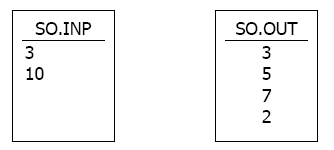
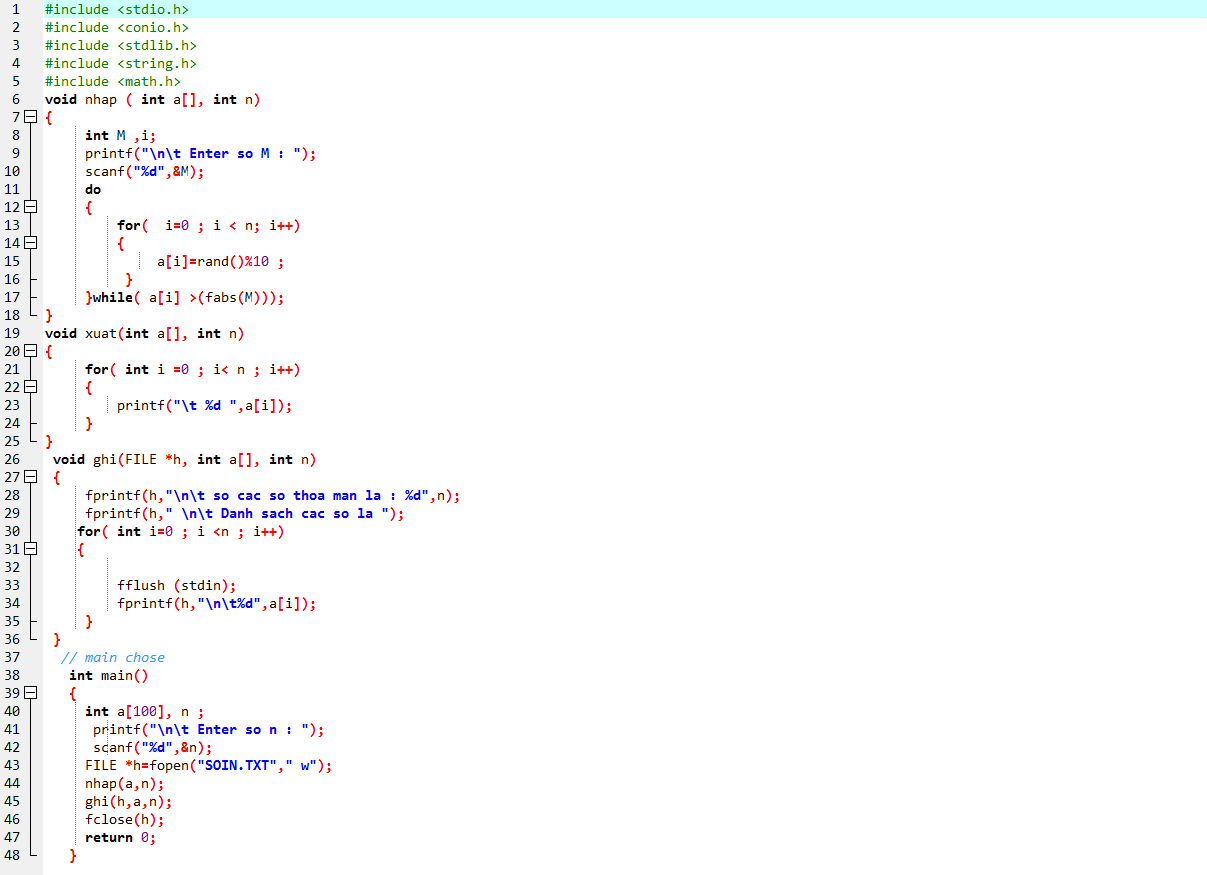
**BÀI TẬP FILE**

**BÀI 1:**Viết chương trình tạo tập tin văn bản SO.OUT gồm n số nguyên, các số của dãy được tạo ngẫu nhiên có giá trị tuyệt đối không vượt quá M ( n, M đọc từ tập tin SO.INP). Kết quả chương trình là 1 tập tin văn bản có dòng thứ nhất ghi số n; n dòng tiếp theo ghi các số tạo được, mỗi số trên một dòng.

