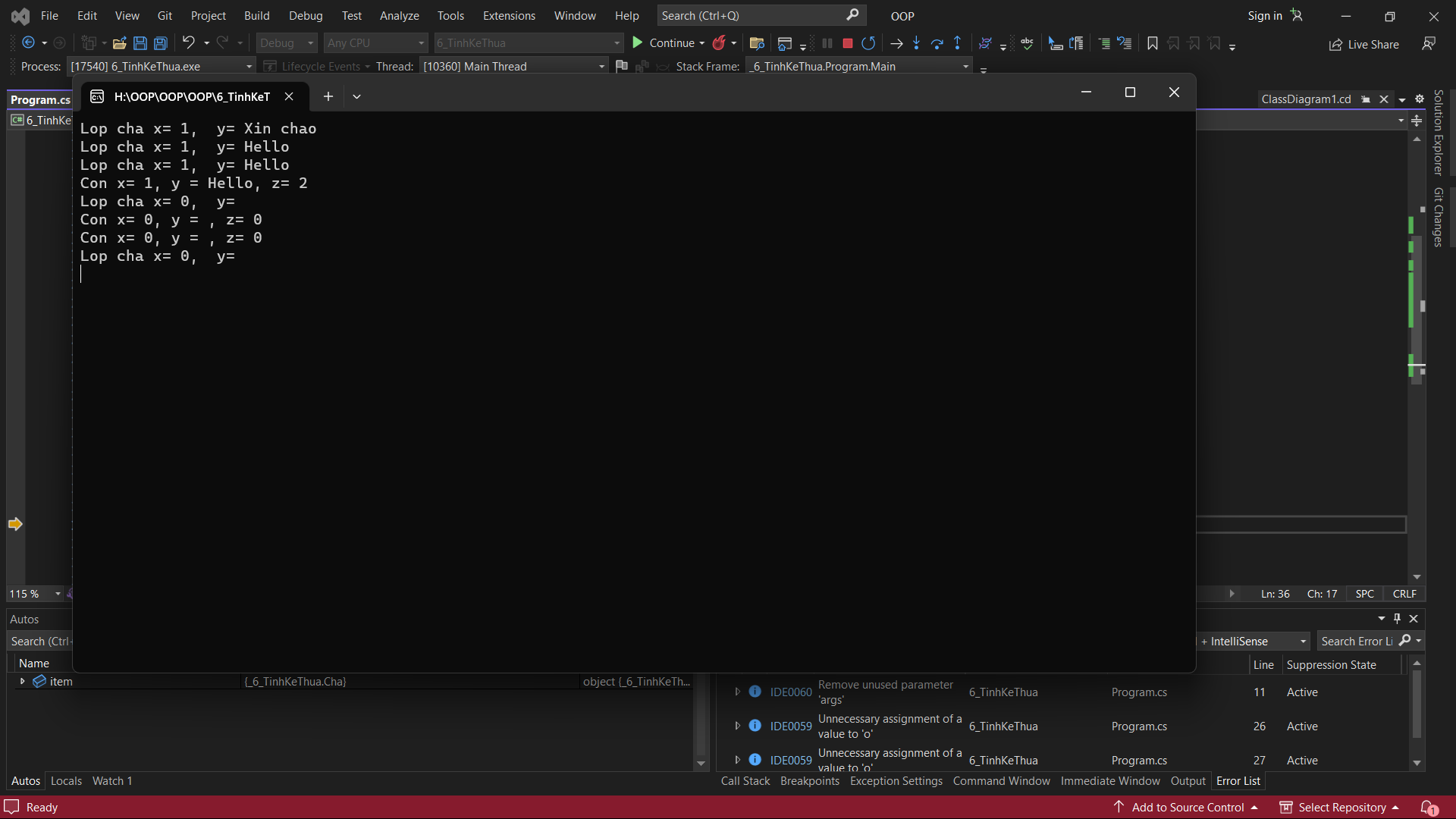
Lab 7: Tính kế thừa

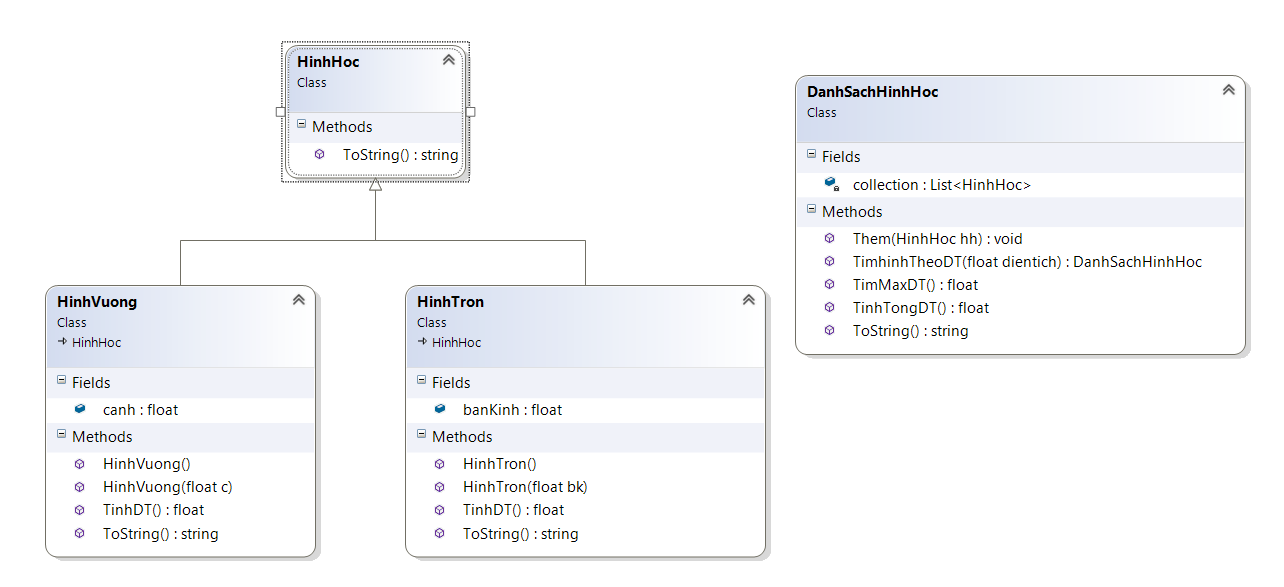
Bài 1: Sử dụng dự án 6\_TinhKeThua trong Code mẫu

