

## Bài tập 1

1. Xây dựng lớp Stack
2. Viết chương trình cho phép thực hiện các chức năng sau:
  - Nhập vào một biểu thức dạng trung tố
  - Chuyển biểu thức đó sang dạng hậu tố
  - Tính giá trị của biểu thức

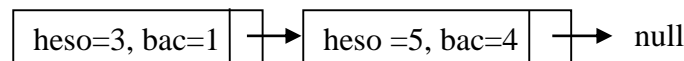
## Bài tập 2

1. Sử dụng lớp danh sách liên kết đơn xây dựng lớp đa thức, với các phương thức sau:

- Toán tử nhập, xuất
- Tính giá trị của đa thức
- Tính tổng, hiệu, tích hai đa thức

Lưu ý: Mỗi phần tử của danh sách lưu hệ số của đa thức và bậc của phần tử đó, các hệ số = 0 không được lưu trong danh sách

Ví dụ: nếu  $p(x) = 3x + 5x^4$  thì danh sách là



2. Xây dựng hàm main có các chức năng
  - Nhập vào hai đa thức từ bàn phím hoặc từ file
  - In lên màn hình các đa thức, tổng, hiệu tích của hai đa thức đó.
  - Ghi đa thức tổng, hiệu, tích, thương của hai đa thức ra file

## Bài tập 3

Ứng dụng lớp Vector để quản lý các đối tượng là các sinh viên. Mỗi sinh viên cần quản lý các thông tin sau: Mã SV, Họ tên, Ngày sinh, giới tính, lớp. Trong lớp sinh viên xây dựng toán tử so sánh theo họ tên.

1. Bổ sung vào lớp Vector phương thức sắp xếp theo thuật toán Heapsort, để sắp xếp các phần tử theo thứ tự một hàm so sánh được vào cho hàm HeapSort
2. Bổ sung vào lớp Vector phương thức **tìm kiếm** theo phương pháp tìm kiếm nhị phân trên mảng với hàm so sánh là một đối của phương thức tìm kiếm.
3. Viết chương trình cho phép thực hiện các chức năng sau:
  - Nhập thêm các danh sách sinh viên, sau khi hoàn thành nhập thông tin của một sinh viên, chương trình đưa ra câu hỏi có nhập nữa không (c/k)? Nếu người dùng nhập: c thì tiếp tục nhập, nhập k thì kết thúc.
  - Xóa đi một sinh viên
  - Sửa đổi thông tin của một sinh viên bất kỳ trong danh sách
  - Sắp xếp danh sách sinh viên theo họ tên

- Hiển thị toàn bộ danh sách sinh viên hiện có trong Vector
- Tìm kiếm sinh viên theo họ tên

## Bài tập 4

1. Ứng dụng lớp danh sách liên kết kép xây dựng lớp để quản lý các đối tượng là các sinh viên có các chức năng

- Đọc danh sách sinh viên từ file vào danh sách
- Ghi danh sách sinh viên vào file
- Nhập thêm một sinh viên vào cuối danh sách
- Xóa đi một sinh viên
- Hiển thị toàn bộ danh sách sinh viên hiện có trong danh sách
- Sửa đổi thông tin của một sinh viên bất kỳ trong danh sách
- Tìm kiếm sinh viên theo *họ và tên* bằng phương pháp tìm kiếm tuần tự

Mỗi sinh viên cần quản lý các thông tin sau: Mã SV, Họ và tên, Ngày sinh, giới tính, lớp.