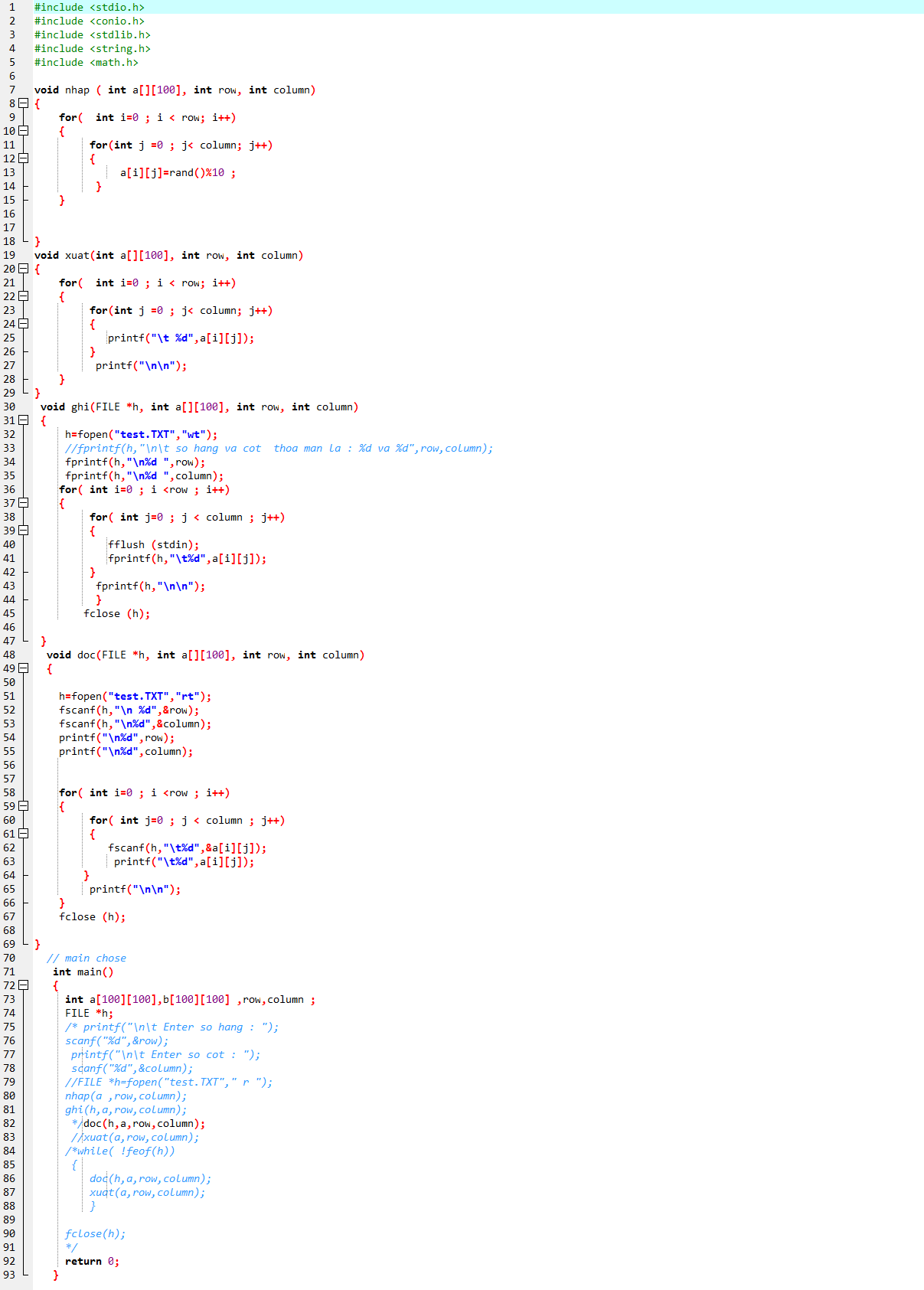
**



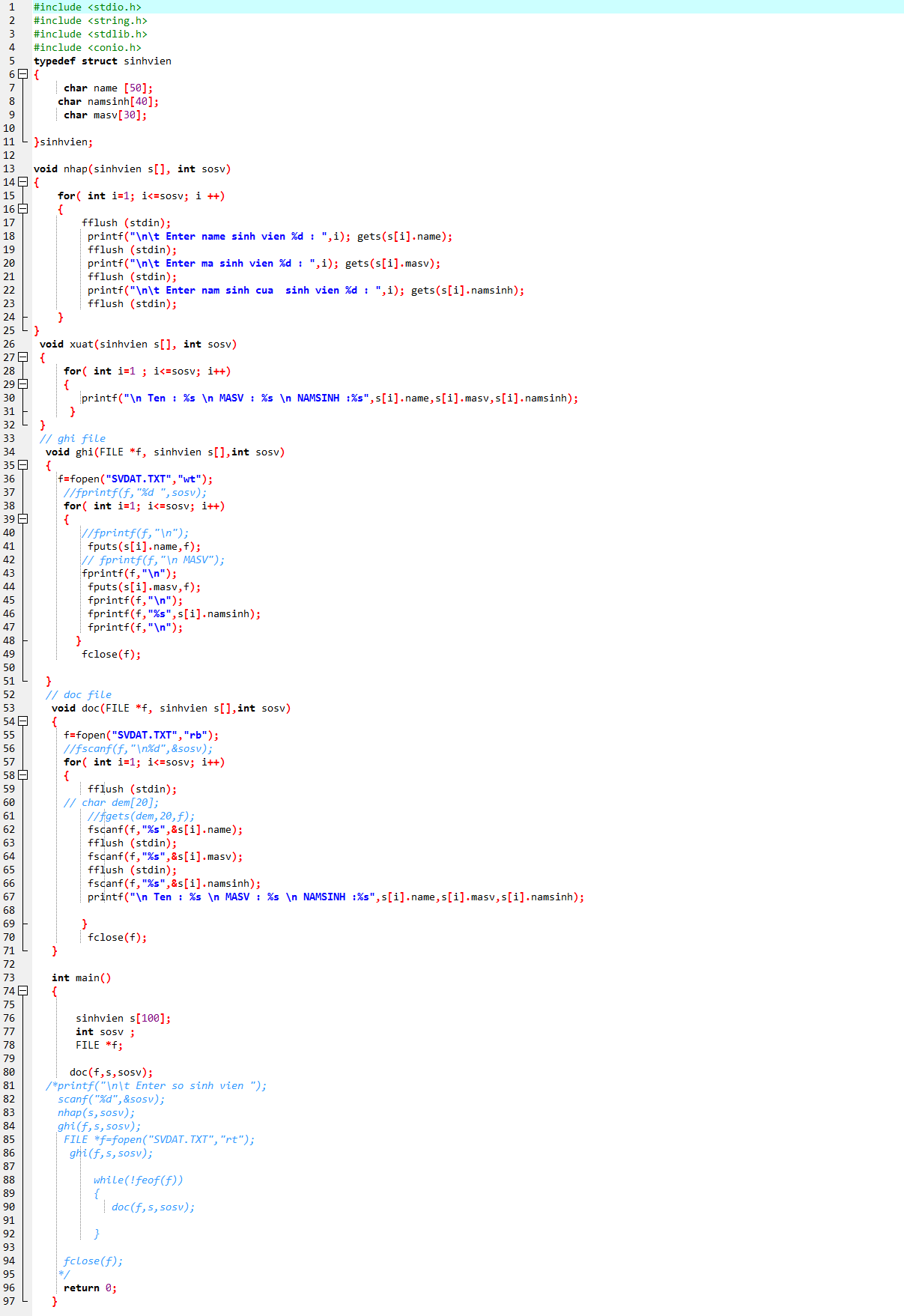
**BÀI 2:**Viết chương trình phát sinh ngẫu nhiên ma trận a kích thước 5x6,lưu ma trận này vào file **test.inp**. Đọc lại file **test.inp** đưa dữ liệu vào ma trận b và xuất ra màn hình xem kết quả lưu đúng không? Cấu trúc của file **test.inp** như sau:

- Dòng đầu lưu 2 số nguyên: m, n thể hiện số dòng và số cột của ma trận.

- m dòng tiếp theo, mỗi dòng gồm n phần tử là giá trị các phần tử trên 1 dòng của ma trận.



**BÀI 3:** Viết hàm đọc/ghi một danh sách sinh viên của một lớp vào tập tin SV.DAT



**BÀI 4:**ĐẾM TỪ KHÁC NHAU

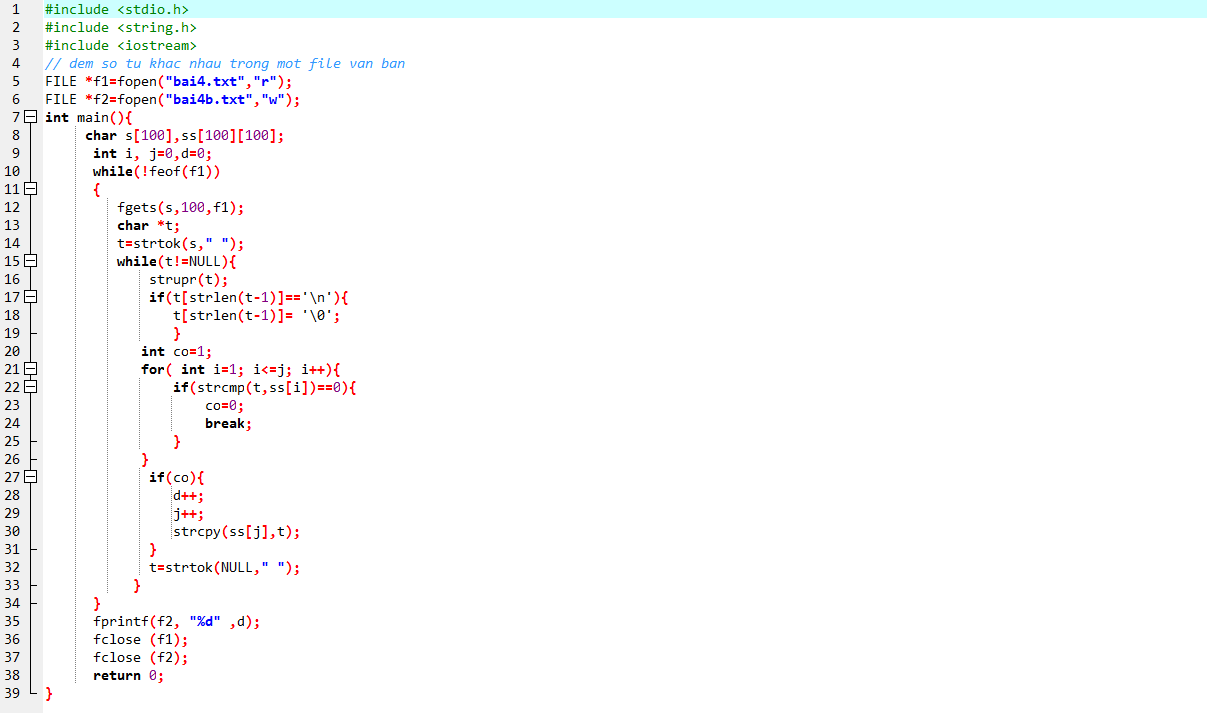
Viết chương trình đếm số từ khác nhau của một file văn bản.

## Dữ liệu vào: Gồm nhiều dòng, chỉ bao gồm các chữ cái.

## Kết quả: Viết ra số từ khác nhau trong file.

## Ví dụ:

|  |  |
| --- | --- |
| **Dữ liệu vào** | **Kết quả** |
| Xin chao cac ban  Xin moi cac ban tap trung hoc tap | 8 |



**BÀI 5:**

Cho tập các số tự nhiên ghi lại trong file data.in theo từng dòng; mỗi dòng ghi lại nhiều nhất 10 số; hai số được viết cách nhau một vài khoảng trống; mỗi số tự nhiên có thể xuất hiện nhiều lần ở những vị trí khác nhau trong file. Ta gọi tần xuất xuất hiện số tự nhiên X trong file data.in là trong đó N(X) là số lần xuất hiện số tự nhiên X trong file data.in, K là số các số tự nhiên trong file data.in. *Sử dụng CTDL mảng*, hãy viết chương trình tìm số vừa nguyên tố vừa thuận nghịch X có P(X) lớn nhất đầu tiên tìm được trong file data.in. Ví dụ với file data.in được cho dưới đây, ta tìm được số X = 131 vừa là số nguyên tố, vừa là số thuận nghịch với tần xuất xuất hiện lớn nhất là P(X) = 3/30 = 0.1

## data.in

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 131 | 171 | 13 | 37 | 27 | 30 | 23 | 77 | 444 |
| 10 | 131 | 171 | 20 | 23 | 77 | 23 | 27 | 77 | 444 |
| 10 | 131 | 171 | 20 | 23 | 77 | 23 | 27 | 77 | 444 |



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |

**BÀI 6:** *TÌM TỪ THUẬN NGHỊCH DÀI NHẤT TRONG FILE VĂN BẢN*

Cho một file văn bản bất kỳ. Hãy tìm ra từ thỏa mãn tính chất ***thuận nghịch có độ dài lớn nhất*** trong file đó và cho biết từ đó ***xuất hiện bao nhiêu lần***. Nếu có nhiều từ cùng có độ dài lớn nhất thì in ra tất cả các từ đó theo thứ tự xuất hiện trong file ban đầu.

