**BÀI TẬP THỰC HÀNH MÔN KỸ THUẬT LẬP TRÌNH**

**------o0o------**

**Bài tập thực hành số 01: Cấu trúc – File**

1. Sử dụng NOTEPAD tạo 1 tập tin văn bản có nội dung như sau, đặt tên là SV.TXT, lưu trên thư mục gốc ổ đĩa C:

**14110201**

**Nguyen Van Muoi Hai**

**19**

**3**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**INPR123**

**Nhap mon Lap trinh**

**4**

**6.5**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**PRTE345**

**Ky thuat Lap trinh**

**4**

**8.0**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

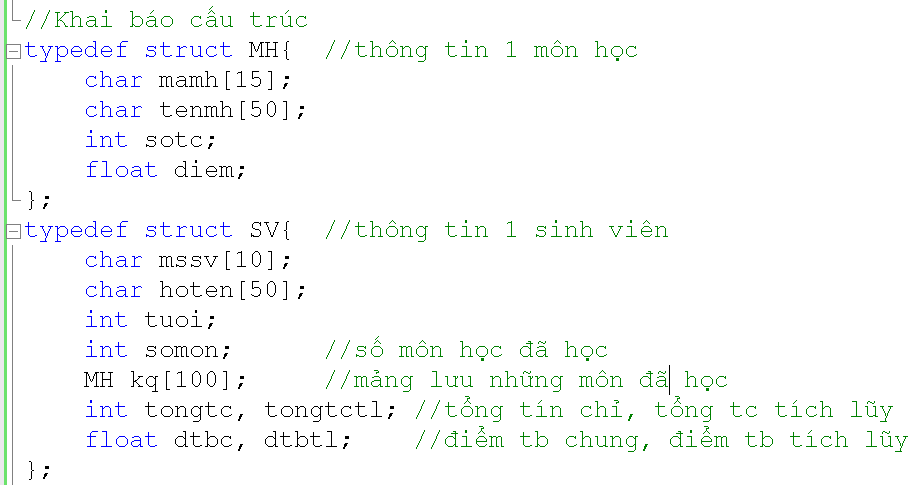
**MATH345**

**Toan cao cap A1**

**3**

**4.0**

1. Tạo một Project mới dạng C++ Console Application trên Visual Studio, khai báo các cấu trúc sau:



1. Khai báo các hàm xử lý cho chương trình, bao gồm hàm đọc dữ liệu sinh viên từ file, tính toán các thông số còn lại, hàm in thông tin một sinh viên ra màn hình và hàm ghi thông tin của 1 sinh viên lên tập tin văn bản.

//Khai báo các hàm thao tác

//1. Đọc dữ liệu từ file --> cấu trúc SV

//Đồng thời tính các thông số còn lại (tongtc,...)

void **docfile(**char **filename[], SV &s);**

//2. In thông tin của 1 sinh viên ra màn hình

void **insv(SV s);**

//3. Ghi dữ liệu từ cấu trúc SV --> file (văn bản)

void **ghifile(**char **filename[], SV s);**

int **main()**

{

**SV a;**

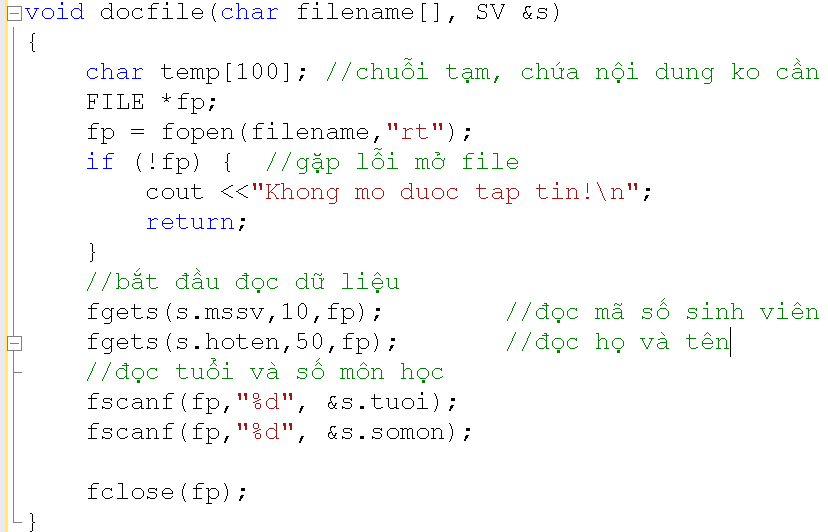
**docfile(**"C:/SV.TXT", a);

**insv2(a);**

return **0;**

}

1. Bắt đầu xây dựng thân các hàm trên, đoạn đầu của hàm docfile được viết như sau:



1. Hàm insv() cơ bản được viết như sau:

//2. In thông tin của 1 sinh viên ra màn hình

void **insv(SV s){**

**printf(**"Ma so SV : %s\n",s.mssv);

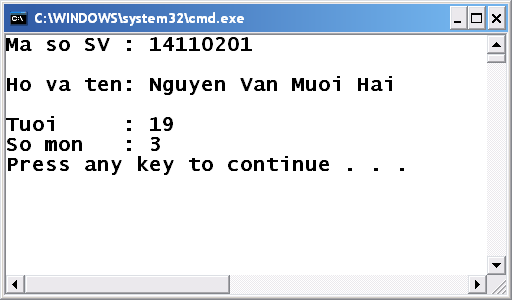
**printf(**"Ho va ten: %s\n",s.hoten);

