

# Bài Tập Thực Hành Cấu Trúc Dữ Liệu & Giải Thuật

Khối: Cao Đẳng nghề và Trung Cấp  
Năm 2009



## Hướng dẫn:

- Bài tập thực hành dựa trên giáo trình: C & Data Structure
- Bài tập thực hành được chia theo làm nhiều Module
- Mỗi Module được thiết kế cho thời lượng 4-6 tiết **thực hành tại lớp** với sự hướng dẫn của giảng viên.
- Tùy theo số tiết phân bổ, mỗi tuần học có thể thực hiện nhiều Module.
- Sinh viên phải làm tất cả các bài tập trong các Module ở tuần tương ứng. Những sinh viên chưa hoàn tất phần bài tập tại lớp có trách nhiệm tự làm tiếp tục ở nhà.
- Các bài có dấu (\*) là các bài tập nâng cao dành cho sinh viên khá giỏi.

## Module 1

### Bài 1

Viết hàm tính tổng  $S=1+1/2+1/3\dots+ 1/N$ . Sử dụng hàm để tính S khi người dùng nhập N.

### Bài 2

Nhập 2 số a và n. Tính  $S = a^1 + a^2 + a^3 + \dots + a^n$ .

### Bài 3

Viết chương trình tính tổng sau:  $S = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$

### Bài 4

Viết chương trình tính tổng sau:  $S = 1/1! + (1+2)/2! + (1+2+3)/3! + \dots + (1+2+3+\dots+n)/n!$

### Bài 5

Viết chương trình tính tổng sau:  $S = 1 + x/1! + x^2/2! + x^3/3! + \dots + x^n/n!$

### Bài 6

Viết chương trình tính tổng sau:  $S = 1 - x + x^2 - x^3 + \dots (-1)^n x^n$

### Bài 7

Viết một hàm tính tổng các chữ số của một số nguyên.

Viết chương trình nhập vào một số nguyên, dùng hàm trên kiểm tra xem số đó có chia hết cho 3 không. (Một số chia hết cho 3 khi tổng các chữ số của nó chia hết cho 3)

### Bài 8

Bài toán tính tiền lãi gửi ngân hàng: Nhập tiền vốn ban đầu, số tháng gửi N và lãi suất hàng tháng. Tính số tiền nhận được sau mỗi tháng gửi biết rằng tiền lãi hàng tháng được gộp vào tiền vốn. Số tiền nhận được sau mỗi tháng sẽ hiển thị trên màn hình dạng:

Tháng 1: 100

Tháng 2: 102

...

*Ví dụ, tiền vốn là 100, lãi suất tháng là 2%. Sau 1 tháng gửi sẽ có số tiền là:*

*Số tiền =  $100 + 100 \cdot 0.02 = 102$*

*Sau 2 tháng gửi sẽ có số tiền là: Số tiền =  $102 + 102 \cdot 0.02 = 104.04$*

*Công thức tính tiền thu được sau mỗi tháng gửi là: Số tiền := Tiền vốn + Tiền vốn \* Lãi suất*

***Số tiền này lại trở thành tiền vốn của tháng sau, tức là:***

***Tiền vốn := Số tiền;***

*Quy trình cứ lặp đi lặp lại từ tháng 1 đến tháng N.*

### Bài 9

Trở lại bài toán tính tiền gửi ngân hàng có tiền lãi hàng tháng gộp vào vốn (bài tập trên). Câu hỏi bây giờ là: cần gửi tối thiểu là bao nhiêu tháng để có được số tiền > S cho trước.

## Module 2

### Bài 1

Nhập 2 số  $n$  và  $k$  nguyên. Tính tổ hợp chập  $k$  của  $n$  theo công thức như sau  $C(k,n)=n!/(k!(n-k)!)$

### Bài 2

Tìm các số Fibonacci. Dãy số Fibonacci  $\{ 1, 1, 2, 3, 5, 8, 13, 21, \dots \}$  được là dãy được nhắc nhiều trong khối kỹ thuật, nó được xây dựng như sau:

$U_0=1, U_1=1, U_k=U_{k-1} + U_{k-2}$  với mọi  $k= 2, 3, 4, \dots$

- Viết chương trình xuất ra màn hình số Fibonacci ở vị trí thứ  $n$  (sử dụng đệ quy)
- Viết chương trình xuất ra màn hình số Fibonacci ở vị trí thứ  $n$  (KHÔNG sử dụng đệ quy)
- Viết chương trình nhập số nguyên  $n$  và xuất ra  $n$  số Fibonacci đầu tiên