* Đóng mở các chú thích trong dự án để chạy và kiểm tra kết quả
* Giải thích các dòng lệnh trong dự án và kết quả thực hiện chương trình:

+



Bài 2: Sử dụng dự án 7\_DuAnHinhHoc và thực hiện các yêu cầu sau:



class DanhSachHinhHoc

{

List<HinhHoc> collection = new List<HinhHoc>();

public void Them(HinhHoc hh)

{

collection.Add(hh);

}

public float TinhTongDT()

{

float tong = 0;

foreach (var item in collection)

{

if (item is HinhTron)

tong += ((HinhTron)item).TinhDT();

if (item is HinhVuong)

tong += ((HinhVuong)item).TinhDT();

}

return tong;

}

public float TimMaxDT()

{

float max = -1;

foreach (var item in collection)

{

float dt = 0;

if (item is HinhTron)

dt = ((HinhTron)item).TinhDT();

if (item is HinhVuong)

dt = ((HinhVuong)item).TinhDT();

if (max < dt)

max = dt;

}

return max;

}

public DanhSachHinhHoc TimhinhTheoDT(float dientich)

{

DanhSachHinhHoc kq = new DanhSachHinhHoc();

foreach (var item in collection)

{

float dt = 0;

if (item is HinhTron)

dt = ((HinhTron)item).TinhDT();

if (item is HinhVuong)

dt = ((HinhVuong)item).TinhDT();

if (dt == dientich)

kq.Them(item);

}

return kq;

}

public override string ToString()

{

string s = "";

foreach (var item in collection)

{

s += "\n" + item;

}

return s;

}

}

Trong lớp DanhSachHinhHoc thực hiện các yêu cầu sau:

1. Bổ sung lớp hình chữ nhật

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace \_7\_DuAnHinhHoc

{

internal class HinhChuNhat : HinhHoc

{

public float ChieuDai

{

get; set;

}

public float ChieuRong

{

get; set;

}

public HinhChuNhat(float chieuDai, float chieuRong) : base(chieuDai)

{

this.ChieuDai = chieuDai;

this.ChieuRong = chieuRong;

}

public override float TinhChuVi()

{

return (float)(2 \* (ChieuDai + ChieuRong));

}

public override float TinhDienTich()

{

return (float)(ChieuDai \* ChieuRong);

}

public void Xuat()

{

Console.WriteLine($"Hinh chu nhat co chieu dai {ChieuDai} va chieu rong {ChieuRong} co chu vi la {TinhChuVi()} va dien tich la {TinhDienTich()}");

}

}

}

1. Nhập từ File

public void NhapTuFile(string filename)

{

StreamReader sr = new StreamReader(filename);

string line;

while ((line = sr.ReadLine()) != null)

{

var data = line.Split(',');

if (data.Length > 0)

{

float a, b, c;

switch (data[0])

{

case "HT":

a = float.Parse(data[1]);

Them(new HinhTron(a));

break;

case "HV":

b = float.Parse(data[1]);

Them(new HinhVuong(b));

break;

case "HCN":

a = float.Parse(data[1]);

b = float.Parse(data[2]);

Them(new HinhChuNhat(a, b));

break;

default:

break;

}

}

}

}

case Menu.NhapTuFile:

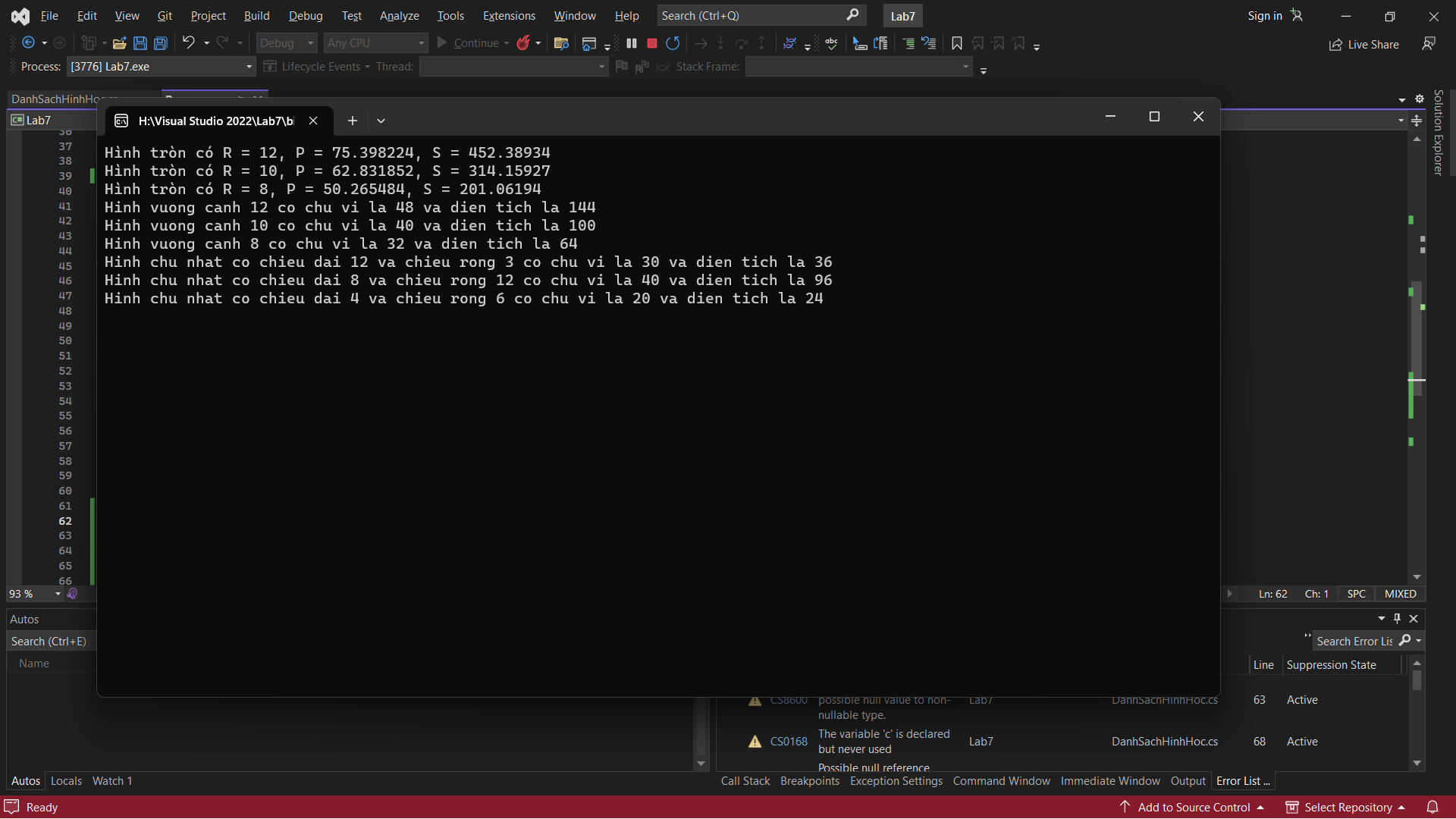
Console.Clear();

qlhh.NhapTuFile("data.txt");