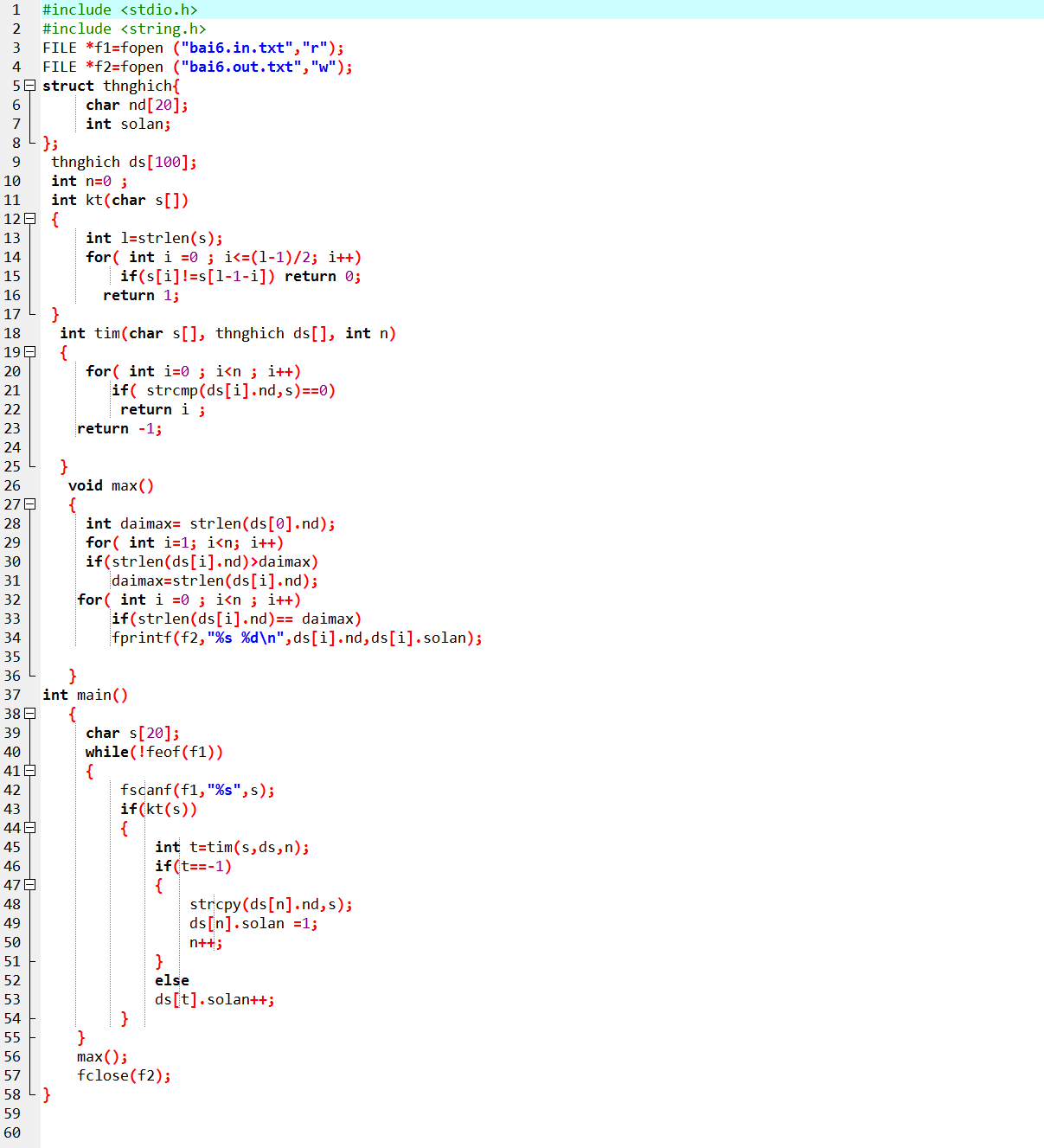
## Dữ liệu vào: Gồm một đoạn văn bản bất kỳ. Không quá 1000 từ.

**Kết quả** (ghi ra màn hình):

## Ví dụ:

* Ghi ra trên một dòng từ thuận nghịch có độ dài lớn nhất và số lần xuất hiện của nó.
* Nếu có nhiều từ cùng có độ dài lớn nhất thì các từ được liệt kê theo thứ tự xuất hiện ban đầu.

|  |  |
| --- | --- |
| **Dữ liệu vào** | **KẾT QUẢ** |
| AAA BAABA HDHDH ACBSD SRGTDH DDDDS DUAHD AAA AD DA HDHDH AAA AAA AAA AAA DDDAS HDHDH HDH AAA AAA AAA AAA AAA AAA AAA AAA  DHKFKH DHDHDD HDHDHD DDDHHH HHHDDD TDTD | HDHDH 3 |



**BÀI 7:** *TÌM TỪ DÀI NHẤT TRONG FILE*

Cho một file văn bản bất kỳ. Hãy tìm ra từ có độ dài lớn nhất trong file. Nếu có nhiều từ khác nhau có độ dài bằng nhau và bằng giá trị lớn nhất thì in ra tất cả các từ đó theo thứ tự xuất hiện trong file dữ liệu vào (nhưng một từ dù xuất hiện nhiều lần cũng chỉ được liệt kê một lần).

## Dữ liệu vào:

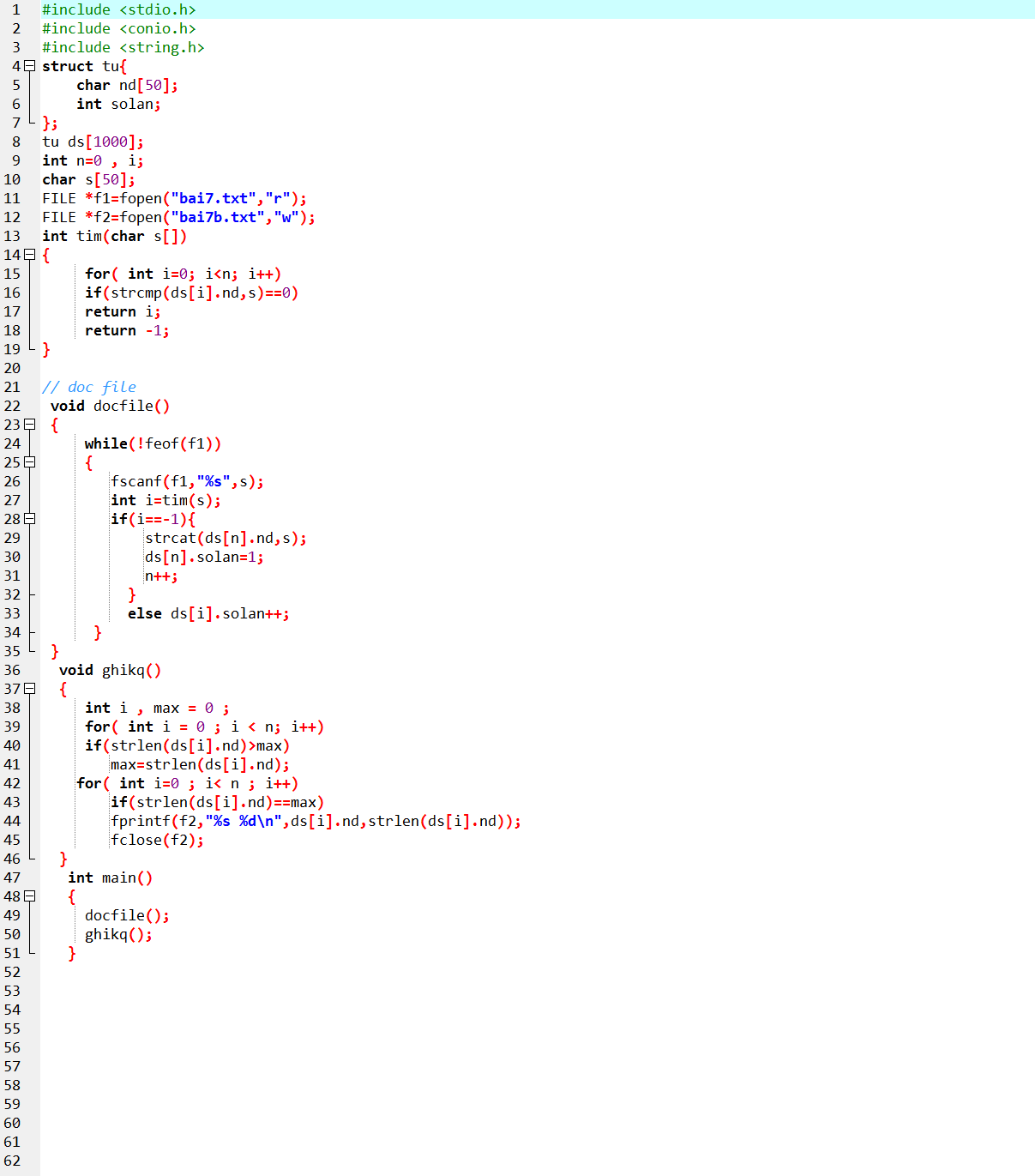
Gồm một đoạn văn bản bất kỳ. Không quá 1000 từ.