**printf(**"Tuoi : %d\n",s.tuoi);

**printf(**"So mon : %d\n",s.somon);

}

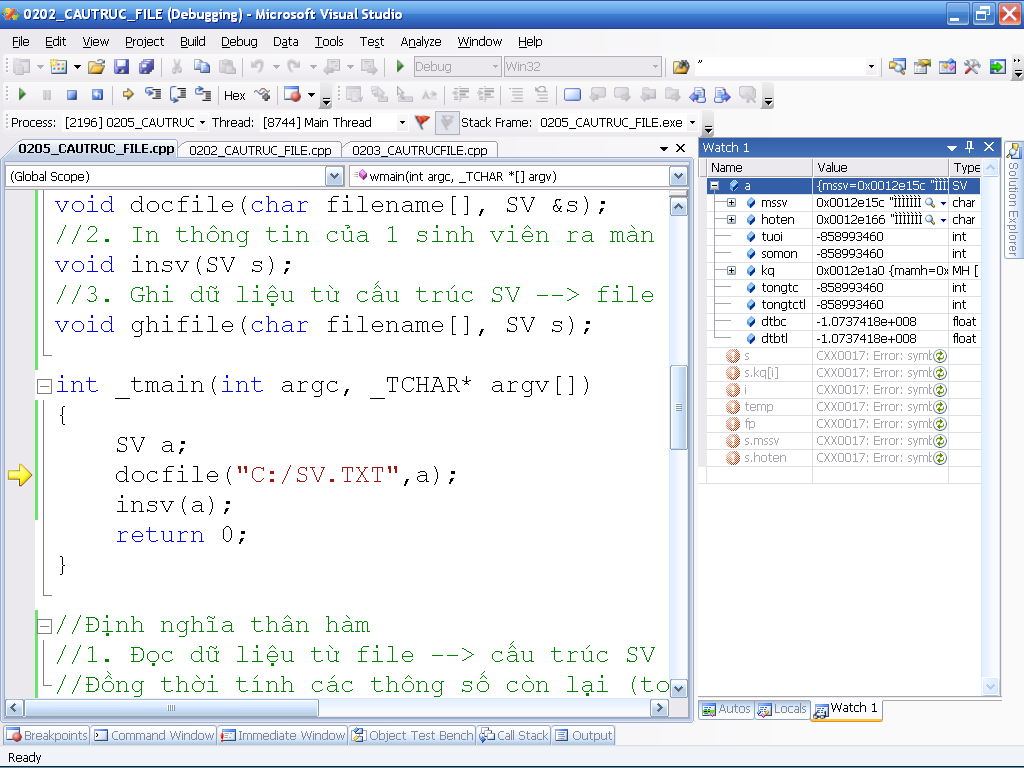
1. Hãy chạy thử chương trình, quan sát và nhận xét kết quả:



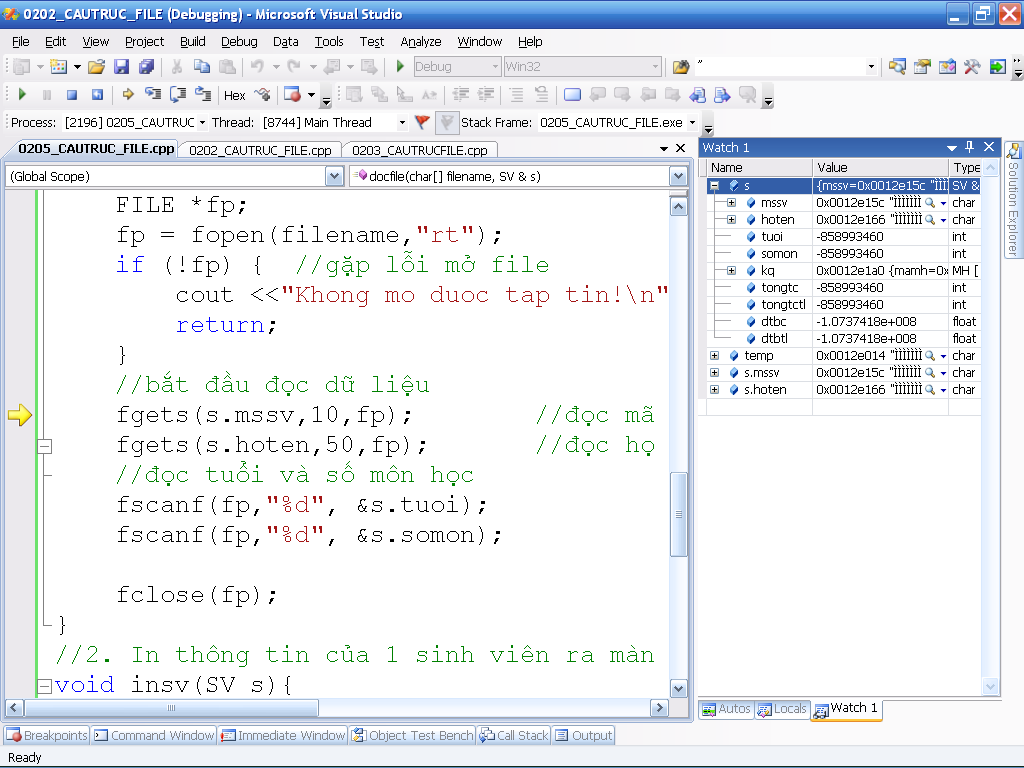
1. Chạy debug từng bước chương trình trên để phát hiện lỗi sai:

Hướng dẫn:

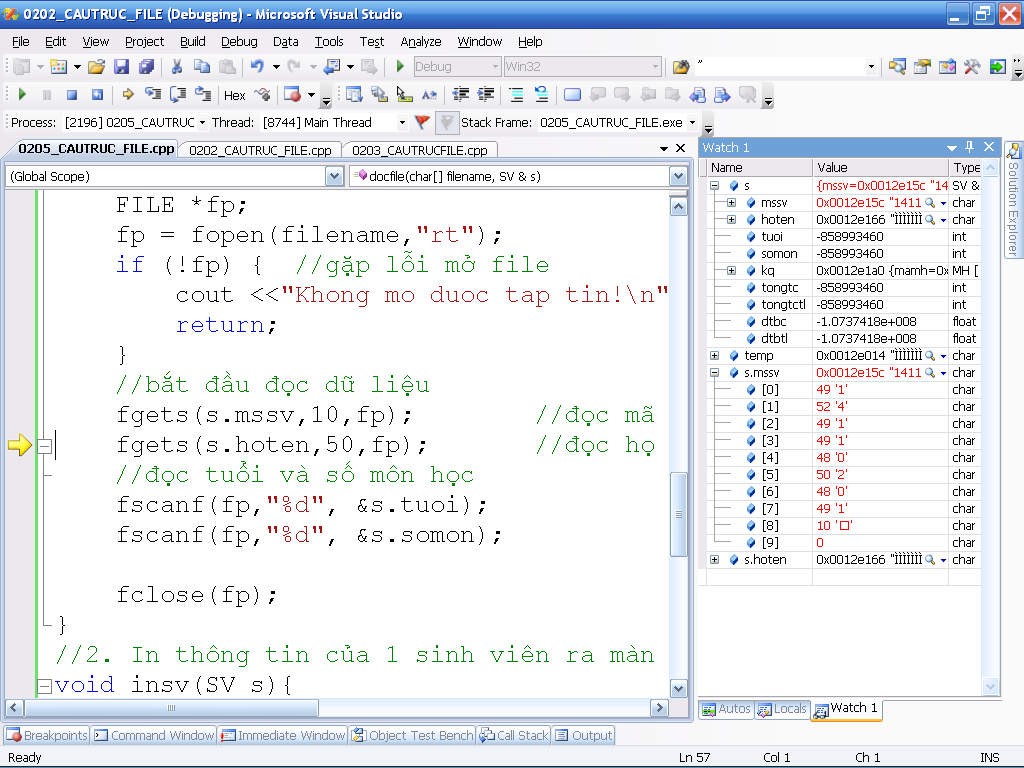
* Dùng phím F10 để chạy từng lệnh.
* Khi muốn chui vào bên trong một hàm (chương trình con), nhấn phím F11.

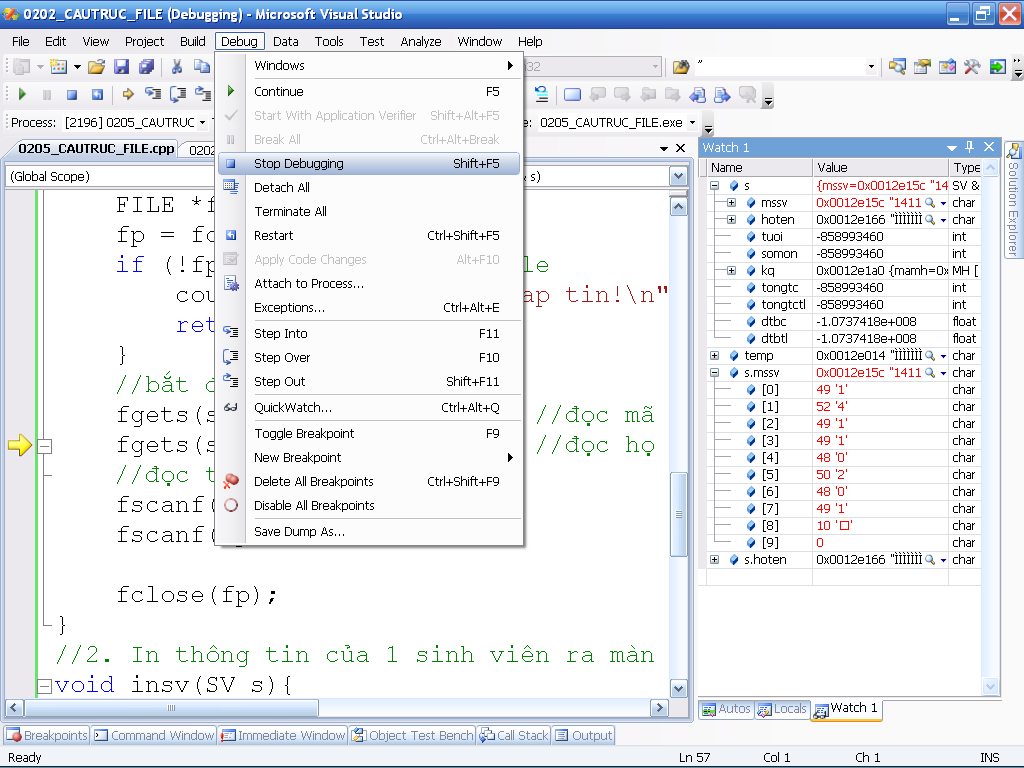


* Mở cửa sổ Watch để theo dõi nội dung các biến.

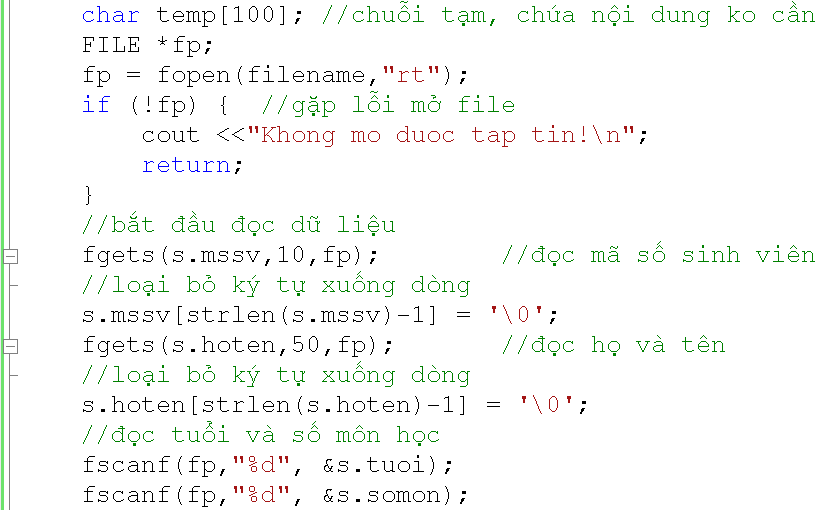


* Dừng debug để quay lại sửa lỗi chương trình khi phát hiện lỗi sai (trong hình ví dụ dưới đây là chuỗi s.mssv đọc được từ file bị dư ký tự xuống dòng (mã 10).