2. Viết hàm main thực hiện các chức năng trên

## Bài tập 5

1. Xây dựng lớp chuỗi ký tự có các phương thức:

- Cắt các ký trắng (dấu cách) ở hai đầu của chuỗi
- Cắt toàn bộ ký tự trắng thừa có trong chuỗi (giữa hai từ mà có  $n$  ( $n > 2$ ) ký tự trắng thì số ký tự trắng thừa là  $n-1$ ).
- Chuyển đổi toàn bộ chuỗi ký tự thành chuỗi ký tự hoa
- Chuyển đổi toàn bộ chuỗi ký tự thành chuỗi ký tự thường
- Thực hiện nối thêm một chuỗi ký tự vào sau chuỗi ký tự hiện có
- Phương thức trích ra  $n$  ký tự ở phía phải của chuỗi
- Phương thức trích ra  $n$  ký tự ở phía trái của chuỗi
- Phương thức trích ra  $n$  ký tự kể từ vị trí thứ  $i$  của chuỗi
- Lấy địa chỉ của thuộc tính lưu trữ chuỗi ký tự
- Nhập
- In

2. Viết chương trình kiểm tra sự hoạt động của lớp đã xây dựng

## Bài 6

1. Xây dựng lớp cây tìm kiếm nhị phân
2. Viết chương trình quản lý một cây mà tại mỗi nút của nó lưu một từ tiếng Anh và nghĩa tiếng việt của nó có các chức năng:
  - Đọc cây từ file

- Ghi cây vào file (*duyet theo thứ tự giữa, duyệt đến nút nào thì ghi vào file giá trị của nút đó*)
- Thêm nút vào cây
- Xóa bỏ nút của cây
- Thay đổi giá trị của một nút
- Tìm kiếm một từ trên cây

## Bài 7

1. Xây dựng lớp biểu diễn một bảng băm sử dụng một trong những hàm băm đã học
2. Sử dụng lớp bảng băm để xây dựng lớp biểu diễn từ điển Anh Việt. Mỗi phần tử của từ điển là một cặp (tiếng anh, nghĩa tiếng việt).
  - Với các phương thức:
    - Nạp từ điển từ file vào bảng băm
    - Tìm kiếm bằng phương pháp tìm kiếm trên bảng băm
    - Sửa đổi các từ
    - Thêm từ mới
    - Xóa từ
    - Lưu từ điển vào file
3. Xây dựng chương trình có các chức năng
  - Nạp từ điển từ file (có thể nạp tự động khi chạy chương trình)
  - Tra từ điển (nhập vào một từ , hiển thị nghĩa của nó nếu có trong từ điển)
  - Sửa đổi từ
  - Thêm từ mới
  - Xóa từ
  - Lưu từ điển vào file

## Bài 8

Viết chương trình thực hiện các công việc sau trên một danh sách các số (*sử dụng cấu trúc dữ liệu danh sách liên kết đơn để lưu trữ*).

1. Tạo danh sách số, quá trình nhập sẽ dừng lại khi nhập dấu “#”.
2. Thêm một phần tử vào danh sách, vị trí thêm vào do người dùng chọn.
3. Nhập vào một số k ( $k \neq 0$ ), đếm xem trong dãy có bao nhiêu số có giá trị = k?
4. Kiểm tra xem trong danh sách có 3 số chẵn dương đứng cạnh nhau hay không? Nếu có hãy in ra vị trí của các số này.
5. Sắp xếp danh sách theo thứ tự tăng dần.
6. Xóa tất cả các số nguyên tố trong danh sách.
7. Xóa tất cả các phần tử có giá trị trùng nhau trong danh sách, chỉ giữ lại 1 số.

## Bài 9

Viết chương trình quản lý danh sách sinh viên bằng **danh sách liên kết kép**, mỗi sinh viên là một cấu trúc gồm:

- Mã số sinh viên
- Họ đệm, tên sinh viên
- Điểm trung bình
- Tên lớp

Chương trình gồm các chức năng sau:

- Tạo danh sách sinh viên: Quá trình nhập danh sách sẽ dừng lại khi nhập mã sv  $\leq 0$ .
- Thêm 1 sinh viên vào danh sách, vị trí sinh viên thêm vào do người dùng chọn.
- Liệt kê danh sách sinh viên trên màn hình theo thứ tự giảm dần của tổng điểm.
- Căn cứ vào ĐTB của sinh viên, liệt kê tất cả sinh viên đạt loại giỏi (có ĐTB  $\geq 8.0$ )
- Nhập vào tên lớp, in ra màn hình danh sách sinh viên của lớp này.
- Nhập vào mã số sinh viên, sau đó in ra vị trí của sinh viên trong danh sách.
- Sắp xếp danh sách sinh viên theo thứ tự tăng dần của tên
- Xoá toàn bộ danh sách những sinh viên có ĐTB  $< 5.0$