### Bài 3

- Nhập 2 số nguyên dương  $m$  và  $n$ . Tính Ước số chung nhỏ nhất và bội số chung lớn nhất của  $m$  và  $n$  (sử dụng đệ quy)
- Nhập 2 số nguyên dương  $m$  và  $n$ . Tính Ước số chung nhỏ nhất và bội số chung lớn nhất của  $m$  và  $n$  (không sử dụng đệ quy)

### Bài 4

Viết hàm *nhập số giây* và đổi thành *giờ : phút : giây*.

Ví dụ: nhập 3665 giây -> 1 giờ 1 phút 5 giây

### Bài 5

- Viết hàm kiểm tra 1 số có phải là số nguyên tố hay không (số chỉ chia hết cho 1 và chính nó)
- Viết hàm hiển thị tất cả các số nguyên tố bé hơn số  $n$  nhập từ bàn phím
- Viết hàm hiển thị tất cả các số nguyên tố bé hơn số  $n$  nhập từ bàn phím, in lên màn hình mỗi dòng 19 số
- Viết hàm hiển thị tất  $n$  số số nguyên tố đầu tiên tính từ 2 ( $n$  nhập từ bàn phím). . Ví dụ  $n=8$  thì kết quả là: 2,3,5,7,11,13,17,19

### Bài 6

Tìm số  $n$  bé nhất thỏa  $s = 1 + 2 + 3 + \dots + n > 10000$  và hãy cho biết khi đó  $s$  có giá trị là bao nhiêu ?

### Bài 7

Viết hàm dùng để phân tích một số nguyên dương ra thừa số nguyên tố.

Ví dụ: khi nhập  $n = 100$  thì hiển thị :  $100=5*5*2*2$

## Module 3

### Bài 1

VCT nhập vào 2 phân số  $a/b$  và  $c/d$ .

Hãy tính tổng của phân số này, yêu cầu là phân số kết quả phải ở dạng tối giản.

ví dụ :  $1/6+1/3=1/2$

### Bài 2

Viết một hàm đảo ngược thứ tự các phần tử của một mảng số nguyên.

Ví dụ: mảng nhập vào 1 2 3 4 5 7 9 10. sau khi đảo mảng thành 10 9 7 5 4 3 2 1

### Bài 3

Viết chương trình nhập vào một mảng số tự nhiên. Hãy xuất ra màn hình:

- Dòng 1 : gồm các số lẻ, tổng cộng có bao nhiêu số lẻ.
- Dòng 2 : gồm các số chẵn, tổng cộng có bao nhiêu số chẵn.
- Dòng 3 : gồm các số nguyên tố.
- Dòng 4 : gồm các số không phải là số nguyên tố

### Bài 4

Viết chương trình nhập vào một mảng, hãy xuất ra màn hình:

- Phần tử lớn nhất của mảng.
- Phần tử nhỏ nhất của mảng.
- Tính tổng của các phần tử trong mảng .
- Tính trung bình cộng
- Tính tổng của các phần tử là số nguyên tố trong mảng .
- Tính số lượng phần tử là số nguyên tố trong mảng .
- Phần tử âm lớn nhất của mảng.
- Phần tử dương nhỏ nhất của mảng.
- Tổng các phần tử có căn bậc hai nguyên .
- Gồm các số lẻ, tổng cộng có bao nhiêu số lẻ.
- Gồm các số chẵn, tổng cộng có bao nhiêu số chẵn.
- Kiểm tra tính đối xứng của mảng
- Tìm phần tử là số nguyên tố đầu tiên trong mảng
- Tìm 2 phần tử là 2 số nguyên tố đầu tiên trong mảng

## Module 4

### Bài 1

Tổ chức và xây dựng 2 hàm : GiảiPT\_bac1 và GiảiPT\_bac2 với các tham số hợp lý.  
Sử dụng 2 hàm này để giải pt bậc 2.

### Bài 2

Tổ chức hàm với các tham số hợp lý dùng để tính:

- Một hàm dùng để tính chu vi và diện tích hình tròn từ bán kính
- Một hàm dùng để tính chu vi và diện tích hình chữ nhật từ chiều dài-rộng
- Một hàm dùng để tính MAXMIN của 2 số integer.
- Một hàm dùng để tính bội số chung nhỏ nhất – USC lớn nhất của 2 số nguyên dương.