## Kết quả:

Từ dài nhất và độ dài của nó, cách nhau một khoảng trống. Nếu có nhiều từ như vậy thì liệt kê lần lượt các từ theo thứ tự xuất hiện trong file ban đầu.

## Ví dụ:

|  |  |
| --- | --- |
| **Dữ liệu vào** | **KẾT QUẢ** |
| Tiet hoc cuoi cung da ket thuc. Mon hoc Tin hoc co so 2 da ket thuc. Cac ban co gang on tap tot de thi dat ket qua cao. Chuc cac ban ngay cang gat hai duoc nhieu thanh cong tren con duong da chon | thuc. 5  nhieu 5  thanh 5  duong 5 |



**BÀI 8:**

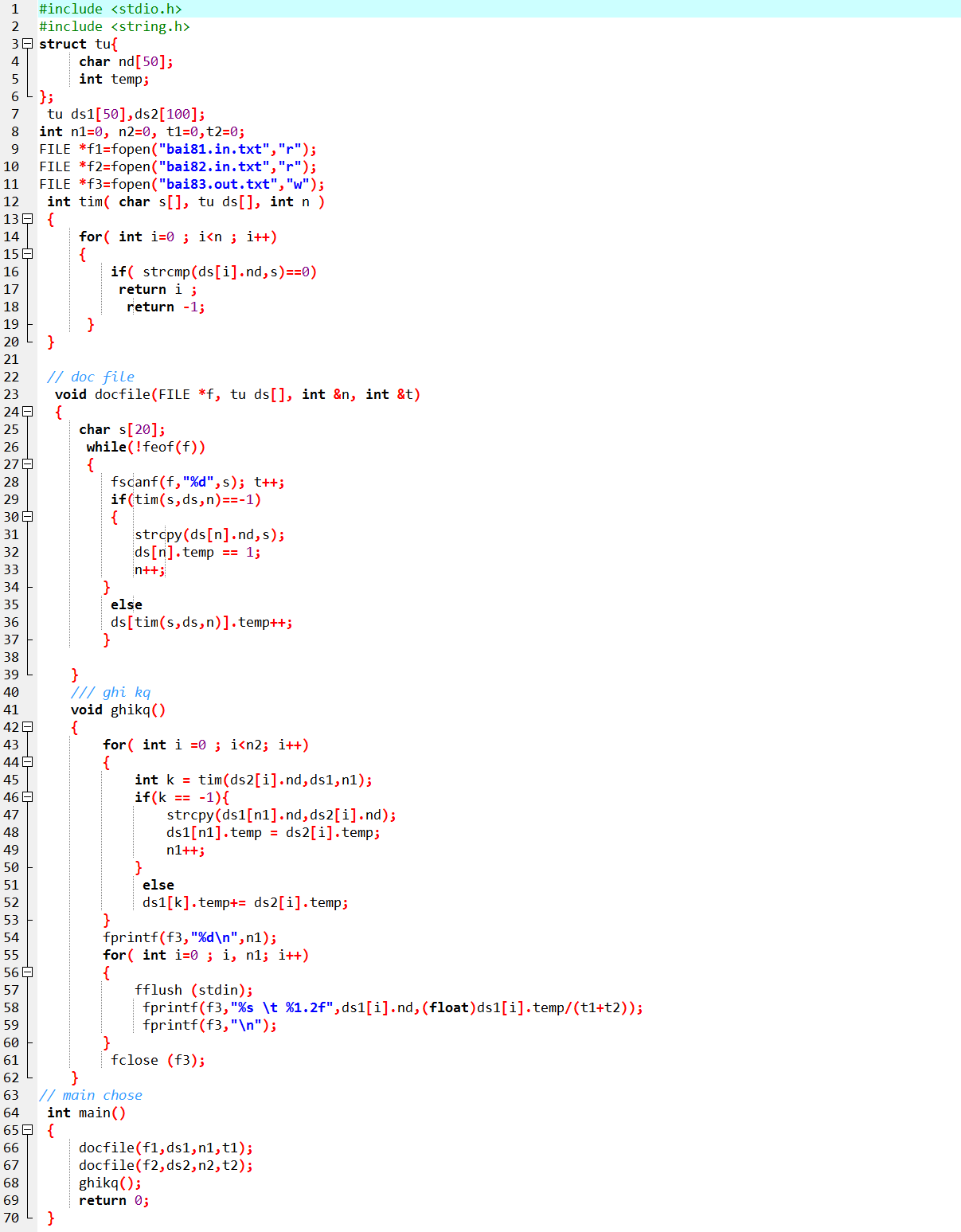
Ta định nghĩa một từ là dãy các kí tự không chứa khoảng trống (space), dấu tab, dấu xuống dòng (‘\n’), dấu về đầu dòng (‘\r’) và dấu kết thúc dòng (‘\0’). Tần xuất xuất hiện của từ *W* trong tập văn bản *D*1 *và D*2, ký hiệu là *P*(*W*) được tính theo công thức: trong đó *N* (*W*) là số lần xuất hiện từ *W* trong *D*, *N*(*D* ) là tổng số từ của tập văn bản *D*i (i=1,2).

Cho hai file văn bản *data1.in* và *data2.in*. *Sử dụng CTDL Mảng*, hãy tìm tập các từ và tần xuất xuất hiện của các từ xuất hiện trong file *data1.in* nhưng không xuất hiện trong file *data2.in*. Tập các từ tìm được ghi lại trong file Ketqua.out theo khuôn dạng:

* + Dòng đầu tiên ghi lại số tự nhiên *K* là số từ *W* tìm được theo yêu cầu của bài toán.
  + *K* dòng kế tiếp, mỗi dòng ghi lại một từ *W* và tần xuất xuất hiện *P*(*W*) thỏa mãn yêu cầu của bài toán. *W* và *P*(*W*) được viết cách nhau bởi một vài khoảng trống.

Ví dụ với file data1.in và data2.in dưới đây sẽ cho ta file Ketqua.out của bài toán.

|  |  |  |
| --- | --- | --- |
| **data1.in** | **data2.in** | **Ketqua.out** |
| AB AC AD AE AF AB AC AD AE AF | AB AC AD AH AK AB AC AD AH AK | 7  AB 0.2  AC 0.2  AD 0.2  AE 0.1  AF 0.1  AH 0.1  AK 0.1 |

**BÀI**

**9:** *SẮP XẾP THÍ SINH*

Hãy sắp xếp danh sách thí sinh đã có trong file theo tổng điểm giảm dần. Mỗi thí sinh gồm các thông tin:

* Mã thí sinh: là một số nguyên, tự động tăng. Tính từ 1.
* Tên thí sinh, ngày sinh
* Điểm môn 1, điểm môn 2, điểm môn 3

## Dữ liệu vào:

Dòng đầu chứa số thí sinh. Mỗi thí sinh viết trên 3 dòng:

* + Dòng 1: Tên thí sinh
  + Dòng 2: Ngày sinh
  + Dòng 3,4,5: 3 điểm thi tương ứng. Các điểm thi đều đảm bảo hợp lệ (từ 0 đến 10).

## Kết quả:

In ra danh sách thí sinh đã sắp xếp theo tổng điểm giảm dần. Nếu 2 thí sinh bằng điểm nhau thì thí sinh nào xuất hiện trước trong file sẽ viết trước. Mỗi thí sinh viết trên một dòng gồm: mã, tên, ngày sinh và tổng điểm. Các thông tin cách nhau đúng 1 khoảng trống. Điểm tổng được làm tròn đến 1 số sau dấu phẩy.



