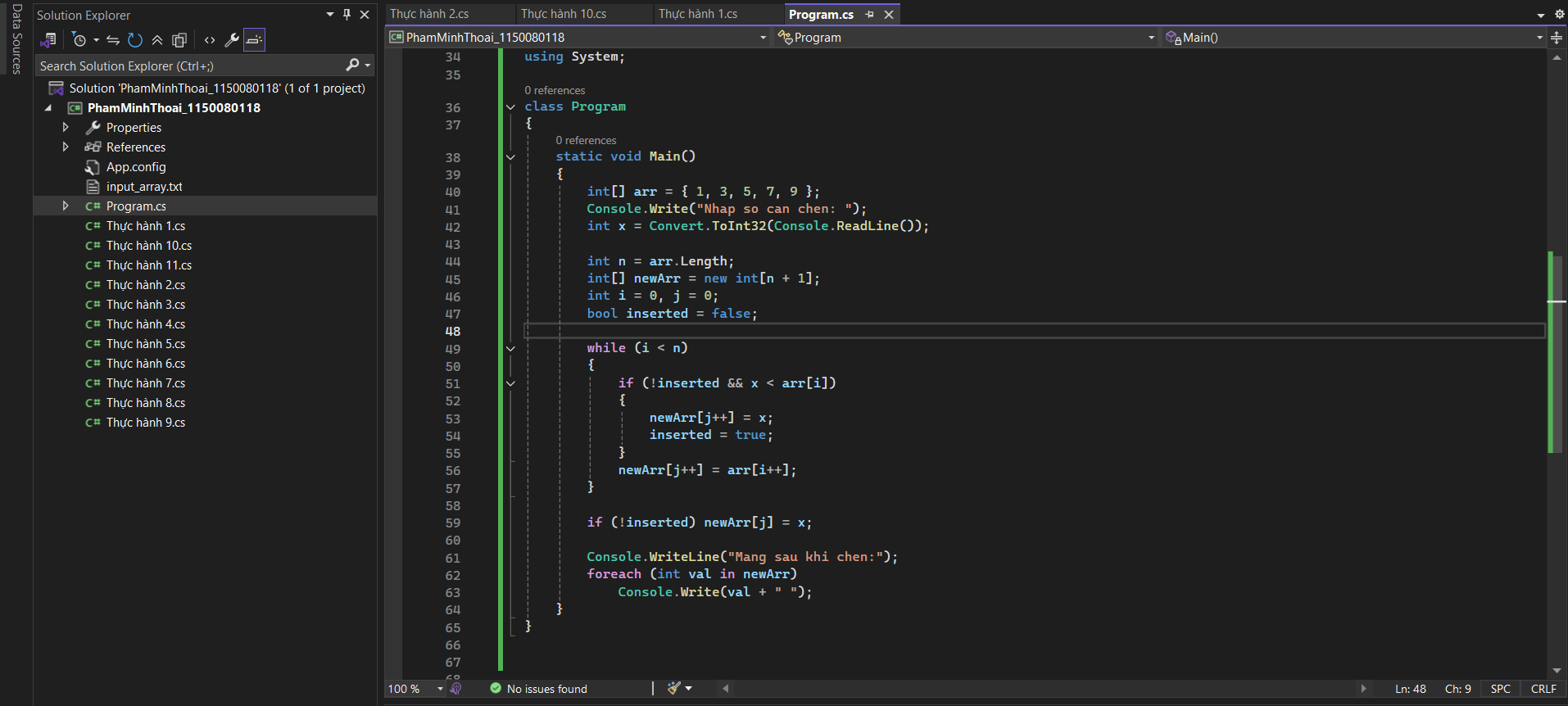
****

**Thực hành 10:** Sắp xếp mảng bằng Selection Sort

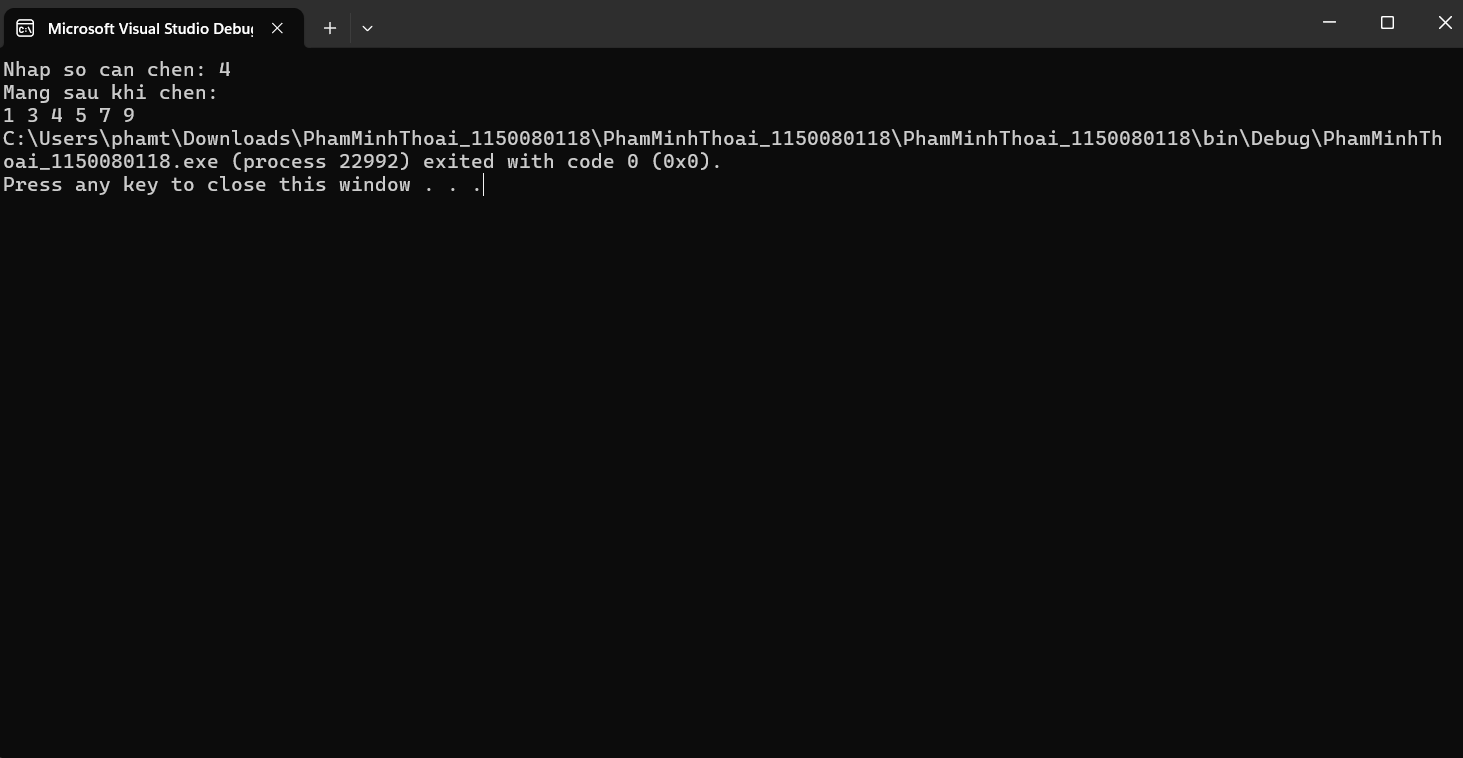


**Kết quả:**

****

**Thực hành 11:** Chèn số vào mảng đã sắp xếp

**Kết quả:**

****