## Bài 10

Viết chương trình quản lý nhân sự bằng danh sách liên kết đơn, mỗi cán bộ là một cấu trúc gồm:

- Mã cán bộ
- Họ đệm, tên cán bộ
- Phòng ban
- Chức vụ
- Hệ số lương

Chương trình gồm các chức năng sau:

- Tạo danh sách cán bộ: Quá trình nhập danh sách sẽ dừng lại khi nhập mã cán bộ  $\leq 0$ .
- Thêm 1 cán bộ vào danh sách, vị trí thêm vào do người dùng chọn.
- Tính lương cho nhân viên, biết rằng:  $\text{Lương} = \text{Hệ số lương} * 1350000$
- In lên màn hình tất cả cán bộ có hệ số lương  $\geq 4.4$
- Tìm và in danh sách cán bộ theo Chức vụ
- Tìm và in danh sách cán bộ theo hệ số lương và phòng ban (nghĩa là nhập vào hệ số lương, tên phòng ban cần tìm, sau đó in danh sách những cán bộ thỏa mãn cả 2 điều kiện này).
- Sắp xếp danh sách cán bộ theo thứ tự của tên.

## Bài 11

Để kiểm tra lỗi chính tả trong các văn bản, người ta thực hiện kiểm tra từng từ của văn bản có trong từ điển của ngôn ngữ đó hay không?. Việc kiểm tra lỗi chính tả được sử dụng trong nhiều ứng dụng như các phần mềm soạn thảo văn bản.

Hãy viết một chương trình kiểm tra lỗi chính tả văn bản tiếng anh theo các yêu cầu sau:

- Nạp một từ điển ngôn ngữ tiếng Anh vào một cấu trúc thích hợp từ file
- Viết chương trình nhập vào một đoạn văn, thực hiện kiểm tra xem trong đoạn văn đó có từ nào sai lỗi chính tả hay không?
- Liệt kê các từ sai lỗi chính tả, người dùng kiểm tra những từ mà phân báo sai chính tả nhưng nó không sai thì thêm nó vào từ điển
- Ghi từ điển vào file nếu có bổ sung từ mới.

## Bài 12

1. Xây dựng lớp biểu diễn đồ thị đồ thị (Graph) vô hướng có trọng số bằng danh sách kề. Mỗi đỉnh của đồ thị là một số tự nhiên có giá trị duy nhất trong đồ thị. Mỗi cung của đồ thị được xác lập bằng một cặp đỉnh và một trọng số là số thực.
2. Cài đặt các thao tác cơ bản trên đồ thị như sau
  - Khởi tạo một đồ thị ban đầu rỗng có  $n$  đỉnh
  - **InsertEdge**(int s, int d, float weight): bổ sung một cung có trọng số  $w$  giữa hai đỉnh  $s, d$  ( $s, d \leq n$ )
  - **InsertVertex**(name): thêm một đỉnh mới vào đồ thị
  - **GetWeight**(int s, int d): lấy ra trọng số của một cạnh
  - **Print**: hiển thị hiện trạng của đồ thị
  - Cài đặt thuật toán tìm cây khung nhỏ nhất **Prim** hoặc **Kruskal**
3. Xây dựng một chương trình nhận đầu vào là một file text chứa thông tin về một đồ thị liên thông cho trước, tạo lập đồ thị, hiển thị đồ thị và cây khung nhỏ nhất của đồ thị đó lên màn hình.

## Bài 13

Một công ty điện thoại cần xây dựng một chương trình tính tiền điện thoại cho các khách hàng. Hiện tại, công ty có lưu trữ 2 tệp tin, một tệp về khách hàng, một tệp về các cuộc điện thoại đã gọi của khách hàng.