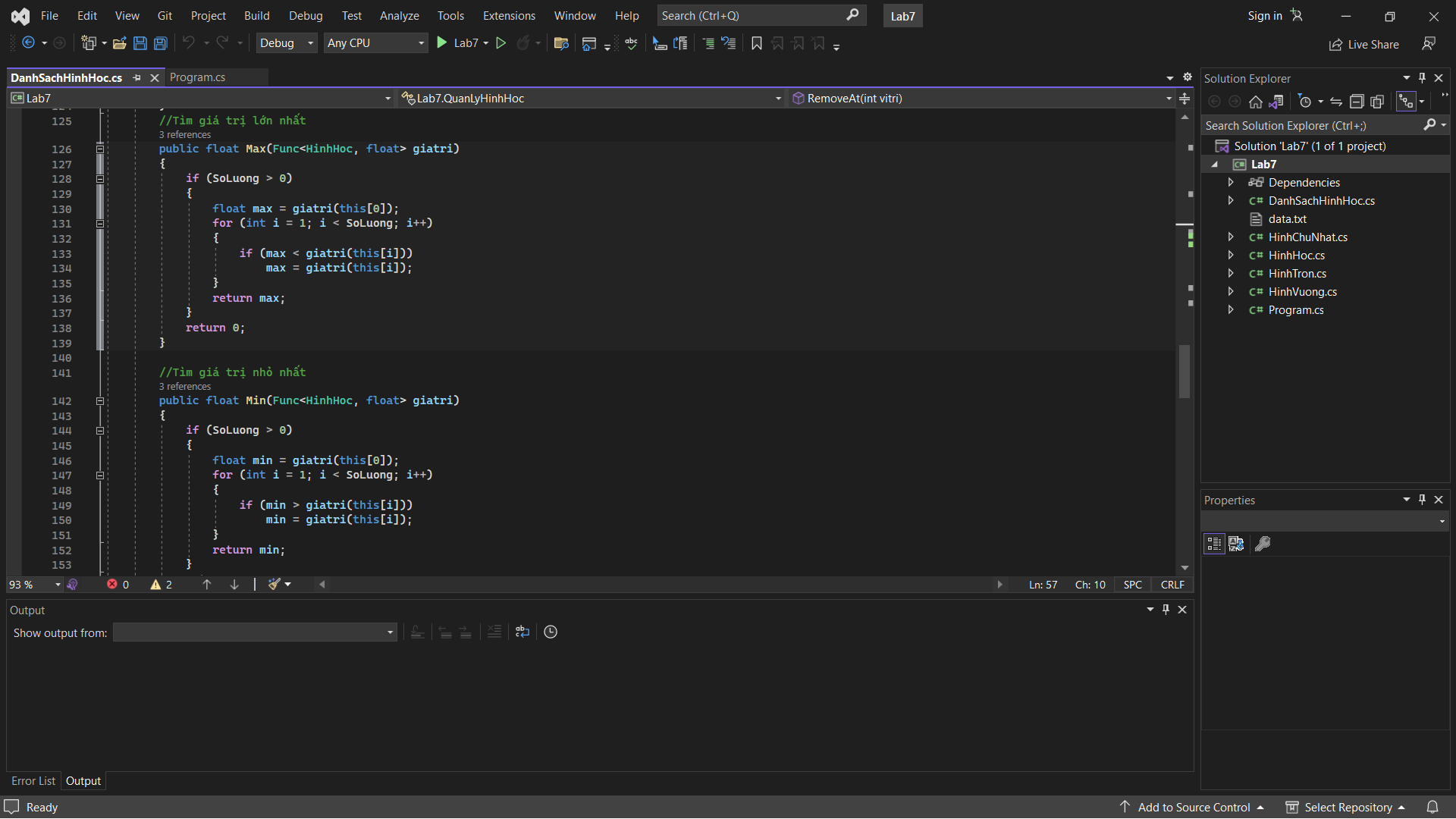
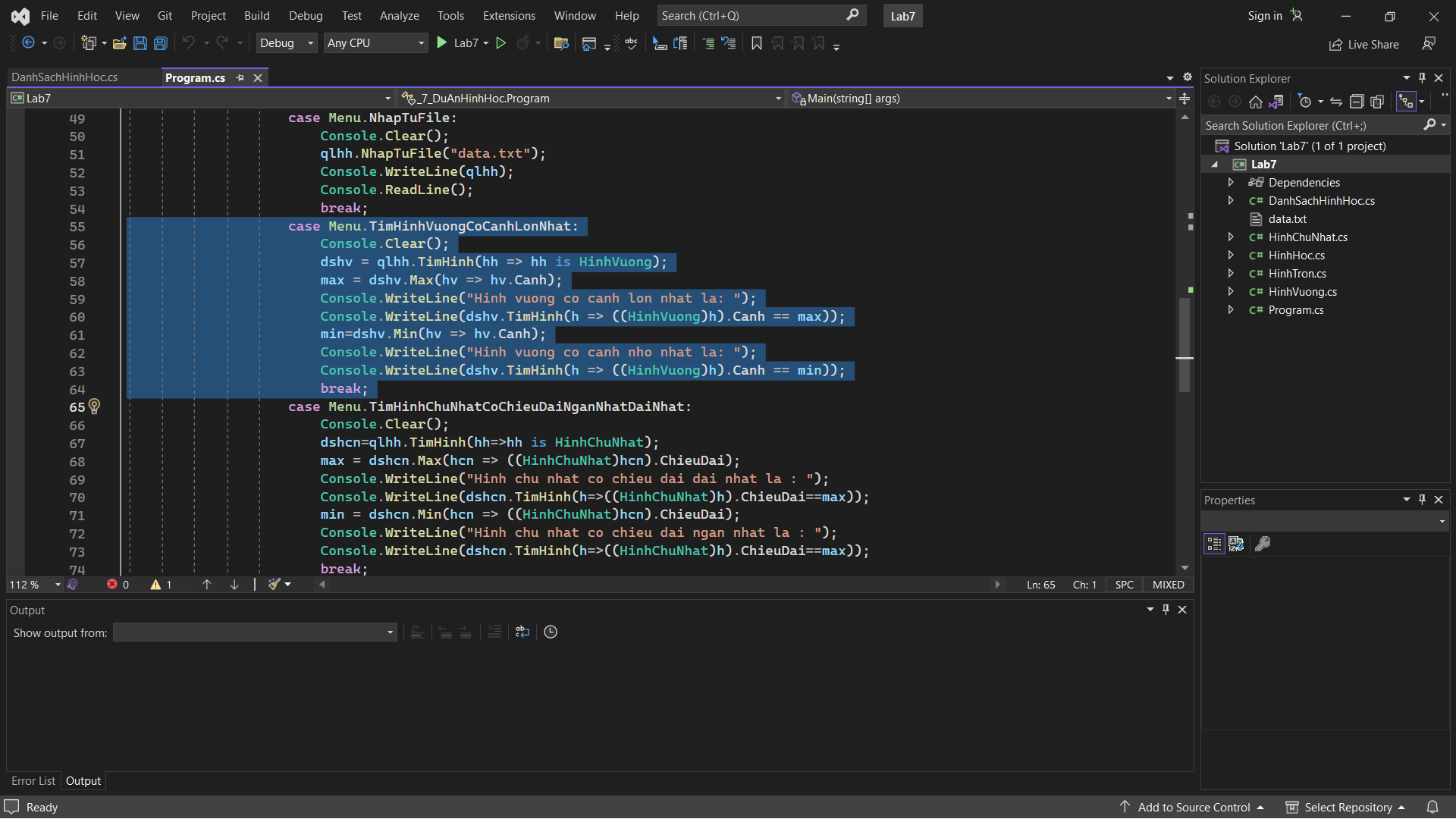
Console.WriteLine(qlhh);