### Bài 3

Viết chương trình tính lương cho các công nhân tại xưởng may. Mỗi công nhân sẽ có giờ vào và giờ ra trong một ngày. Tiền lương được tính như sau:

- Từ 5h-8h: mỗi giờ 20,000 đ
- Từ 8h-11h: mỗi giờ 15,000 đ
- Từ 11h-14h: mỗi giờ 30,000 đ
- Từ 14h-17h: mỗi giờ 22,000 đ
- Từ 17h-24h: mỗi giờ 40,000 đ

Chú ý: Xây dựng hàm với tham số hợp lý.

### Bài 4

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý gồm con trỏ pHead và pTail (đầu và cuối xâu). Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

- Thêm một node vào đầu danh sách
- Thêm một node vào cuối danh sách
- Thêm nhiều node vào đầu danh sách
- Thêm nhiều node vào cuối danh sách
- Hiển thị giá trị node thứ n
- Tìm một node dựa theo giá trị nhập vào
- Thêm một node vào sau một node nào đó (nhập giá trị để tìm)
- Đếm số lượng node trong xâu
- Hiển thị danh sách các node trong xâu
- Hủy một node (nhập giá trị cần tìm)
- Hủy toàn bộ danh sách
- Tính tổng giá trị toàn bộ danh sách
- Tìm giá trị lớn nhất và nhỏ nhất trong danh sách
- Sắp xếp danh sách với giá trị tăng dần (tự chọn thuật toán)

### Bài 5

Viết chương trình xây dựng và quản lý danh sách liên kết đơn. Thành phần quản lý xâu chỉ gồm con trỏ pHead (đầu xâu). Hiển thị menu thực hiện các chức năng tương tự bài tập trên.

## Module 5

### Bài 1

Viết chương trình xây dựng và quản lý danh sách liên kết vòng đơn. Thành phần quản lý gồm con trỏ pHead. Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

- Hiển thị toàn bộ danh sách
- Thêm một node vào danh sách
- Tìm một node trong danh sách (giá trị nhập vào)
- Hiển thị các node là số nguyên tố
- Hủy một node trong danh sách (giá trị nhập vào- đầu tiên tìm thấy)
- Hủy tất cả các node là số nguyên tố
- Tăng giá trị mỗi node lên 2
- Tính tổng giá trị toàn bộ danh sách
- Tìm giá trị lớn nhất và nhỏ nhất trong danh sách
- Sắp xếp danh sách tăng dần
- Hủy toàn bộ danh sách

### Bài 2

Viết chương trình xây dựng và quản lý danh sách liên kết đôi. Thành phần quản lý gồm con trỏ pHead và pTail. Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

- Hiển thị giá trị toàn bộ danh sách theo thứ tự từ đầu đến cuối
- Hiển thị giá trị toàn bộ danh sách theo thứ tự từ cuối đến đầu
- Thêm một node vào đầu danh sách
- Thêm một node vào cuối danh sách
- Thêm một node vào sau một phần tử (giá trị nhập vào)
- Tìm một node trong danh sách.
- Đếm số lần xuất hiện của một node trong danh sách
- Hủy phần tử đầu danh sách
- Hủy phần tử cuối danh sách
- Hủy phần tử trong danh sách (giá trị nhập vào)
- Hiển thị các node là số nguyên tố
- Hủy một node trong danh sách (giá trị nhập vào- đầu tiên tìm thấy)
- Hủy tất cả các node là số nguyên tố
- Tăng giá trị mỗi node lên 2
- Tính tổng giá trị toàn bộ danh sách
- Tìm giá trị lớn nhất và nhỏ nhất trong danh sách
- Sắp xếp danh sách tăng dần
- Hủy toàn bộ danh sách

### Bài 3

Viết chương trình xây dựng và quản lý 2 danh sách liên kết đôi. Thành phần quản lý cho mỗi danh sách chỉ gồm con trỏ pHead. Hiển thị menu thực hiện các chức năng sau:

- Nhập n phần tử ngẫu nhiên vào mỗi danh sách (n nhập vào)
- Hiển thị giá trị từng danh sách, mỗi danh sách một hàng
- Ghép danh sách 2 vào phía sau danh sách 1, đồng thời hủy danh sách 2.
- Tìm giá trị lớn nhất của 2 danh sách
- Tính tổng giá trị 2 danh sách
- Chèn danh sách 2 vào giữa danh sách 1 (sau 1 phần tử với giá trị nhập vào)
- Sắp xếp 2 danh sách tăng dần.
- Ghép danh sách 2 vào danh sách 1 sao cho sau khi ghép, danh sách 1 vẫn tăng dần.

