

DANH MỤC CÁC BÀI LỚN

Bài 1

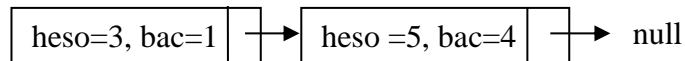
1. Xây dựng lớp Stack
2. Xây dựng lớp tính giá trị biểu thức
3. Viết chương trình cho phép thực hiện các chức năng sau:
 - Nhập vào một biểu thức dạng trung tố
 - Chuyển biểu thức đó sang dạng hậu tố
 - Tính giá trị của biểu thức

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

Bài 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
 - Hiện 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 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
 - 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 từ điển Anh-Việt, từ điển được lưu vào cây nhị phân mà tại mỗi nút của nó lưu một từ tiếng Anh và nghĩa tiếng Việt tương ứng có các chức năng:
 - Đọc từ điển từ file để xây dựng cây
 - Ghi từ điển trong cây vào file (*duyet theo thứ tự giữa, duyệt đến nút nào thì ghi vào file giá trị của nút đó*)
 - Thêm một từ mới từ điển đang lưu trong cây
 - Xóa bỏ một từ của từ điển đang lưu trong cây
 - Cập nhật lại một từ đang lưu trong cây
 - Tìm kiếm một từ trong 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 ≤ 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 ≥ 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
- Xóa 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 ≤ 0 .
- Thêm 1 cán bộ vào danh sách, vị trí thêm vào do người dùng chọn.
- Tính lương cho nhân viên, biết rằng: $Lương = Hệ\ số\ lương * 1350000$
- In lên màn hình tất cả cán bộ có hệ số lương ≥ 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 phần 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 chuỗi 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 miền.*

Giá cơ bản là 1.100 đồng 1 phút.

Hệ số miền 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
 - Trừ hai số lớn
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. Xây dựng cấu trúc dữ liệu biểu diễn đồ thị
3. Áp dụng cấu trúc dữ liệu hàng đợi thiết kế 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
4. Viết hàm main thực hiện các công việc trên

Bài 18

1. Xây dựng cấu trúc dữ liệu ngăn xếp
2. Xây dựng lớp biểu diễn đồ thị vô hướng có trọng số bằng ma trận kề có các phương thức:
 - a. Nhập đồ thị từ file
 - b. Ghi đồ thị ra file
 - c. Duyệt đồ thị theo chiều sâu (DFS)
 - d. Tìm đường đi ngắn nhất giữa 2 đỉnh bất kỳ
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

Bài 20

1. Ứng dụng lớp Vector xây dựng lớp để quản lý một cửa hàng bán tạp hóa, thông tin quản lý của cửa hàng gồm:
 - + **Danh mục mặt hàng** gồm các thông tin: Mã hàng, Tên hàng, nhà sản xuất
 - + **Hóa đơn mua bán hàng:** Mã hóa đơn, Mã hàng, Loại hóa đơn (Bán hoặc Mua), Số lượng, Ngày mua bán, Giá mua bán.
 - Đọc danh thông tin quản lý của cửa hàng từ file vào chương trình
 - Ghi thông tin quản lý của cửa hàng vào file
 - Nhập xuất các mặt hàng
 - Hiện thị toàn bộ danh sách mặt hàng hiện có
 - Thống kê số lượng mặt hàng tồn của mỗi mặt hàng
2. Viết hàm main thực hiện các chức năng trên

Bài 21

1. Ứng dụng danh sách liên kết đơn để giải quyết bài toán đồ thị
 - Xây lớp đồ thị (Graph) vô hướng có trọng số, đồ thị được mô tả bằng danh sách kề, có các phương thức:
 - + Đọc đồ thị từ file
 - + Ghi đồ thị ra file