## Ví dụ

|  |  |
| --- | --- |
| **Dữ liệu vào** | **Kết quả** |
| 3  Nguyen Van A 12/12/1994  3.5  7.0  5.5  Nguyen Van B 1/9/1994  7.5  9.5  9.5  Nguyen Van C 6/7/1994  4.5  4.5  5.0 | 2 Nguyen Van B 1/9/1994 26.5  1 Nguyen Van A 12/12/1994 16.0  3 Nguyen Van C 6/7/1994 14.0 |

**BÀI 10** *SẮP XẾP MẶT HÀNG*

Hãy sắp xếp danh sách các mặt hàng đã có trong file theo lợi nhuận giảm dần. Mỗi mặt hàng gồm các thông tin:

* Mã mặt hàng: là một số nguyên, tự động tăng. Tính từ 1.
* Tên mặt hàng, nhóm hàng: là các xâu ký tự
* Giá mua, giá bán: là các số thực (không quá 9 chữ số)

## Dữ liệu vào: Dòng đầu chứa số mặt hàng. Mỗi mặt hàng viết trên 4 dòng:

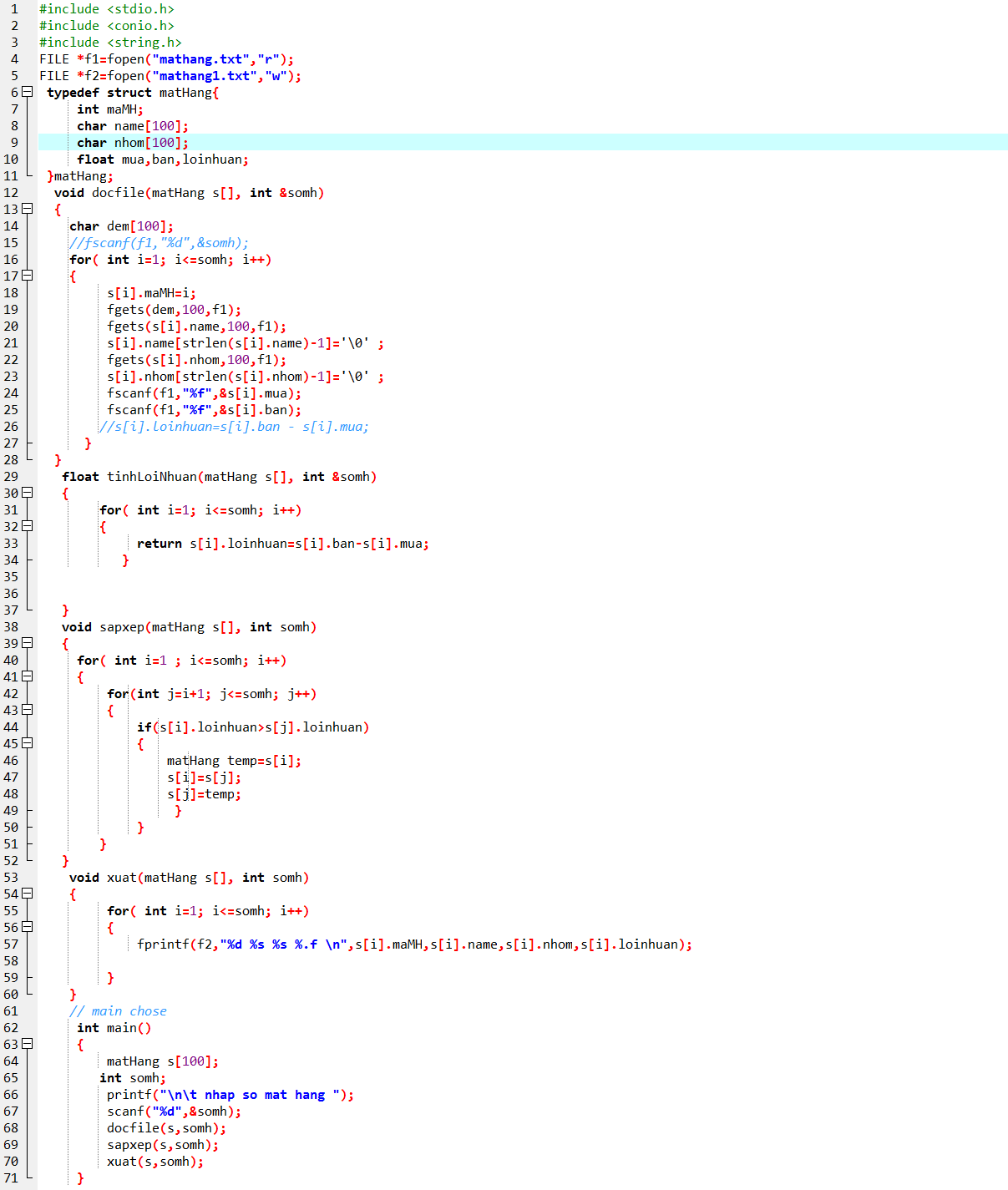
* + Dòng 1: Tên mặt hàng
  + Dòng 2: Nhóm hàng
  + Dòng 3: Giá mua.
  + Dòng 4: Giá bán

## Kết quả:

In ra danh sách mặt hàng đã sắp xếp theo lợi nhuận giảm dần (lợi nhuận tính bằng giá bán trừ đi giá mua). Nếu 2 mặt hàng lợi nhuận bằng nhau thì mặt hàng nào xuất hiện trước trong file sẽ viết trước. Mỗi mặt hàng viết trên một dòng gồm: mã, tên, nhóm hàng và lợi nhuận. Các thông tin cách nhau đúng 1 khoảng trống.

## Ví dụ

|  |  |
| --- | --- |
| **Dữ liệu vào** | **Kết quả** |
| 3  May tinh SONY VAIO Dien tu  16400  17699  Tu lanh Side by Side  Dien lanh 18300  25999  Banh Chocopie Tieu dung 27.5  37 | 2 Tu lanh Side by Side Dien lanh 7699  1 May tinh SONY VAIO Dien tu 1299  3 Banh Chocopie Tieu dung 10.5 |