1. Tệp khách hàng có tên *khachhang.txt*. Mỗi dòng của tệp này gồm có:

*Tên khách hàng; số điện thoại*

- Tên của khách hàng: tên là một xâu ký tự, độ dài không vượt quá 25 và kết thúc bởi dấu “;”.

- Số điện thoại của khách hàng gồm 10 chữ số bắt đầu bằng chữ số 0

Ví dụ:

Nguyen Anh Tuấn; 0987654233

Le Nhat Anh; 0967456321

...

2. Tập lưu trữ các cuộc điện thoại có tên **cuocgoi.txt**. Mỗi dòng có những thông tin về một cuộc điện thoại như sau:

*Số điện thoại ; Số phút; Thời điểm bắt đầu gọi; Ngày gọi (dd/mm/yyyy); Vùng (Vùng: nội hạt, lân cận, xa và rất xa, được viết tắt là: NH, LC, X, RX).*

Ví dụ:

0987654233; 4; 8h23; 14/010/2018; NH

0987654233; 6; 15h10; 19/10/2018; LC

0967456321; 1; 23h05; 15/10/2018; RX

....

Hãy sử dụng cấu trúc dữ liệu thích hợp viết chương trình tính tiền điện thoại theo yêu cầu sau:

- Đọc thông tin trong tệp *khachhang.txt* và **cuocgoi.txt**, tính tiền cho từng khách hàng và ghi ra tệp *ketqua.txt*, mỗi dòng có các thông tin sau:

*Tên KH; số dt; Số tiền, số cuộc gọi NH, số gọi LC, số cuộc X, số cuộc gọi RX.*

**Cách tính tiền cho mỗi cuộc gọi như sau:**

*Tiền = Giá cơ bản \* Số phút \* Hệ số miễn.*

*Giá cơ bản là 1.100 đồng 1 phút.*

*Hệ số miễn đối với nội hạt là 1, với lân cận là 2, với xa là 3, với rất xa là 4. Đối với các cuộc gọi bắt đầu từ 23h00 đến 5h00 các ngày trong tuần và ngày Thứ Bảy, Chủ nhật thì được giảm giá 30%.*

Lưu ý: trong danh sách cuộc gọi mỗi khách có thể có nhiều cuộc gọi hoặc không có cuộc gọi nào.

## Bài 14.

1. Xây dựng lớp biểu diễn một bảng băm sử dụng một trong những hàm băm đã học

2. Sử dụng lớp bảng băm để xây dựng lớp quản lý một thư viện giúp người quản lý thư viện có thể tra cứu nhanh một loại sách nào đó nằm ở đâu trong thư viện.

Với các phương thức:

- Nạp thông tin của thư viện từ file vào bảng băm
- Tìm kiếm bằng phương pháp tìm kiếm trên bảng băm
- Thêm sách mới vào thư viện
- Xóa sách
- Lưu thông tin thư viện vào file

**Lưu ý:** mỗi loại sách gồm có *mã sách, tên sách, mô tả vị trí trong thư viện*

## Bài 15

1. Xây dựng lớp biểu diễn đồ thị đồ thị (Graph) vô hướng có trọng số bằng danh sách kề. Mỗi đỉnh của đồ thị là một số tự nhiên có giá trị duy nhất trong đồ thị. Mỗi cung của đồ thị được xác lập bằng một cặp đỉnh và một trọng số là số thực.
2. Cài đặt các thao tác cơ bản trên đồ thị như sau
  - Khởi tạo một đồ thị ban đầu rỗng có  $n$  đỉnh
  - **InsertEdge**(int s, int d, float weight): bổ sung một cung có trọng số  $w$  giữa hai đỉnh  $u, v$  ( $u, v \leq n$ )
  - **InsertVertex**(name): thêm một đỉnh mới vào đồ thị (đỉnh  $n+1, n+2, \dots$ )
  - **GetWeight**(int s, int d): lấy ra trọng số của một cạnh
  - **Print**: hiển thị hiện trạng của đồ thị
  - Cài đặt thuật toán **Dijkstra** tìm đường đi ngắn nhất trong đồ thị
3. Xây dựng một chương trình nhận đầu vào là một file text chứa thông tin về một đồ thị cho trước, tạo lập đồ thị, hiển thị đồ thị lên và từng bước của quá trình tìm đường đi ngắn nhất từ đỉnh  $(u, v)$  được người dùng nhập từ bàn phím.