- + Xây dựng các đồ thị con của nó, mỗi đồ thị con là một thành phần liên thông, Phương thức trả lại danh sách đồ thị con của đồ thị
 - + Duyệt đồ thị theo chiều rộng (BFS) nếu đồ thị liên thông
 - + Duyệt đồ thị theo chiều sâu (DFS) nếu đồ thị liên thông
 - + Tìm đường đi ngắn nhất giữa 2 đỉnh bất kỳ nếu đồ thị liên thông
2. Viết hàm main, thực hiện các chức năng của lớp Graph

Bài 22

1. Cho cây nhị phân trừu tượng chứa các phần tử so sánh được với nhau thỏa mãn các tính chất sau:

- a. Phần tử gốc lớn hơn hoặc bằng mọi phần tử còn lại của cây
- b. 2 cây con trái và phải cũng thỏa mãn tính chất trên

2. Hãy cài đặt cấu trúc dữ liệu cây nhị phân trên có các thao tác push(x) thêm phần tử vào cây và pop() lấy phần tử gốc khỏi cây, top() lấy giá trị của nút gốc sao cho luôn luôn đảm bảo tính chất cây heap

Áp dụng cấu trúc cây này như một hàng đợi ưu tiên thực hiện các việc sau

- Nhập vào 1 dãy số xuất ra dãy giảm dần
- Giải quyết bài toán nối thanh kim loại <http://laptrinhonline.club/problem/Tèonoidam>

Công việc cơ khí thật là mệt nhọc, muốn nối một thanh kim loại độ dài a với một thanh kim loại độ dài b thì kinh phí để thuê nối tốn mất a+b đơn vị tiền tệ. Hiện nay Tèo cần nối n thanh kim loại lần lượt có độ dài là a_1, a_2, \dots, a_n thành một đoạn theo bạn Tèo nên bố trí thế nào để tổng số tiền phải trả là ít nhất

Input:

Dòng đầu chứa số nguyên dương n ($1 \leq n \leq 10^5$)

Dòng tiếp theo là n số nguyên dương tương ứng là độ dài các thanh muốn nối ($1 \leq a_i \leq 10^3$)

Output:

Một số nguyên dương là số kinh phí ít nhất phải trả

Ví dụ

Input

3

8 4 6

Output

28

Giải thích: Nếu ta nối thanh 8 với thanh 4 tốn chi phí là $8+4=12$ sau khi nối xong còn 2 thanh độ dài 12 và 6 nối lại với nhau tốn $12+6=18$ tổng chi phí nối là $12+18=30$. Nếu ta nối 4 với 6 trước tốn 10 và còn 2 thanh 10 và 8 nối lại với nhau tốn 18 do đó tổng kinh phí ít hơn chỉ còn 28

Gợi ý: Bằng cách nhập vào dãy a là độ dài thanh mỗi bước lấy ra 2 thanh ngắn nhất nối với nhau rồi lại thêm thanh nối được vào

Bài 23

Tạo một cây từ điển để giải bài toán xây dựng danh bạ sau:

<http://laptrinhonline.club/problem/Tèodanhba>

Tèo muốn xây dựng danh bạ liên lạc để lưu vào trong máy tính. Trong quá trình xây dựng danh bạ Tèo để ý và thấy rằng rất nhiều từ nọ là tiền tố của từ kia. Nếu từ x được ghép bởi hai từ y và z tức là $x=yz$ thì y được gọi là tiền tố và z gọi là hậu tố của x (Ví dụ từ tichpx có các tiền tố t, ti, tic, tich, tichp, tichpx). Trong khi cập nhật danh bạ với mỗi từ Tèo có 1 trong hai thao tác

- add là bổ sung một liên lạc vào danh bạ
- find là tìm xem một từ là tiền tố của bao nhiêu liên lạc đã có

Bạn hãy lập trình để xác định giúp Tèo nhé

Input:

- Dòng đầu chứa số nguyên dương n là số thao tác ($1 \leq n \leq 105$) ($1 \leq n \leq 105$)
- n dòng tiếp theo mỗi dòng thuộc một trong hai thao tác hoặc là add hoặc là find một liên lạc là một xâu gồm toàn chữ thường tiếng anh có độ dài không quá 30 ký tự