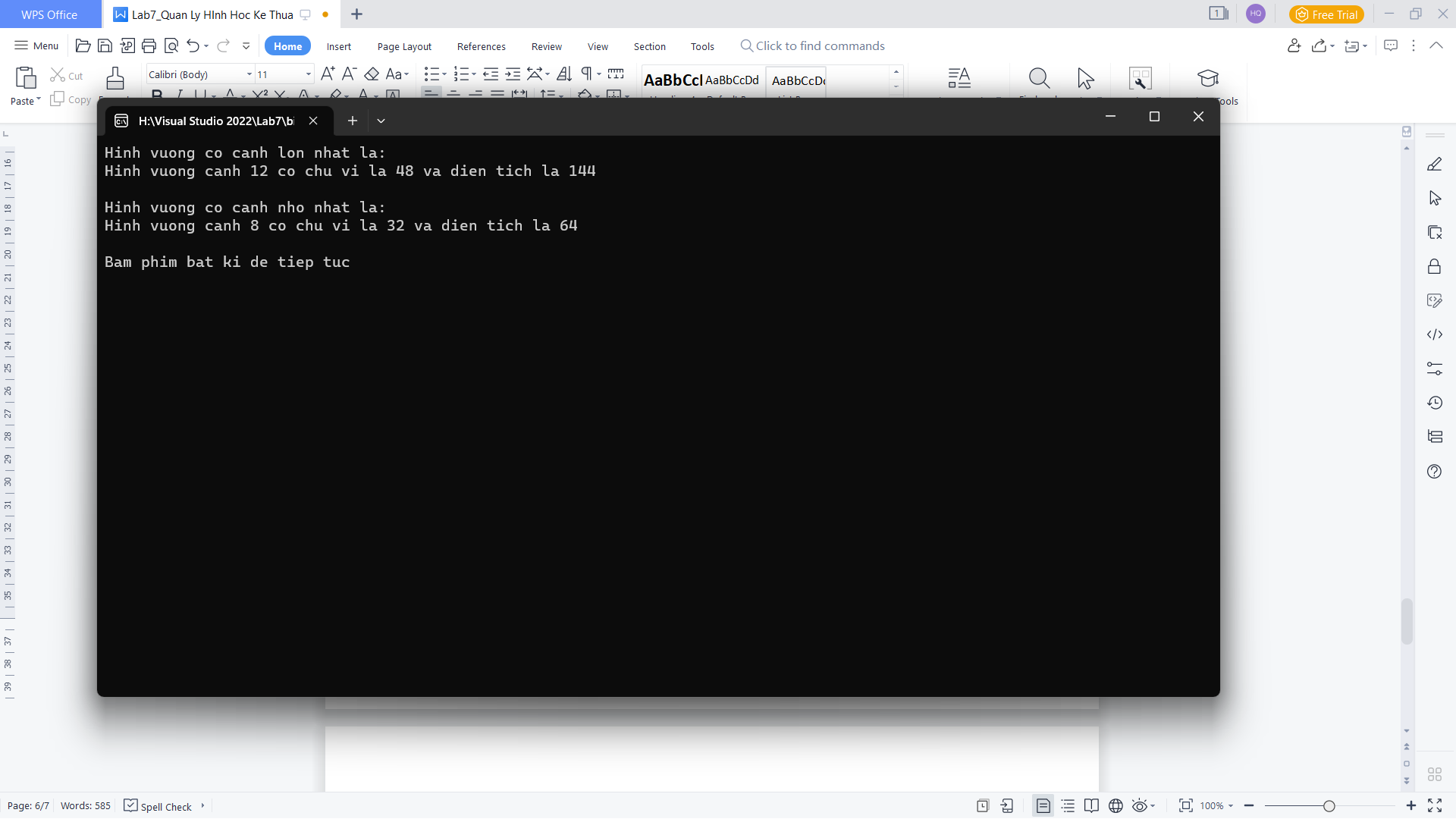
Console.ReadLine();

break;