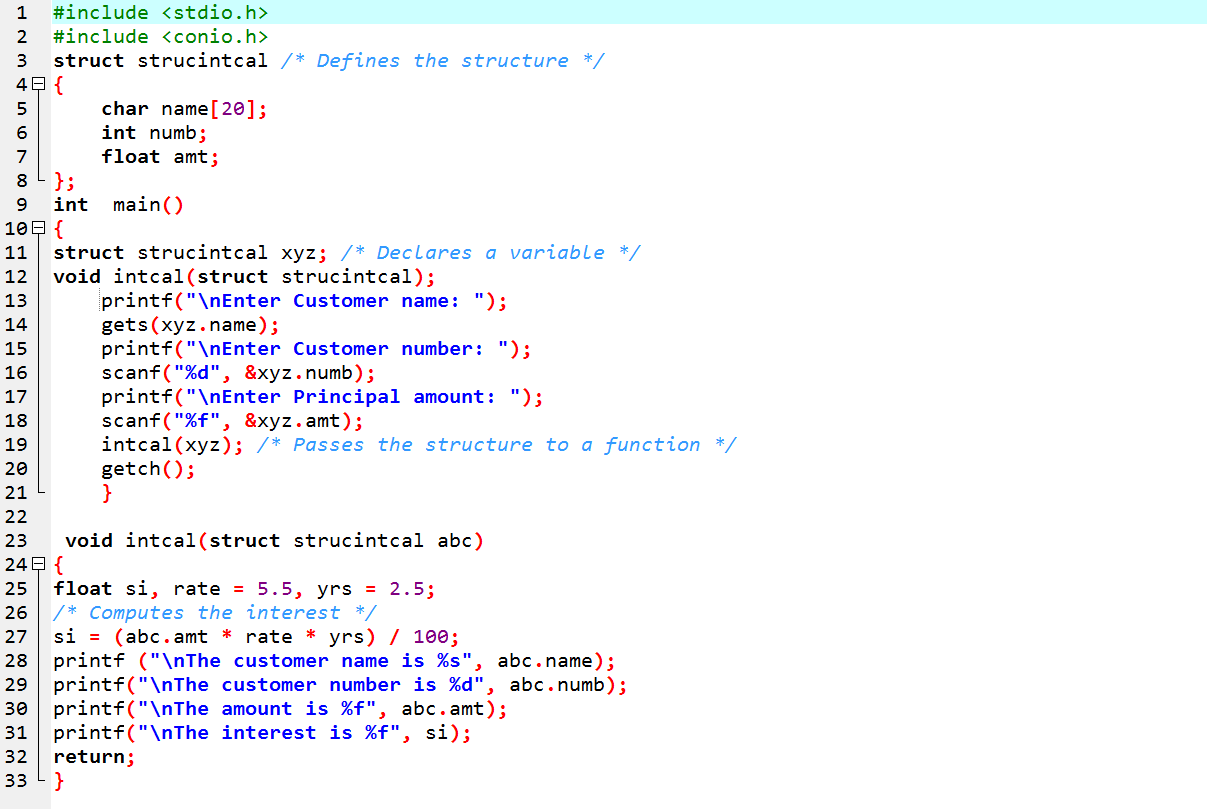


**BÀI TẬP STRUCT**

**BÀI 1.** Viết chương trình nhập vào toạ độ hai điểm trong mặt phẳng và tính tổng hai toạ độ này.

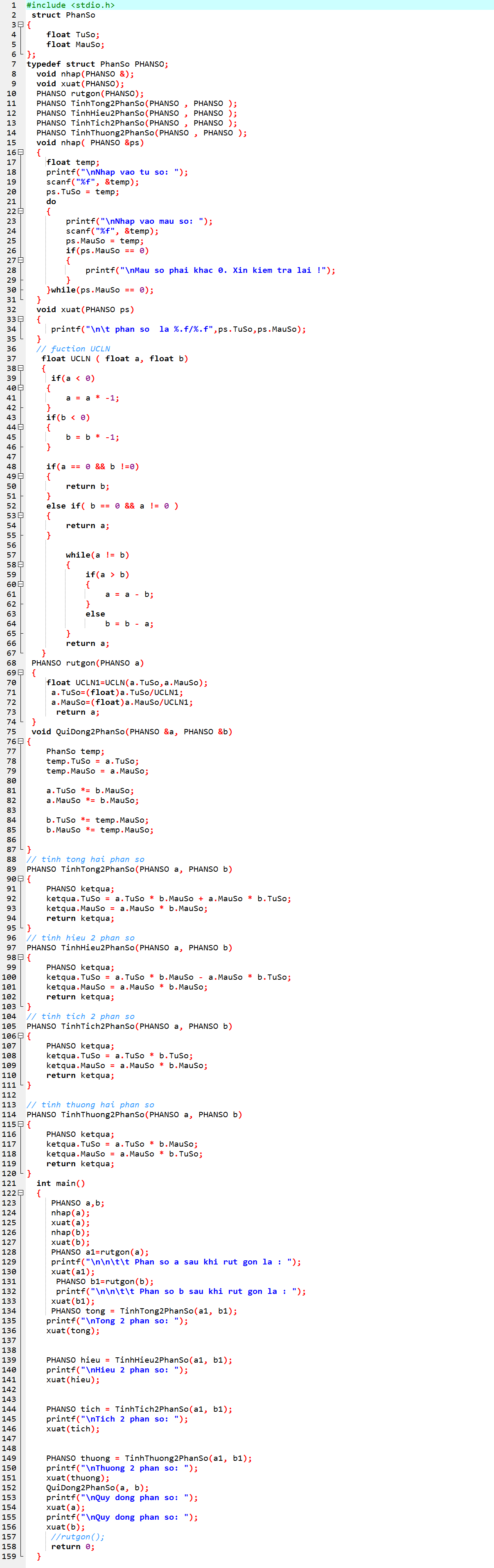


**BÀI 2.** Nhập vào 1 danh sách các sinh viên gồm các thông tin: Mã SV, họ tên , năm sinh, giới tích và điểm số 3 môn học. (Struct). In ra thông tin các sinh viên đó.



**BÀI 3.** Xây dựng các thao tác sau cho hai phân số:

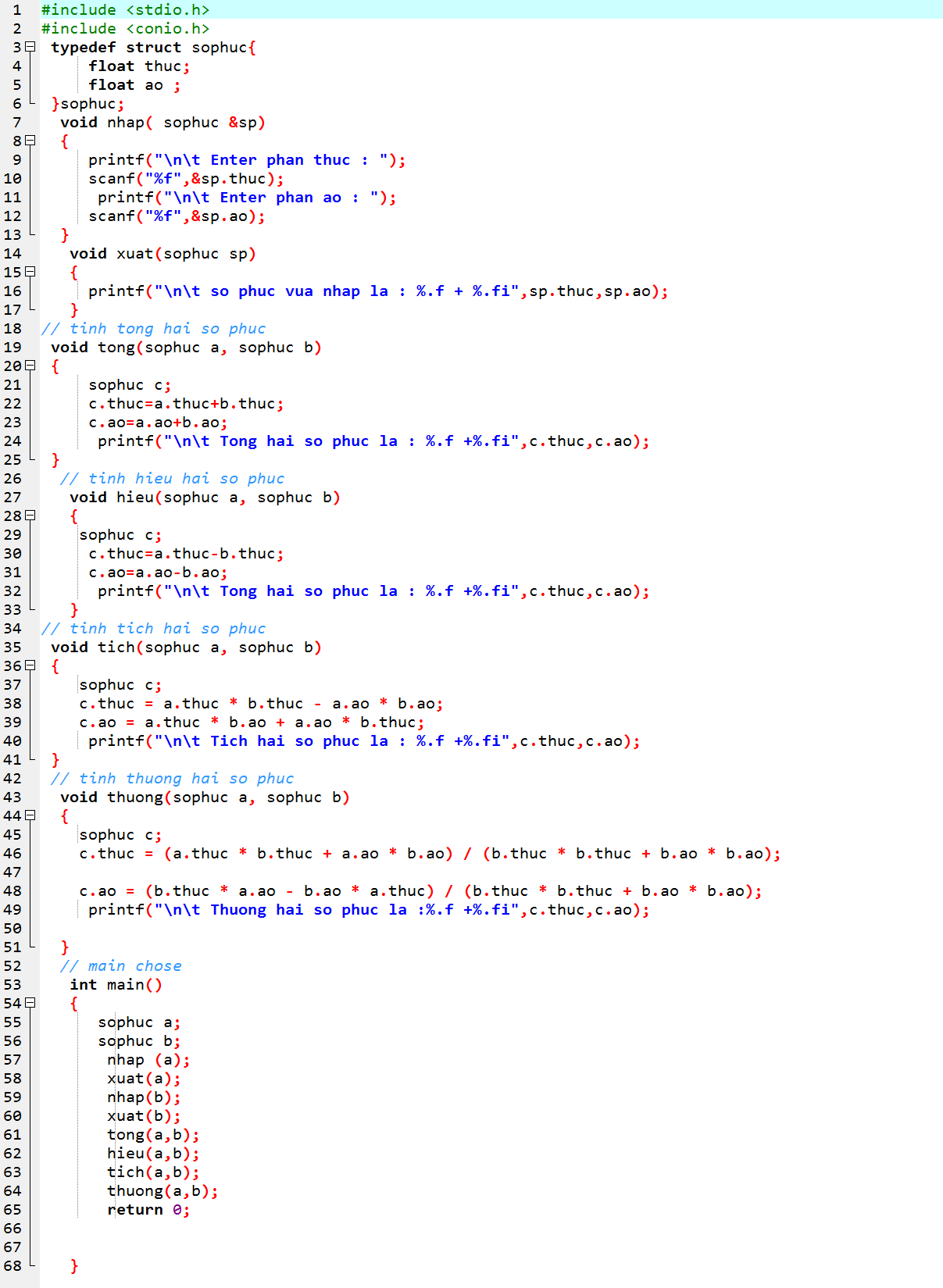
1. Tạo lập 2 phân số.
2. Rút gọn phân số
3. Quy đồng hai phân số
4. Tổng, hiệu, tích, thương 2 phân số.



**BÀI 4.** Xây dựng các thao tác sau cho hai số phức:

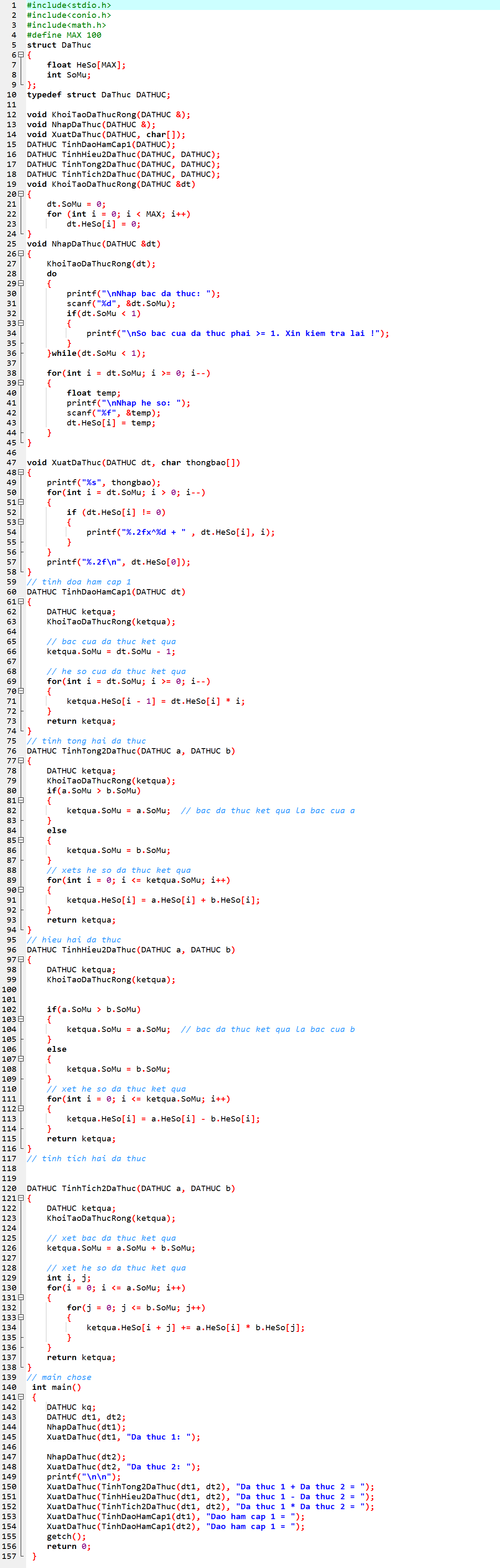
1. Tạo lập 2 số phức.