Output:

Với mỗi thao tác find xuất ra màn hình số liên lạc đã có trong danh bạ có tiền tố muốn tìm

Ví dụ

Input

```
6
add daihocgiaothongvantai
add daihocthuyloi
find daihoc
add daihocxaydung
find hocvien
find dai
```

Output

```
2
0
3
```

Bài 24

Cài đặt cấu trúc dữ liệu double list và các bộ duyệt xuôi và ngược có các thao tác cơ bản thêm bớt ở 2 đầu, chèn và xóa theo bộ lặp ngoài ra còn có các thao tác bổ sung sau:

- Xóa hết các phần tử xuất hiện từ lần 2 trở lên ra khỏi list để mỗi phần tử chỉ xuất hiện 1 lần
- Sắp xếp danh sách tăng dần
- Chèn thêm 1 phần tử vào danh sách đã sắp tăng để được danh sách sắp tăng dần
- Tác động 1 hàm fun(x) vào từng phần tử x của list để biến đổi list
- Viết hàm main thực hiện các thao tác sau:
 - o Nhập 1 dãy số nguyên vào list
 - o Xóa đi những phần tử trùng nhau
 - o Biến đổi từng phần tử x thành $(x^2-7x+3) \% 100$ sau đó lọc đi những số giống nhau chỉ giữ lại 1
 - o Sắp xếp tăng dần
 - o Nhập x và chèn vào list để dãy vẫn tăng dần
 - o Xuất dãy theo chiều ngược lại để được thứ tự giảm dần

Bài 25

1. Cài đặt cấu trúc dữ liệu single list với bộ lặp xuôi với kiểu dữ liệu trừu tượng
2. Cài đặt class hàng hóa gồm tên hàng, số lượng, đơn giá các thao tác nhập xuất
3. Cài đặt class giỏ hàng với một danh sách liên kết đơn các hàng hóa cho phép thêm một hàng vào giỏ nếu có rồi thì tăng số lượng chưa có thì thêm mới vào cuối danh sách; bớt mặt hàng nếu bớt số lượng bằng 0 thì xóa khỏi giỏ. Tính tiền giỏ hàng bằng tổng tiền của từng mặt hàng. Xuất dữ liệu về giỏ hàng ra file văn bản hoadon.txt

Bài 26

1. Cài đặt cấu trúc dữ liệu trừu tượng vector có bộ lặp xuôi và ngược
2. Cài đặt class câu hỏi trắc nghiệm mỗi câu hỏi gồm câu hỏi dạng văn bản và 4 phương án trong đó có 1 phương án đúng
3. Cài đặt class thi dùng cấu trúc vector đã xây dựng để đọc dữ liệu từ câu hỏi từ file văn bản lưu vào vector. Cho phép tạo ra đề thi gồm k câu hỏi ngẫu nhiên trong vector các câu hỏi, khi hiển thị sáo trộn vị trí các phương án trả lời cho phép người thi và tính điểm

Bài 27

Cây tìm kiếm nhị phân để tạo cấu trúc dữ liệu tập hợp (set)

1. Cài đặt cấu trúc dữ liệu set là một cấu trúc dữ liệu trừu tượng tổ chức trên cây tìm kiếm nhị phân các phần tử lưu trữ không có trùng lặp (nếu đã có không thêm vào cây)
2. Viết hàm main thực hiện thao tác sau
 - a. Nhập dữ liệu các tập số nguyên A từ file A.txt và tập B từ file B.txt, tập C từ file C.txt
 - b. Thêm lần lượt các phần tử của tập B vào tập A để được tập hợp hợp của A và B
 - c. Bớt dần các phần tử của C khỏi A

Bài 28

Cài đặt lớp từ điển dựa trên bảng băm với hàm băm là tổng mã ASCII của các ký tự của khóa chia dư cho 997. Từ điển bao gồm các chức năng:

- Chèn cặp từ tiếng Anh và từ tiếng Việt vào từ điển với khóa là từ tiếng Anh.
- Lấy nghĩa của từ tiếng Việt với đầu vào là một từ tiếng Anh.
- Xóa cặp từ khỏi từ điển.
- Cập nhật một từ đã có trong từ điển.
- Đọc dữ liệu từ một tệp tin (file) để chèn vào từ điển.
- Xuất dữ liệu của từ điển ra một tệp tin (file).