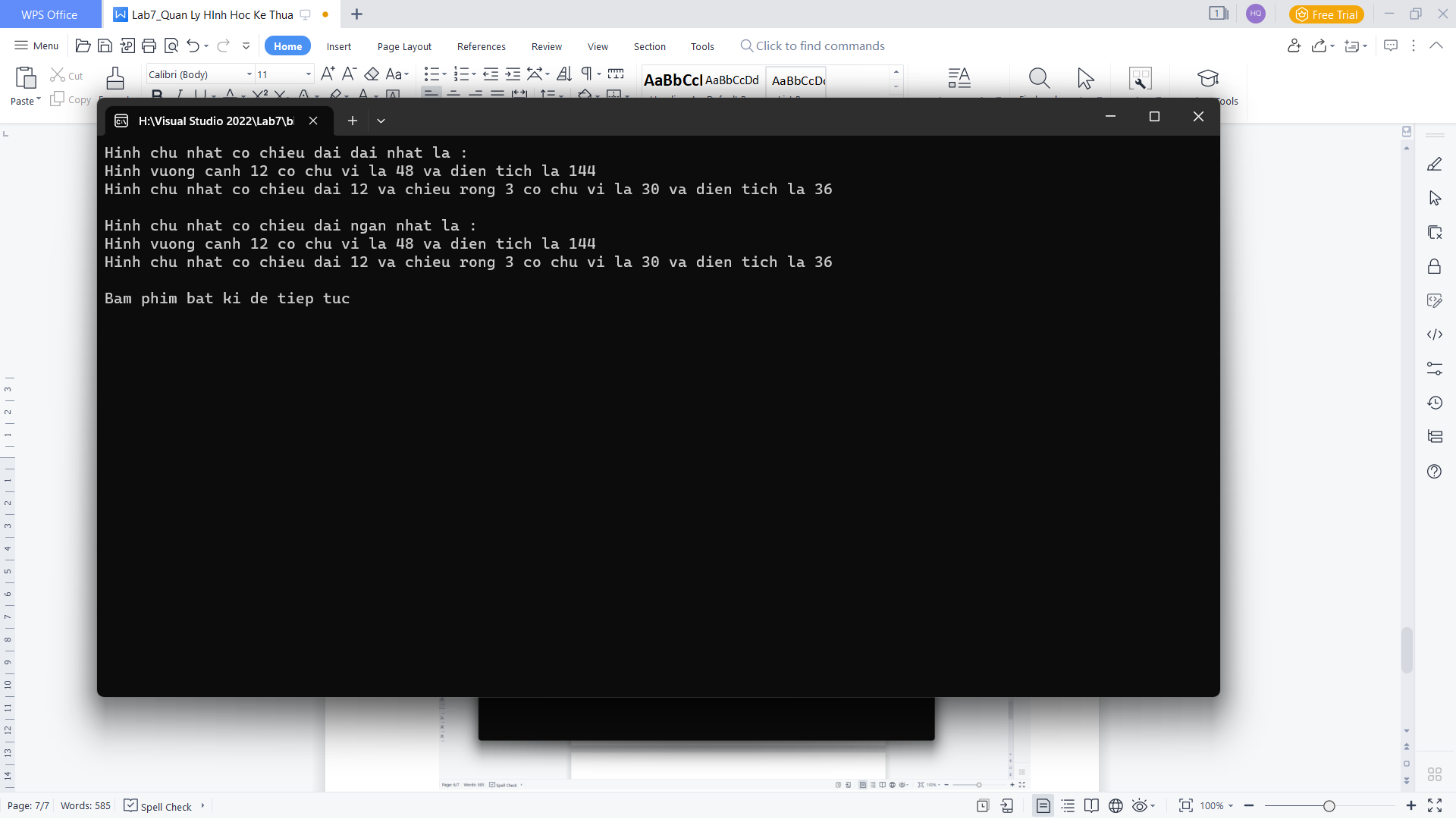
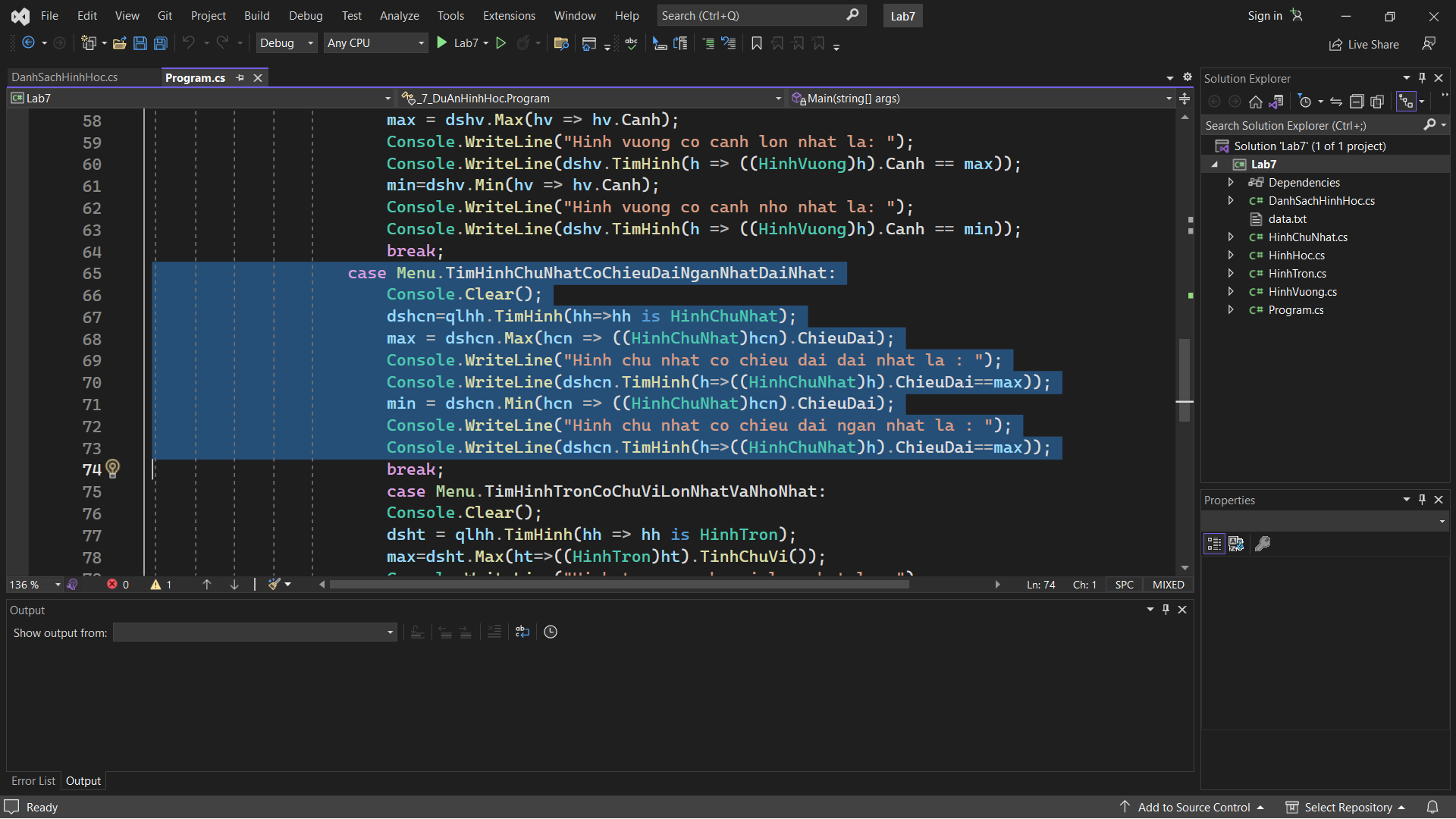