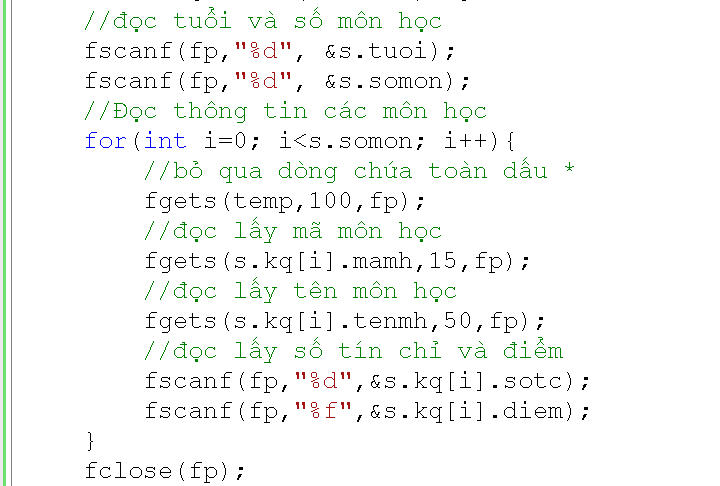




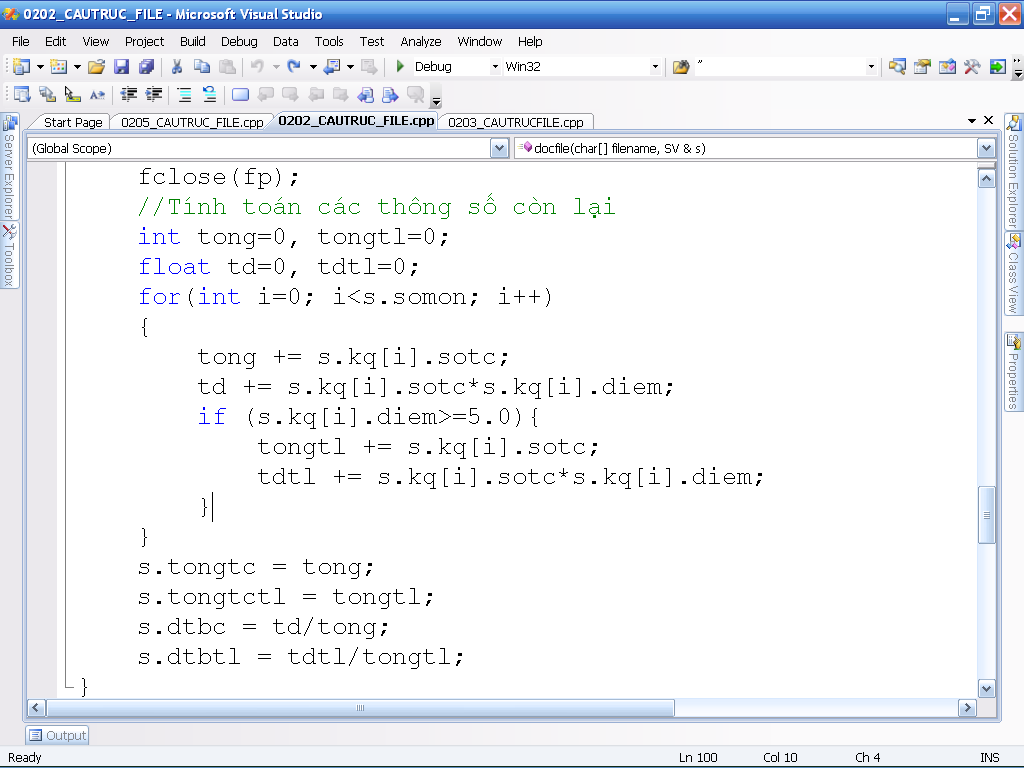
* Code hàm docfile() sau khi điều chỉnh:



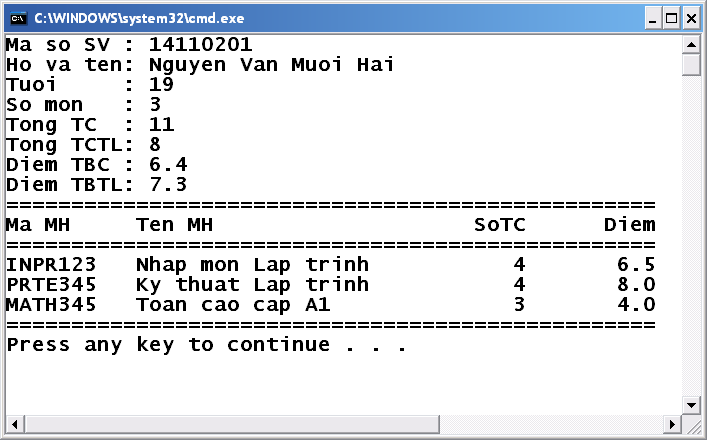
1. Tiếp tục thêm phần code đọc thông tin các môn học:



1. Debug lại và điều chỉnh chương trình cho hoàn chỉnh.
2. Thêm phần code để tính toán các thông số còn lại cho hàm docfile():



1. Hoàn chỉnh hàm insv() để in ra màn hình thông tin của sinh viên với nội dung như sau:



1. Viết thân hàm ghifile() để ghi toàn bộ nội dung như hình trên (ở câu 11) lên tập tin văn bản có tên OUTPUT1.TXT.
2. Viết thêm hàm ghifile2() để ghi lại nội dung thông tin sinh viên vào tập tin văn bản tên OUTPUT2.TXT, cấu trúc giống với tập tin SV.TXT đã dùng để nhập liệu lúc đầu.
3. Mở rộng chương trình trên để cho phép đọc thông tin của nhiều sinh viên từ tập tin có cấu trúc như sau:

**2 //Số lượng SV**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //TTin SV 1**

**14110201 //Mã số SV**

**Nguyen Van Muoi Hai //Họ và tên**

**19 //Tuổi**

**2 //Số môn học**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //TTin môn 1**

**INPR123 //Mã môn học**

**Nhap mon Lap trinh //Tên môn học**

**4 //Số tín chỉ**

**6.5 //Điểm**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**PRTE345**

**Ky thuat Lap trinh**

**4**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //TTin SV 2**

**14110205 //Mã số SV**

**Nguyen Van Ti //Họ và tên**

**19 //Tuổi**

**1 //Số môn học**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* //TTin môn 1**

**INPR123 //Mã môn học**

**Nhap mon Lap trinh //Tên môn học**

**4 //Số tín chỉ**

**8.5 //Điểm**

1. Cài đặt các chức năng sau cho chương trình quản lý danh sách sinh viên:

* In danh sách tổng hợp sinh viên, sắp xếp tăng dần theo Tên, hoặc giảm dần theo điểm trung bình chung.
* Tìm và in ra thông tin của sinh viên giỏi nhất (có điểm trung bình chung cao nhất).
* Liệt kê danh sách những sinh viên có học bổng (ĐTBC từ 7.0 trở lên và không nợ môn nào).
* Liệt kê danh sách những sinh viên bị cảnh cáo học vụ và thử thách học vụ (ĐTBC dưới 4.0).
* Liệt kê danh sách những sinh viên đã học và còn nợ một môn học nào đó (mã môn học nhập từ bàn phím).

**Bài tập thực hành số 02: BIGNUM, MẢNG 2 CHIỀU**

1. Cho chương trình khai báo, nhập, xuất số lớn. Copy vào VS và điều chỉnh lại để chương trình chạy được, cho phép nhập/xuất số lớn.

// 0313\_BIGNUM.cpp : Defines the entry point for the console application.

//

#include"stdafx.h"

#include<iostream>

#include<conio.h>

usingnamespace **std;**

//Khai báo hằng

#define **MAXDIGITS 100** /\* maximum length \*/

#define **PLUS 1** /\* positive sign bit \*/

#define **MINUS -1** /\* negative sign bit \*/

//Khai báo cấu trúc

typedefstruct **{**

char **digits[MAXDIGITS];** /\* the number \*/

int **signbit;** /\* PLUS or MINUS \*/

int **lastdigit;** /\*index of high-order digit\*/

}bignum;

//Khai báo hàm

void **scan\_bignum(bignum &n);**

void **print\_bignum(bignum n);**

int **kiemtra(**char **s[]);**

void **main()**

{

do{

**bignum a, b, c;**

**cout <<**"Nhap a:";

**scan\_bignum(a);**

**cout <<**"Nhap b:";

**scan\_bignum(b);**

**add\_bignum(a,b,c);**

**cout <<**"a+b=";

**print\_bignum(c);**

**cout <<endl;**

**subtract\_bignum(a,b,c);**

**cout <<**"a-b=";

**print\_bignum(c);**

**cout <<endl;**

**cout <<**"Nhan phim ESC de thoat...\n";

**}**while **(getch()!=27);**

}

//Một số hàm xử lý

void **print\_bignum(bignum n)**

{

int **i;**

if **(n.signbit == MINUS) cout <<**"-";

for **(i=n.lastdigit; i>=0; i--)**

**cout <<** int(n.digits[i]);

}

int **kiemtra(**char **s[]){**

//ktra ký tự đầu

if **(s[0]!=**'-' **&& !isdigit(s[0]))**

return **0;**

//ktra các ký tự tiếp theo

int **len = strlen(s);**

for(int **i=1; i<len; i++)**

if **(!isdigit(s[i]))** return **0;**

//không phát hiện vi phạm

return **1;**

}

void **scan\_bignum(bignum &n)**

{

//B1: Nhập chuỗi

char **temp[256];**

**cin.getline(temp,256);**

//B2: Kiểm tra

if **(!kiemtra(temp)){** //không hợp lệ

//gán bignum = 0

**n.signbit = PLUS;**

**n.lastdigit = 0;**

**n.digits[0] = 0;**

**}**

else **{** //hợp lệ. B3: chuyển thành số bignum

//xét 2 trường hợp

if **(temp[0]!=**'-'){ //số dương

**n.signbit = PLUS;**

**n.lastdigit = strlen(temp)-1;**

for(int **i=0; i<=n.lastdigit; i++)**

**n.digits[i] = temp[n.lastdigit-i]-48;**

**}**

else **{** //số âm

**n.signbit = MINUS;**

**n.lastdigit = strlen(temp)-2;**

for(int **i=0; i<=n.lastdigit; i++)**

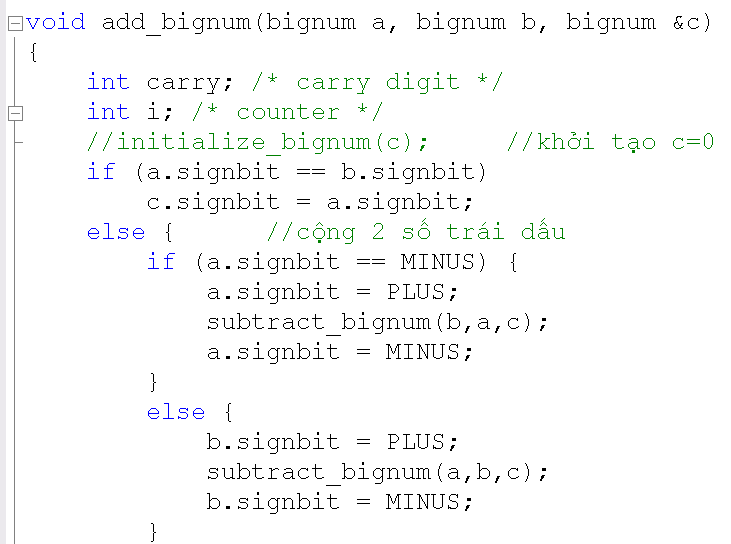
**n.digits[i] = temp[n.lastdigit+1-i]-48;**