## Module 6

### Bài 1

Viết chương trình xây dựng và quản lý danh sách liên kết vòng đôi. Thành phần quản lý gồm con trỏ pHead. Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

- Thêm một node vào cuối danh sách (giá trị nhập vào)
- Lưu trữ danh sách xuống file text
- Đọc danh sách từ file text.
- Sắp xếp danh sách tăng dần.
- Tìm giá trị trung bình của danh sách

### Bài 2

Viết chương trình xây dựng và quản lý danh sách liên kết đôi. Thành phần quản lý gồm con trỏ pHead, pTail. Thành phần dữ liệu trong mỗi node là thông tin một sinh viên, bao gồm các trường:

- +Mã sinh viên (int)
- +Họ tên sinh viên (string)
- +Lớp (string)
- +Điểm Toán (float)
- +Điểm Lý (float)
- +Điểm Hóa (float)

Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm).

- Hiển thị toàn bộ danh sách
- Tìm một sinh viên theo mã sinh viên (nhập vào)
- Thêm một sinh viên vào cuối danh sách. Đảm bảo không có 2 sinh viên nào trùng mã.
- Xóa một sinh viên khỏi danh sách từ mã sinh viên (nhập vào)
- Nhập một lớp. Hiển thị danh sách sinh viên thuộc về lớp đó
- Tính tổng số sinh viên có điểm toán  $\geq 5$
- Hiển thị toàn bộ danh sách sinh viên chứa tên nhập vào
  - (Ví dụ: nhập tên cần tìm là Khuong, hiển thị toàn bộ sinh viên chứa tên Khuong)
- Hiển thị danh sách sinh viên yếu (có điểm trung bình  $\leq 4$ )
- Hiển thị danh sách sinh viên giỏi (có điểm trung bình  $\geq 8$  và không có môn học nào  $\leq 6$ )
- Sắp xếp danh sách tăng dần theo mã sinh viên
- Sắp xếp danh sách tăng dần theo điểm trung bình
- Sắp xếp danh sách tăng dần theo từng lớp, trong mỗi lớp tăng dần theo điểm trung bình
- Sắp xếp danh sách tăng dần theo từng lớp, trong mỗi lớp giảm dần theo điểm trung bình
- Sắp xếp danh sách tăng dần theo điểm toán, rồi đến điểm lý, rồi đến điểm hóa.
- Nhập một lớp. Hủy toàn bộ các sinh viên thuộc về lớp đó.
- Hủy tất cả sinh viên có học lực kém (điểm trung bình  $\leq 3$ ).
- Sắp xếp danh sách tăng dần theo mã sinh viên. Sau đó, khi thêm một sinh viên mới vào, chèn sinh viên này vào đúng vị trí sao cho danh sách sinh viên vẫn thỏa điều kiện tăng dần theo mã.
- Hủy toàn bộ danh sách
- Lưu trữ danh sách sinh viên này vào file text
- Nạp danh sách sinh viên từ file text.

## Module 7

### Bài 1

Sử dụng danh sách liên kết đơn tạo stack. Mỗi thành phần của stack gồm 2 thông tin: Tên Lớp, số số học sinh. Hệ thống menu gồm các mục

- +Lưu stack hiện tại vào file
- +Push phần tử mới (lớp mới) vào stack
- +Pop phần tử vào stack
- +Hiển thị danh sách của stack
- +Nạp stack từ file

Lưu ý:

- +Mỗi khi Push một lớp vào stack, nếu tên lớp đó chưa có, chương trình phải tạo phần tử mới.
- +Nếu lớp đó đã tồn tại thì không thêm lớp mới, mà chỉ cập nhật thêm số số sinh viên (cộng dồn số lượng sinh viên mới vào số số hiện tại).

### Bài 2

Sử dụng danh sách liên kết đôi để quản lý khách hàng cho một nhà ga.. Mỗi thành phần thông tin lưu trữ cho khách hàng gồm: số CMND khách hàng (10 ký tự), Tên khách hàng, Ga đến, giá tiền.