1. Tìm hình vuông có cạnh lớn nhất, nhỏ nhất

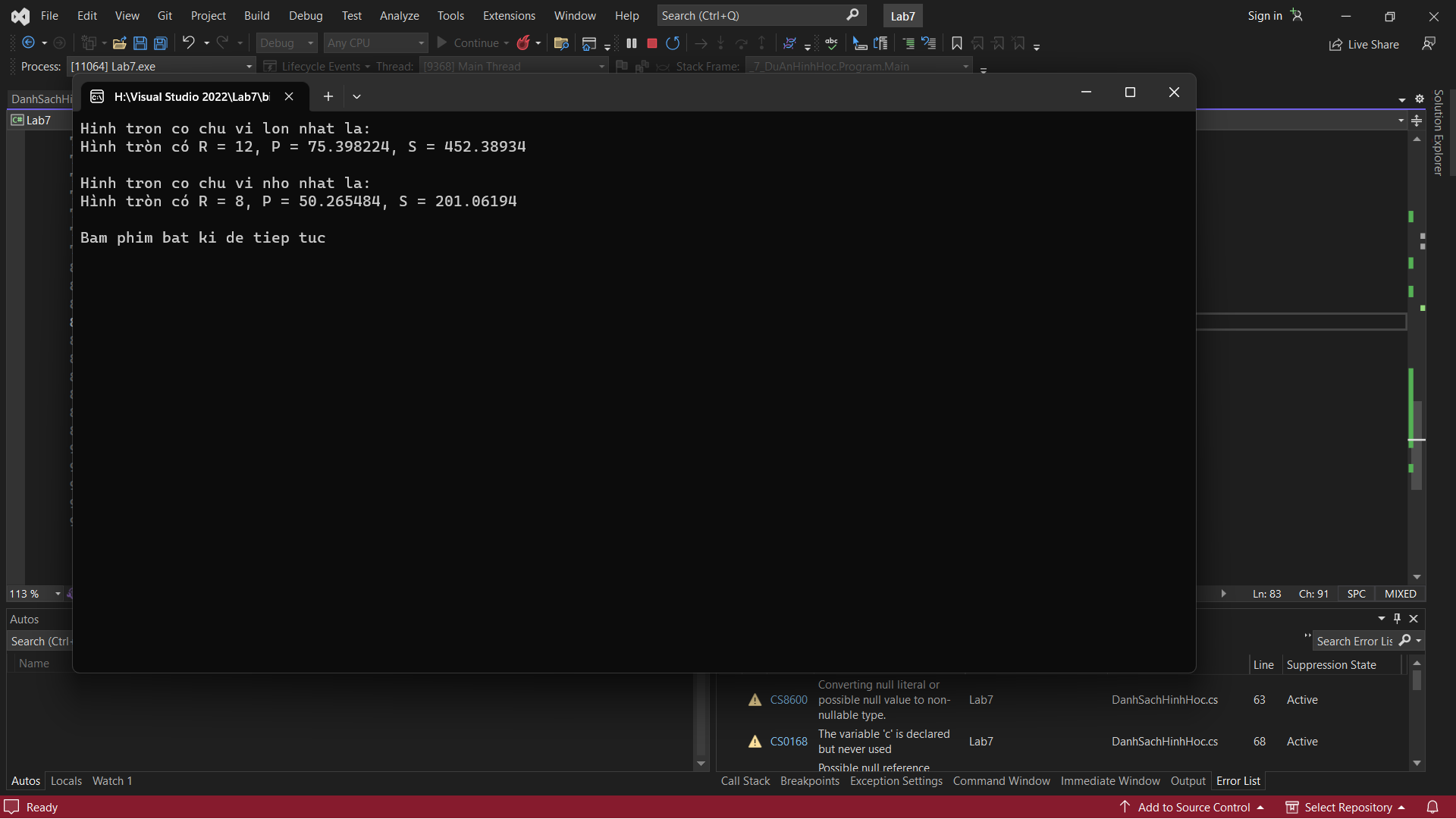
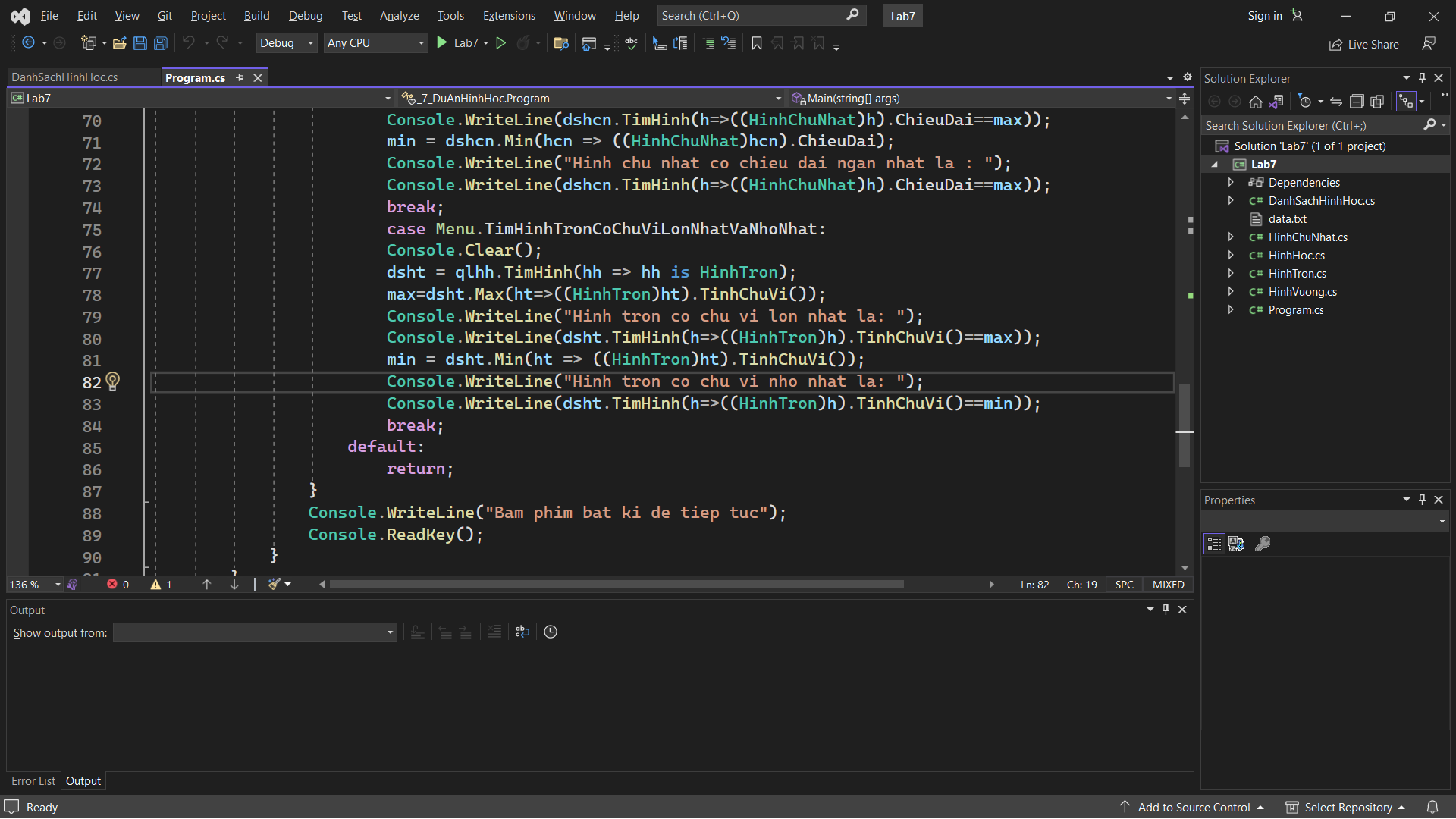




1. Tìm hình chữ nhật có chiều dài ngắn nhất, dài nhất



1. Sắp xếp các hình theo chiều tăng giảm của diện tích
2. Tính tổng chu vi của các hình
3. Tìm hình tròn có chu vi nhỏ nhất, lớn nhất



1. Ghi vào File theo hình (Ví dụ: file HinhTron.txt thì ghi hình tròn).
2. Tìm vị trí của hình có diện tích lớn nhất, nhỏ nhất
3. Tìm vị trí theo hình, theo diện tích
4. Xóa tất cả các hình theo diện tích
5. Xóa hình vuông có diện tích là x
6. Chèn hình tại vị trí x
7. Xóa hình tại vị trí x
8. Tìm loại hình có số lượng nhiều nhất, ít nhất
9. Sử dụng enum tạo thực đơn cho các yêu câu trên