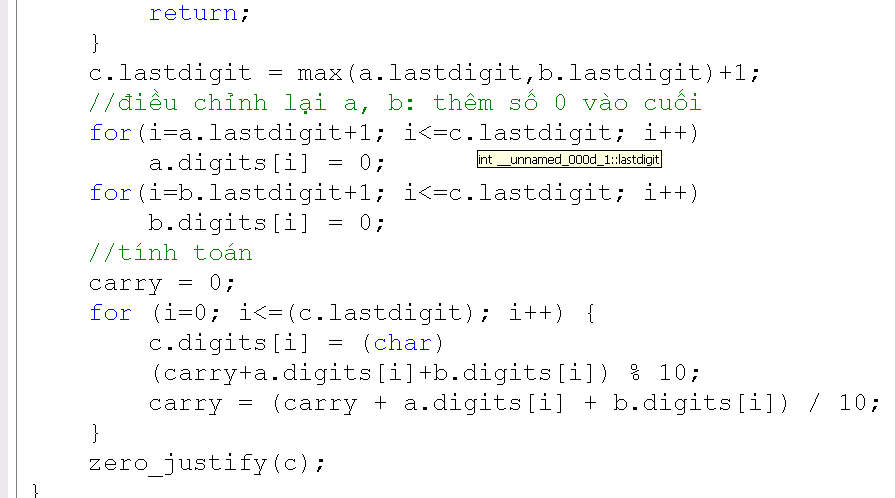
**}**

**}**

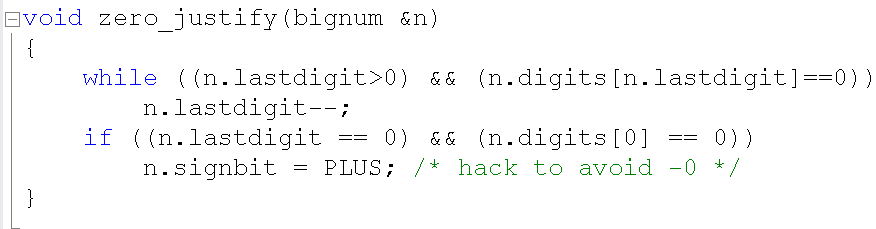
}

1. Copy hàm add\_bignum() từ file Programming Challenges.PDF và điều chỉnh lại (dùng CTRL-H để thay thế phép toán -> bằng phép toán .).

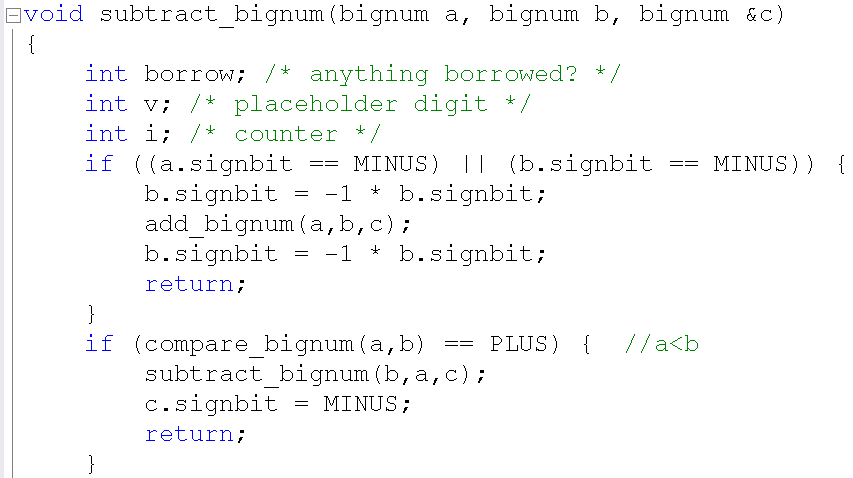


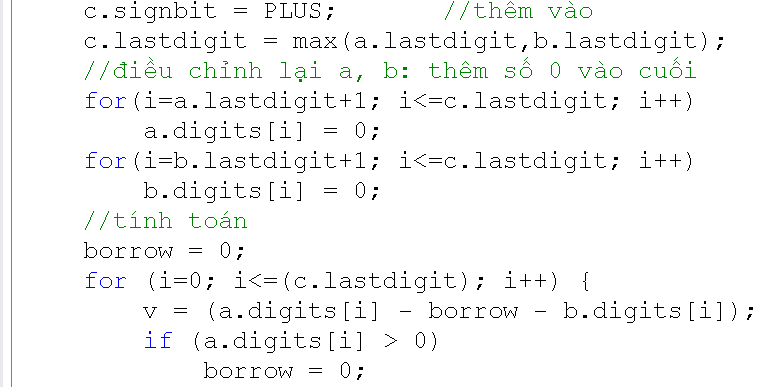


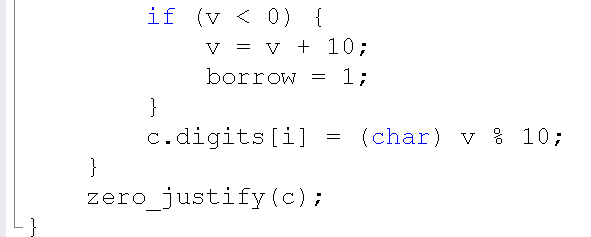
1. Copy hàm zero\_justify() và điều chỉnh lại.