## Bài 16

1. Xây dựng lớp số lớn bằng một trong các cấu trúc dữ liệu véc tơ, danh sách liên kết đơn, danh sách liên kết kép. Mỗi phần tử của danh sách liên kết lưu một chữ số thập phân của số lớn. Ví dụ số 4728

4	7	2	8
---	---	---	---

Có các phương thức nhân một số lớn với một số kiểu int  
Cộng hai số lớn với nhau

2. Viết hàm main thực hiện
  - a. Tính số Fibonacci thứ  $n$  với  $n$  lớn ( $1 \leq n \leq 1000$ )
  - b. Tính giai thừa của số  $n$  lớn ( $1 \leq n \leq 1000$ )

## Bài 17

1. Xây dựng cấu trúc dữ liệu hàng đợi
2. Áp dụng cấu trúc dữ liệu hàng đợi trong thuật toán tìm kiếm theo chiều rộng (BFS)
  - a. Tìm kiếm đường đi trong đồ thị có hướng
  - b. Thuật toán loang tìm đường đi trong mê cung
  - c. Một bài toán cụ thể khác sinh viên tự đề xuất
3. Viết hàm main thực hiện các công việc trên

### **Bài 18**

1. Xây dựng cấu trúc dữ liệu hàng đợi
2. Áp dụng cấu trúc dữ liệu hàng đợi trong thuật toán tìm kiếm theo chiều sâu (DFS)
  - a. Tìm kiếm đường đi trong đồ thị có hướng
  - b. Thuật toán tìm đường đi trong mê cung
  - c. Một bài toán cụ thể khác sinh viên tự đề xuất
3. Viết hàm main thực hiện các công việc trên

### **Bài 19**

1. Tìm hiểu về cây tiền tố (TRIE)
2. Áp dụng cài đặt cho các bài toán sau:
  - a. Cho n xâu kí tự hỏi có tồn tại xâu nào là tiền tố của xâu nào không?
  - b. Cho n số điện thoại gồm các chữ số từ 0 đến 9 hỏi có số nào là tiền tố của các số còn lại không
  - c. Một ứng dụng khác của cây tiền tố do các bạn sinh viên tự đề xuất



# HƯỚNG DẪN LÀM BÀI TẬP LỚN

**Lớp trưởng thu bài làm của các nhóm cho thầy gồm**

1. Một quyển báo cáo
2. Bản báo cáo file doc và chương trình **nén** vào một file .rar, lớp trưởng tổng hợp tất cả các file này lại đẩy lên Google Driver sau đó share **Link** cho thầy.

**Chú ý:**

3. Tên file ghi theo cấu trúc sau: Nhóm – XX – Bai YY

***Nội dung quyển báo cáo gồm các phần sau:***

**I.** Đề bài

**II.** Phân tích bài toán

Xác định các yêu cầu của bài toán, xác định các lớp, các thuộc tính, các phương thức của lớp.

Mô tả chức năng của từng lớp, từng phương thức.

**III.** Cài đặt các lớp và hàm main bằng C++

**IV.** Phân tích thời gian chạy của từng phương thức có trong các lớp

**V.** Danh sách tài liệu tham khảo

Mẫu bìa (trang dưới)

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  
KHOA CÔNG NGHỆ THÔNG TIN

-----o0o-----



**Bài tập lớn môn học**

**CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

Giảng viên hướng dẫn: TS. Hoàng Văn Thông

Nhóm thực hiện: **XX** - Lớp CNTT **YY**-K58

Danh sách sinh viên tham gia xếp theo thứ tự đóng góp vào báo cáo

STT	Mã SV	Họ tên
1		
2		
3		

**Hà Nội tháng 11 năm 2018**