2. Tổng, hiệu, tích, thương 2 số phức.



**BÀI 5.** Cho hai đa thức Pn(x) và Qm(x). Hãy viết chương trình thực hiện những thao tác sau:

1. Tạo lập hai đa thức
2. Tính Pn(x0) và Qm(x0)
3. Tìm đạo hàm cấp l ≤n của đa thức.
4. Tìm Pn(x) + Qm(x)
5. Tìm Pn(x) - Qm(x)
6. Tìm Pn(x) / Qm(x) và đa thức dư



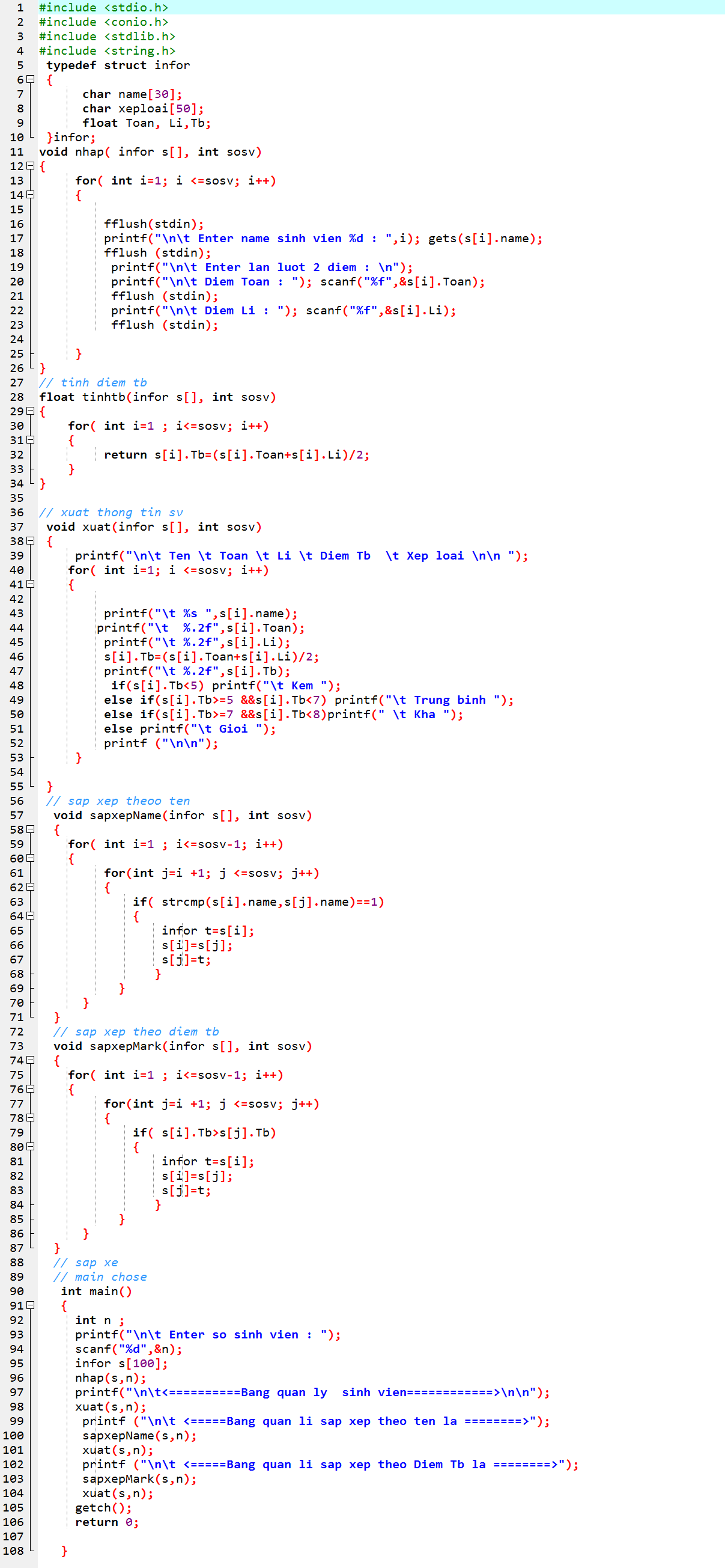
**BÀI 6.** Xây dựng chương trình quản lý sinh viên, sử dụng cấu trúc sau:

struct SV {char ten[25]; float toan, ly, tb; char XepLoai[20]};

Nhập danh sách cho n sinh viên (n là số tự nhiên được nhập vào), chỉ nhập ten và toan, ly.

Tính giá trị trường tb =(toan+ly)/2, sau đó tính giá trị trường xếp loại theo cách sau:

XepLoai=”kem” nếu tb<5, = “Trung binh” nếu 5≤tb<7, = “Kha” nếu 7≤tb<8 và = “Gioi” nếu 8≤tb. Sắp xếp lại danh sách theo trường XepLoai.



**BÀI 7.** Xây dựng chương trình quản lý học sinh. Mỗi học sinh quản lý các thông tin sau:

Họ tên, Năm sinh, Điểm trung bình.Viết chương trình thực hiện các chức những sau:

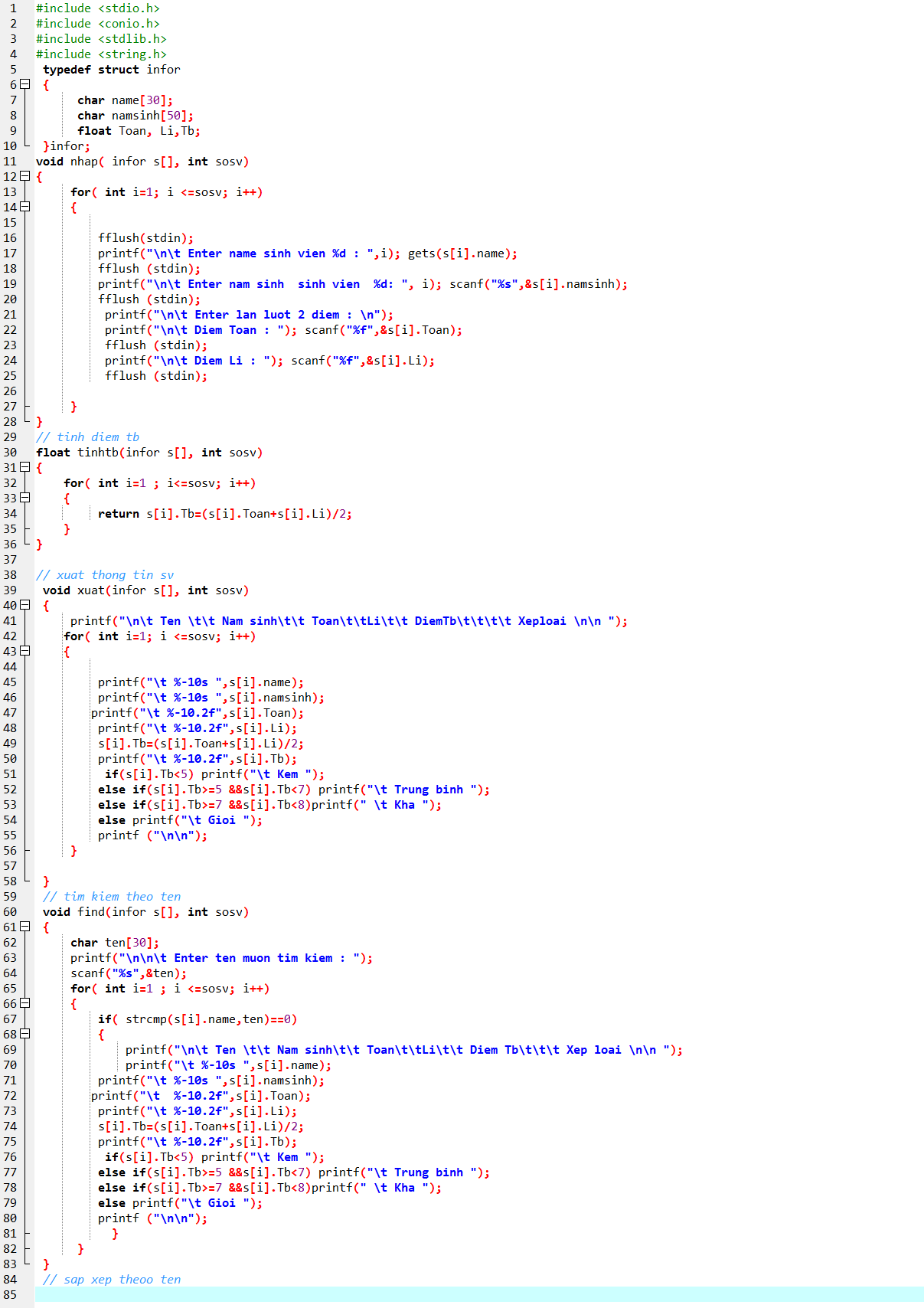
- Nhập số liệu, mỗi lần có thể nhập m học sinh, m>0

- Xem danh sách: trên màn hình.

- Tìm kiếm :theo tên, theo năm sinh, theo tên và năm sinh.

- Sắp xếp: theo tên, theo điểm trung bình.