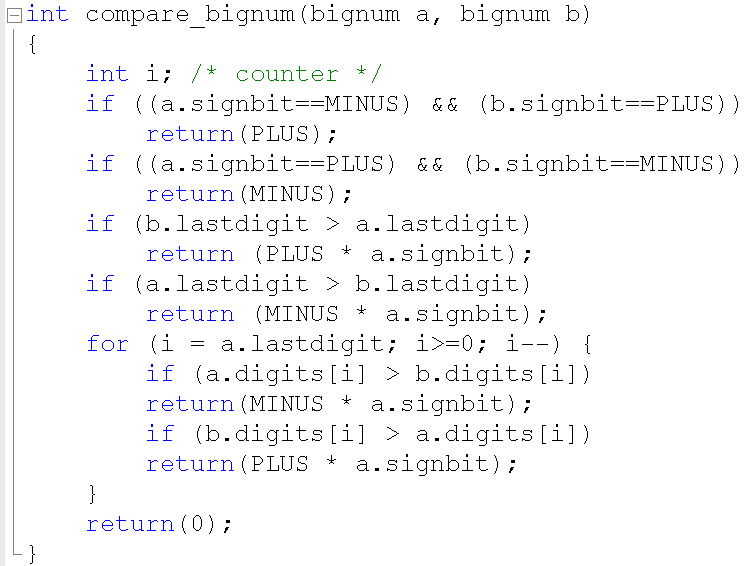
1. Copy hàm subtract\_bignum() và điều chỉnh lại.



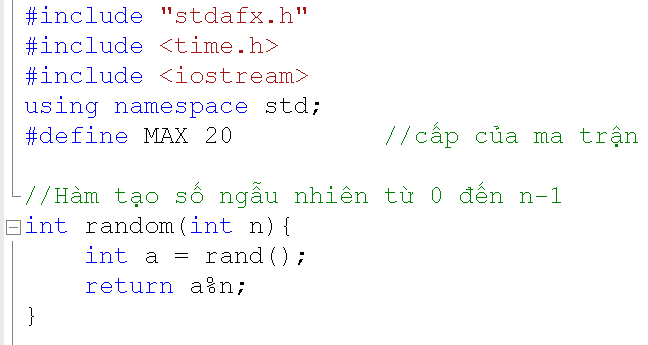




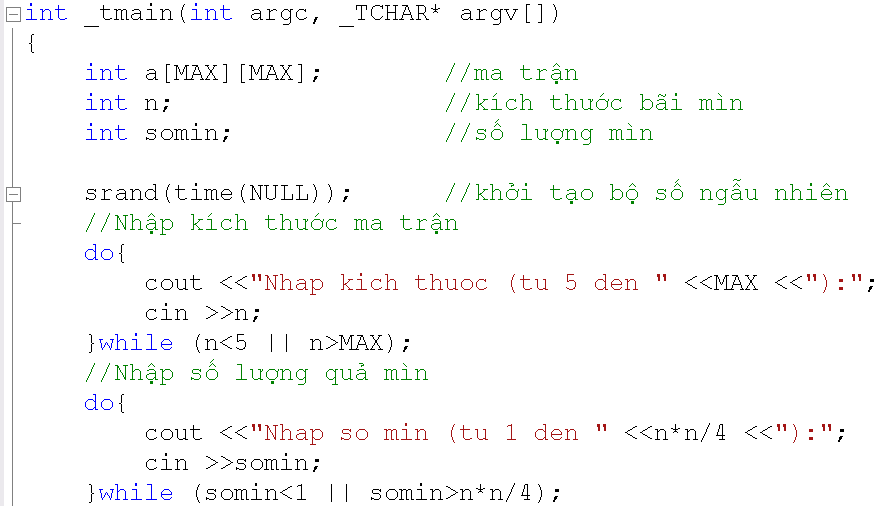
1. Copy hàm compare\_bignum() và điều chỉnh lại.



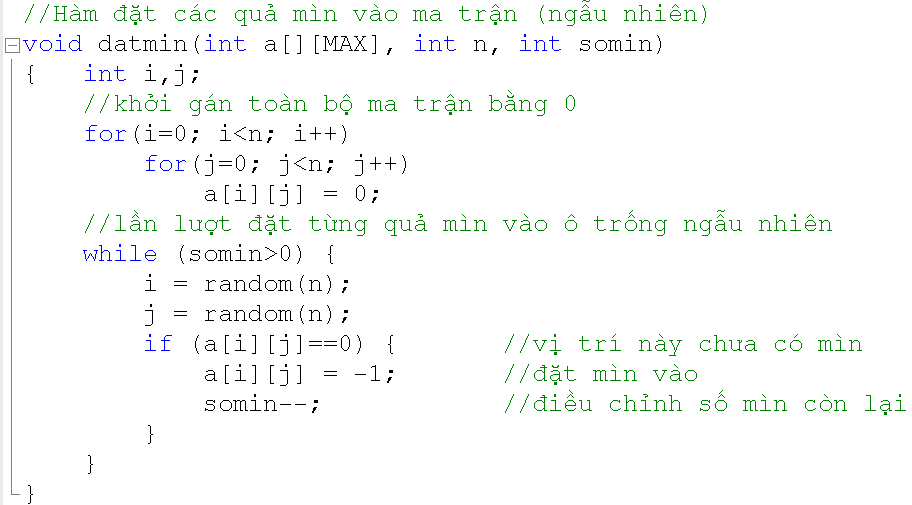
1. Chạy thử và kiểm lỗi, sửa lỗi chương trình.
2. Tiếp tục hoàn thành các hàm nhân, chia số lớn, đổi số nguyên thành số lớn.
3. Cài đặt chương trình dò mìn – MINESWEEPER (yêu cầu chi tiết xem trong SHEET “Domin” của file MATRAN.XLSX
4. Hàm phát sinh số ngẫu nhiên từ 0 đến n-1