Hệ thống menu gồm các mục:

- +Nạp danh sách từ file
- +Thêm một khách hàng mới vào hàng đợi mua vé.
- +Bán một vé cho khách hàng. Chỉ bán cho người đăng ký trước.
- +Hiển thị danh sách khách hàng.
- +Hủy một khách hàng ra khỏi danh sách. (khách hàng không mua vé nữa).
- +Thống kê tình hình bán vé
- +Lưu danh sách vào file
- +Hiển thị danh sách các ga đang chờ mua vé.
- +Hiển thị danh sách các ga đang chờ mua vé và số vé tương ứng cho ga.

Lưu ý:

- +Số khách hàng trong danh sách hiện tại là số khách đang chờ, nhưng chưa có vé. Khi một khách hàng đã mua vé, thì loại khách hàng này ra khỏi danh sách chờ mua vé.
- +Việc mua vé phải có thứ tự: ai vào trước thì mua vé trước (FIFO).
- +Mỗi khi khách hàng mua được vé phải lưu lại khách hàng này để dùng cho việc thống kê.
- +Mỗi khi thêm một khác hàng mới, nếu Số CMND khách hàng đã có thì không tạo phần tử mới mà chỉ cập nhật lại ga và giá tiền đến cho khác hàng đó.
- +Mục thống kê tình hình: cho biết còn bao nhiêu khách hàng chờ nhận vé, bao nhiêu khách hàng đã nhận vé, tổng số tiền đã thu về là bao nhiêu.
- +Việc lưu danh sách: chỉ lưu các khách hàng chờ mua vé. Các khách hàng đã nhận vé xem như kết sổ trong ngày không cần lưu lại.
- +Khi chương trình vừa được chạy, lập tức tự động nạp toàn bộ danh sách khách hàng từ file (cách khách hàng chưa có vé).
- +Khi hiển thị danh sách các ga đến đang chờ mua vé, chỉ hiển thị tên ga đó một lần. (Ví dụ: giá sử 10 khách hàng nhưng đăng ký đi đến 2 ga, thì chỉ hiển thị 2 hàng).



## Module 8

### Bài 1

Viết chương trình xây dựng và quản lý cây nhị phân tìm kiếm (Binary Search Tree). Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

- Thêm một node vào cây (giá trị nhập vào). Nếu node này đã có giá trị thì thông báo không thêm vào node đã có.
- Tìm giá trị trung bình của danh sách
- Xuất danh sách. Khi menu này được chọn, hiển thị menu con cho phép chọn lựa
  - Xuất danh sách theo thứ tự preorder
  - Xuất danh sách theo thứ tự inorder
  - Xuất danh sách theo thứ tự postorder

### Bài 2

Sử dụng bài tập ở câu trên tiếp tục phát triển rộng các menu như sau.

- Lưu toàn bộ cây xuống file
- Nạp cây từ file
- Tính số lượng node của tree
- Tính chiều cao của cây
- Tìm giá trị nhỏ nhất
- Tìm giá trị lớn nhất
- Tìm một node theo giá trị nhập vào
- Hiển thị giá trị tăng dần toàn bộ cây
- Thống kê số lượng node: là số chẵn, là số lẻ là số nguyên tố

Chú ý: Khi người sử dụng thêm 1 node, chương trình phải tự động lưu xuống file ngay lập tức. Khi chương trình vừa khởi động, lập tức nạp hiển thị tree ra màn hình

### Bài 3

Viết chương trình xây dựng và quản lý danh sách sinh viên dựa trên cây nhị phân tìm kiếm (Binary Search Tree). Mỗi sinh viên chứa các thông tin: mã sv (char), tên sv (char), điểm toán, điểm lý, điểm hóa. Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Lập chỉ mục Index cho cơ sở dữ liệu theo mã sinh viên. Không có 2 sinh viên nào trùng mã với nhau.

- Thêm một SV mới
- Xuất danh sách SV tăng dần theo mã SV
- Tìm 1 SV theo mã. Nếu tìm ra hiển thị menu con chứa các mục
  - Hiển thị thông tin sinh viên: tên, các điểm và điểm trung bình
  - Cập nhật (sửa) thông tin SV (tên, điểm toán, điểm lý)
- Lưu danh sách sinh viên xuống file
- Đọc danh sách sinh viên từ file
- Xuất danh sách sinh viên tăng dần theo tên SV.