- Xóa khi biết tên





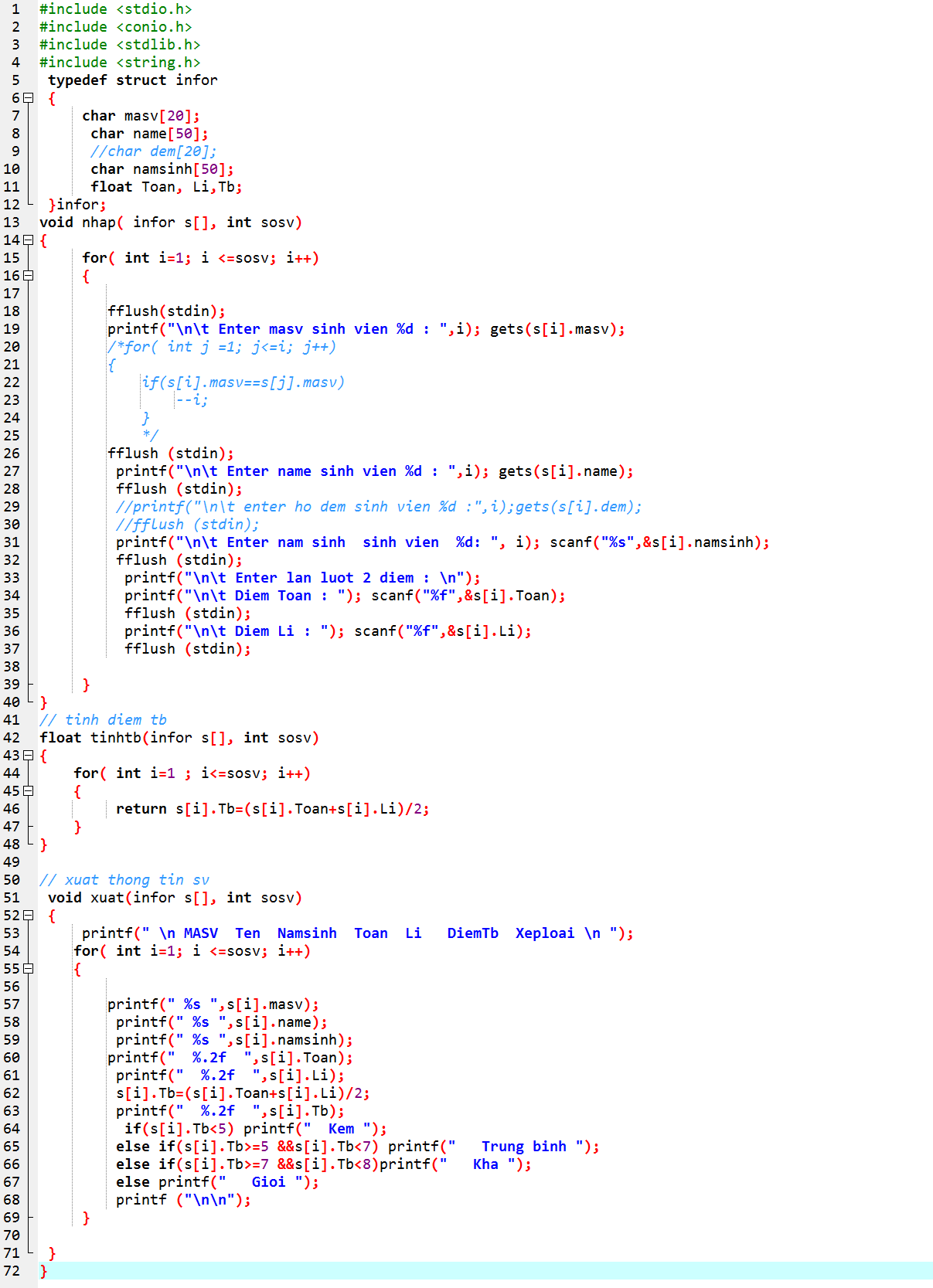
**BÀI 8.** Viết chương trình quản lý danh sách của một lớp học có không quá 500 sinh viên.

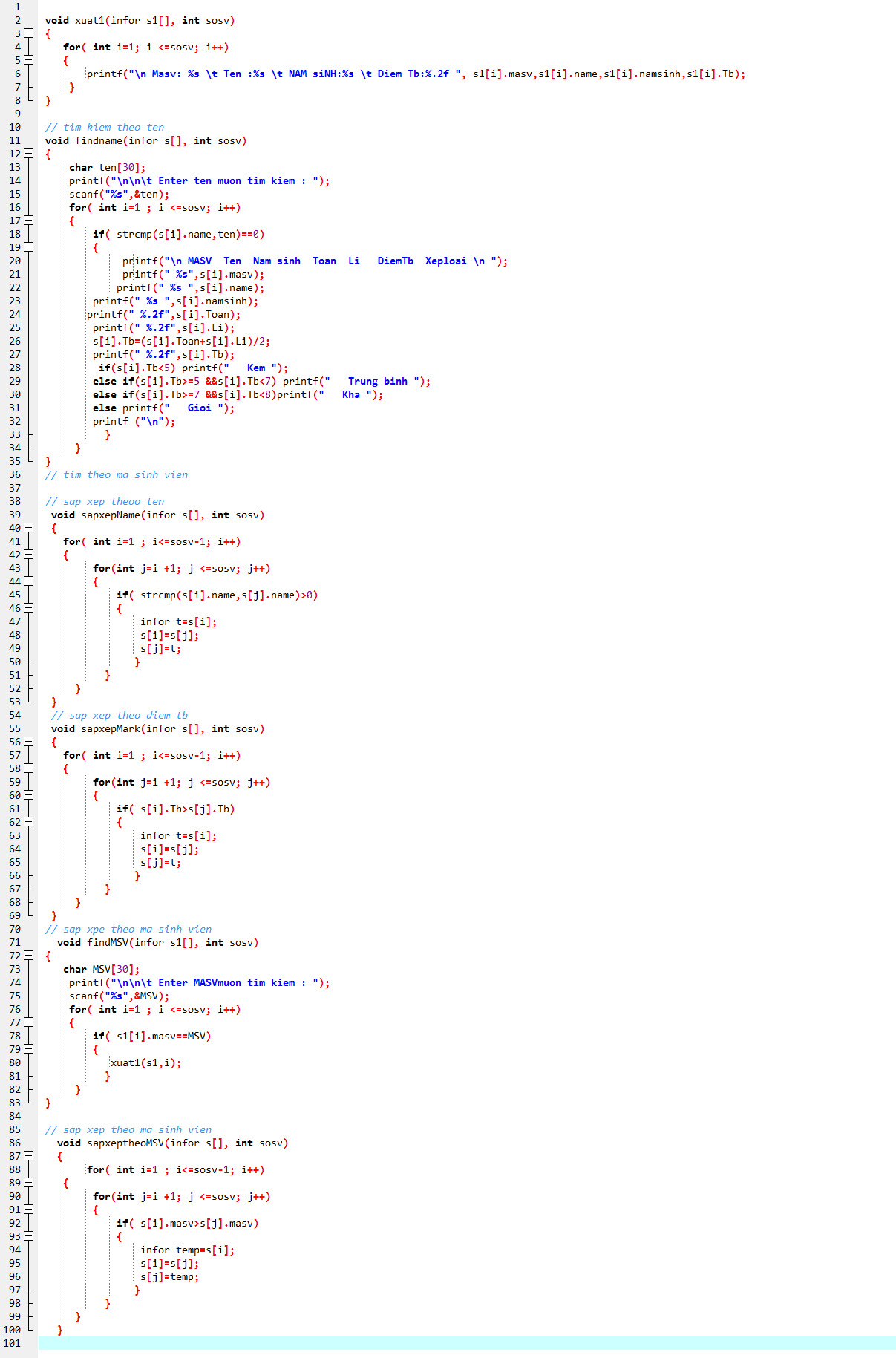
Thông tin của mỗi sinh viên gồm:

* Mã sinh viên: đúng 8 ký tự, không có khoảng trắng.
* Họ và tên: tối đa 30 ký tự (tên: 10, họ và chữ lót: 20)
* Ngày tháng năm sinh
* Giới tính: Nam/Nữ
* Điểm các môn học: dãy gồm đúng 10 điểm số của 10 môn học.

Chương trình phải đáp ứng được các yêu cầu:

* **Có menu cho người dùng chọn các chức năng tương ứng.**
* Nhập mới danh sách sinh viên.
* Xuất danh sách SV ra màn hình theo dạng bảng ngang (mỗi SV trên một dòng).
* Thêm một sinh viên vào danh sách.
* Sắp xếp danh sách tăng dần theo Mã SV hoặc theo Họ và Tên (tùy người dùng chọn).
* Tìm thông tin một SV dựa vào mã SV
* Ghi danh sách ra file DS.DAT.
* Đọc danh sách từ file DS.DAT.
* Loại bỏ một SV ra khỏi danh sách.



**ASS05-OPT1** 