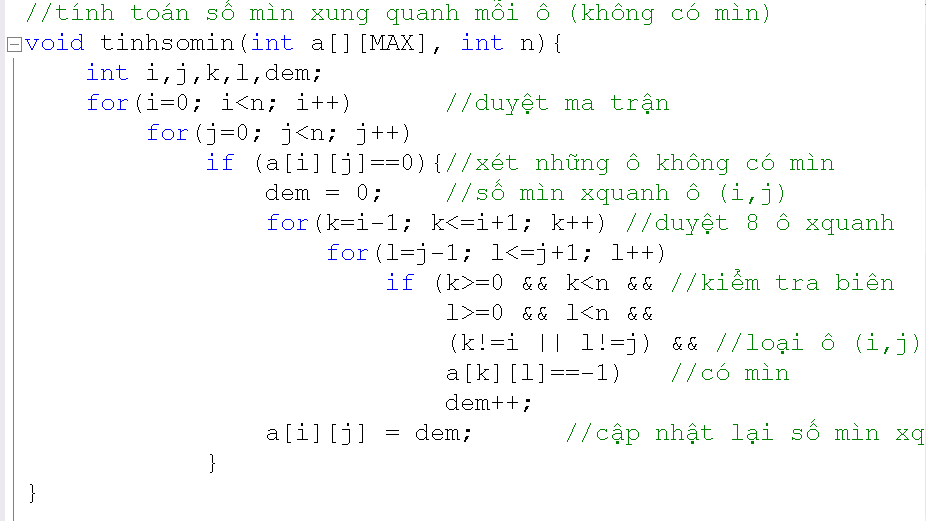
1. Chương trình nhập kích thước của bãi mìn và số lượng mìn, có kiểm tra điều kiện ràng buộc.



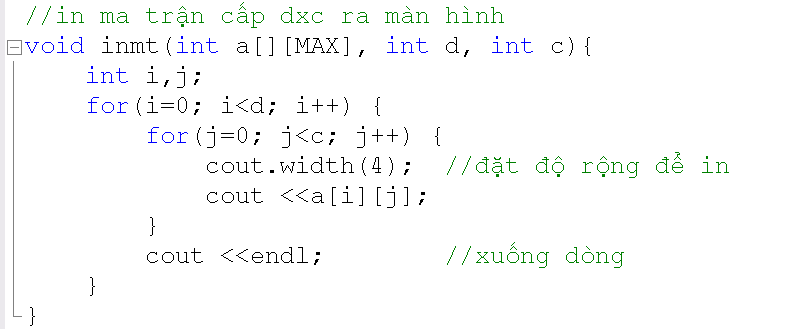
1. Hàm đặt những quả mìn vào các vị trí ngẫu nhiên trong “bãi mìn”.



1. Hàm tính toán số mìn xung quanh mỗi ô.



1. Hàm in ma trận ra màn hình.



1. Viết các hàm thực hiện những yêu cầu sau:

//6.Viết hàm tìm vị trí của ô xung quanh có nhiều mìn nhất

//trả về số mìn nhiều nhất

//các tham số d và c lưu dòng,cột của ô tìm được

int **timmax(**int **a[][MAX],** int **n,** int **&d,** int **&c);**

//7\*.Viết hàm tìm vùng kxk có ít mìn nhất (k cho trước)

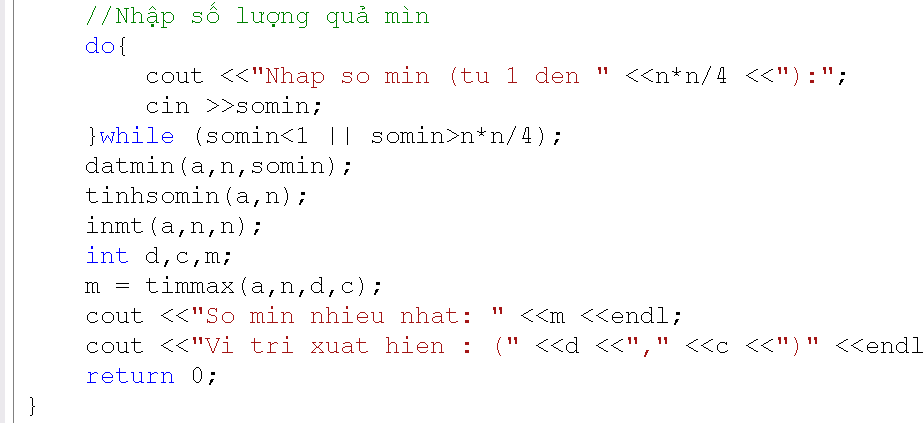
int **timmin(**int **a[][MAX],** int **n,** int **k,** int **&d,** int **&c);**

//8\*\*.Viết hàm cho biết vùng chứa những số 0 liền kề lớn

//nhất gồm bao nhiêu ô.

int **timvung0(**int **a[][MAX],** int **n);**

1. Đoạn hàm main() gọi các hàm.



1. Viết chương trình trò chơi caro (xem chi tiết trong SHEET “Caro” của file MATRAN.XLSX)

#include"stdafx.h"

#include<iostream>

usingnamespace **std;**

#define **MAX 30**

//Khai báo hàm

//đọc dữ liệu từ tập tin văn bản vào ma trận vuông

void **docfile(**char **\*fname,** int **a[][MAX],** int **&n);**

//in ma trận ra màn hình

void **inmt(**int **a[][MAX],** int **d,** int **c);**

//kiểm tra trạng thái hợp lệ của ván cờ

int **ktrahople(**int **a[][MAX],** int **n);**

//kiểm tra thắng thua (k=0 hoặc 1)

int **ktrathang(**int **a[][MAX],** int **n,** int **k);**

int **\_tmain(**int **argc, \_TCHAR\* argv[])**

{

int **a[MAX][MAX], n;**

**docfile(**"F:/CARO.TXT",a,n);

**inmt(a,n,n);**

return **0;**

}

1. Yêu cầu mở rộng: tìm vị trí có thể tạo thành đường 4 chưa bị chặn 2 đầu; tìm vị trí có thể tạo nên đường 3 chưa bị chặn cả 2 đầu đối với quân K (K=0 hoặc K=1).