## Module 9

### Bài 1

Viết chương trình xây dựng và quản lý cây nhị phân tìm kiếm (Binary Search Tree). Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

- Thêm một node vào cây (giá trị nhập vào).
- Lưu cây vào file
- Đọc cây từ file
- Xuất danh sách
  - Inorder
  - Preorder
- Tìm một node trên cây
- Hủy một node trên cây. (chọn node trái cùng nhánh bên phải)
- Hủy một node trên cây. (chọn node phải cùng nhánh bên phải)

Lưu ý: sinh viên vẽ trên giấy cây nhị phân, đồng thời với việc thực thi chương trình cho các thao tác xóa-thêm node và kiểm tra đối chiếu kết quả chương trình với trên giấy.

### Bài 2

Sử dụng bài tập ở câu trên tiếp tục phát triển rộng các menu như sau. (Chú ý: các chức năng thực hiện bằng hàm, không sử dụng biến toàn cục. Sử dụng kỹ thuật đệ quy):

- Đếm số node của cây.
- Đếm số Inode lá của cây
- Đếm số node có đầy đủ 2 con
- Đếm số node chỉ có 1 con
- Đếm số node có giá trị chẵn
- Đếm số node có giá trị lẻ
- Tính tổng giá trị các node
- Tìm giá trị trung bình của danh sách
- Tìm chiều cao của cây
- Tính giá trị trung bình của các node
- Tìm giá trị nhỏ nhất
- Tìm giá trị lớn nhất
- Tìm một node theo giá trị nhập vào
- Tìm cấp (level) của một node theo giá trị nhập vào
- Hủy toàn bộ cây.
- (\*) Xuất ra đường đi từ root đến 1 node bất kỳ (giá trị nhập vào)
- (\*) Tìm đường đi giữa 2 node bất kỳ (với 2 giá trị nhập vào)
- (\*) Kiểm tra 2 node bất kỳ có quan hệ tổ tiên hay không? (với 2 giá trị nhập vào)
- (\*) Xuất ra đường đi giữa 2 node bất kỳ (với 2 giá trị nhập vào)

## Module 10

### Bài 1

Viết chương trình quản lý danh sách lớp. Mỗi sinh viên gồm các thành phần:

- +Mã SV: char[10];
- +Mã Lớp : int
- +Tên SV: char[255];
- +DiemToan
- +DiemLy
- +DiemHoa

Mỗi lớp chứa gồm các thông tin:

- +Mã Lớp: int
- +Tên Lớp: char[10];
- +Khóa

Thành phần khóa chính (và Index) của danh sách sinh viên chính là mã SV. Thành phần khóa chính (và Index) của danh sách lớp chính là mã lớp.

Xây dựng và quản lý danh sách lớp sử dụng cây nhị phân tìm kiếm (Binary Search Tree). Hiển thị menu thực hiện các chức năng sau (mỗi chức năng thực hiện bằng hàm). Thành phần dữ liệu trong mỗi Node là giá trị kiểu integer.

1. Thêm 1 lớp mới.
2. Thêm một sinh viên
  - a. Nếu mã SV đã có thì hiển thị sinh viên đó ra màn hình, cùng với thông báo không thể nhập sinh viên đã có
  - b. Mã lớp phải tồn tại trong danh sách lớp. Nếu chưa có, phải hiện thông báo lỗi.
3. Tìm một sinh viên theo mã SV
  - a. Khi tìm thấy, hiển thị mã, tên, điểm, mã lớp và tên lớp.
4. Lưu danh sách sinh viên-lớp vào file
5. Đọc danh sách sinh viên từ file.
6. Hiển thị danh sách sinh viên
  - a. Tăng dần theo mã SV
  - b. Giảm dần theo mã SV
  - c. Mỗi sinh viên hiển thị điểm toán, lý, hóa và điểm trung bình
7. Tìm tất cả sinh viên theo tên nhập vào
8. Hiển thị tất cả sinh viên theo mã lớp nhập vào
9. Hiển thị tất cả sinh viên theo tên lớp nhập vào
10. Xóa một sinh viên ra khỏi danh sách
11. Xóa một lớp ra khỏi danh sách
12. Tìm tất cả sinh viên có điểm trung bình lớn nhất
13. Tìm tất cả sinh viên có điểm trung bình lớn nhất trong một lớp

Chú ý: Sinh viên sử dụng 2 cây nhị phân, một cho danh sách lớp, một cho danh sách sinh viên.

Để đơn giản, có thể lưu thành 2 file riêng cho danh sách lớp và danh sách sinh viên.

Lưu ý: Mỗi một thao tác thêm sinh viên, xóa sinh viên: chương trình tự động lưu vào file. Lần thực thi kế tiếp, chương trình tự động nạp từ file vào bộ nhớ.

## Module 11

### Bài 1

Viết chương trình quản lý danh sách lớp. Mỗi sinh viên gồm các thành phần:

+Mã SV, Tên SV: char[255], DiemTB

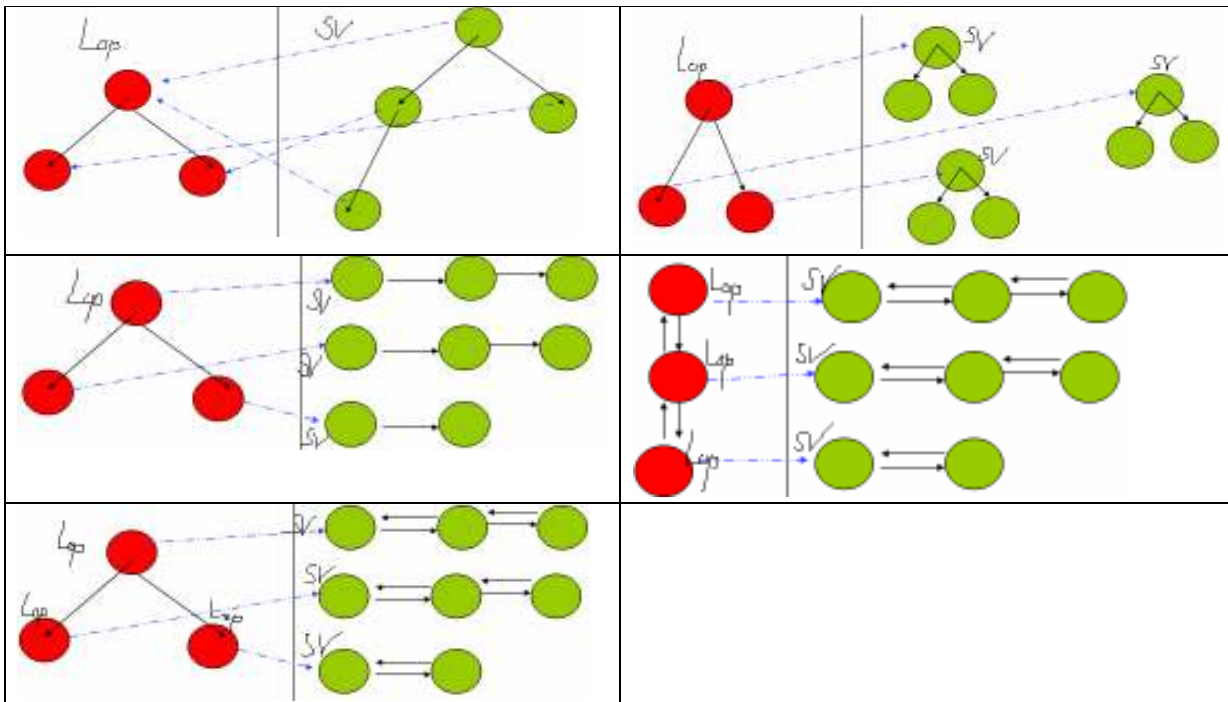
Mỗi lớp gồm các thông tin:

+Mã Lớp: int

+Tên Lớp: char[10];

Mỗi lớp có nhiều Sinh viên.

Sử dụng các cấu trúc dữ liệu sau. Mỗi trường hợp thực hiện 3 chức năng: thêm lớp, thêm sinh viên, tìm tất cả sinh viên của 1 lớp.



### Bài 2

Viết chương trình quản lý danh sách mua vé máy bay – hành khách. Mỗi khách chỉ mua 1 vé. Mỗi vé máy bay gồm các thành phần: Vé máy bay (ID, giá)

Mỗi hành khách gồm các thông tin: Khách (PassID, tên)

Sử dụng các cấu trúc dữ liệu sau. Thực hiện menu với 3 chức năng: thêm vé máy bay, thêm hành khách, bán 1 vé máy bay (vé chưa bán) cho 1 hành khách (chưa mua vé). Khi bán vé, người sử dụng nhập ID của vé và PassID của hành khách.