Bài 29

Tìm hiểu về mã hóa Huffman và thuật toán xây dựng cây Huffman.

- Cài đặt thuật toán xây dựng cây Huffman sử dụng hàng đợi ưu tiên.
- Duyệt cây và gán các từ mã cho các ký tự.
- Xây dựng ứng dụng nén chuỗi ký tự bằng thuật toán nén tĩnh và giải nén chuỗi ký tự đã được nén.

Bài 30.

Tìm hiểu về cấu trúc đồng nhị phân.

1. Ứng dụng cấu trúc đồng nhị phân để cài đặt cấu trúc hàng đợi ưu tiên có các chức năng:
 - Thêm một phần tử vào hàng đợi.
 - Lấy giá trị một phần tử.

- Xóa một phần tử khỏi hàng đợi.
 - Lấy số phần tử của hàng đợi.
2. Ứng dụng cấu trúc hàng đợi ưu tiên ở trên để cài đặt thuật toán tìm cây khung nhỏ nhất Prim.

Bài 31

- Xây dựng lớp cây nhị phân.
 - Ứng dụng cây nhị phân giải các bài toán sau:
1. Cho một cây nhị phân. Hãy liệt kê tất cả các đường đi từ gốc đến lá.
 2. Cho một cây nhị phân chỉ gồm các số từ 0 đến 9, mỗi đường đi từ gốc đến lá có thể biểu diễn một số. Ví dụ: đường đi từ gốc đến lá là 1->2->3 biểu diễn số 123. Tìm tổng cộng tất cả các số từ gốc đến lá.

Ví dụ:

Đầu vào: [1,2,3]

```

1
 / \
2   3

```

Đầu ra: 12 + 13 = 25

3. Cho một cây nhị phân và một số a, xác định xem có đường đi từ gốc đến lá nào mà tổng giá trị của nó bằng a.

Đầu vào: Với cây nhị phân bên dưới và $a = 22$

```

      5
     / \
    4   8
   / \ / \
  11 13 4
 /  \  \
2    7  1

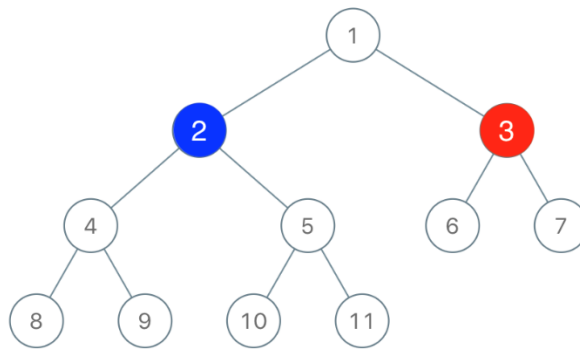
```

Đầu ra: True, tồn tại đường đi 5->4->11->2 mà tổng là 22

Bài 32

- Xây dựng lớp cây nhị phân.
 - Ứng dụng cây nhị phân cho bài toán sau:
- Hai đấu thủ chơi trò chơi theo lượt trên một cây nhị phân. Cho root là gốc của cây nhị phân, số nút trên cây là n với n là số lẻ, mỗi nút có giá trị khác nhau từ 1 đến n. Ban đầu, đấu thủ đầu tiên đưa ra một giá trị x với $1 \leq x \leq n$ và đấu thủ thứ hai đưa ra một giá trị y với $1 \leq y \leq n$ và $y \neq x$. Đấu thủ đầu tiên tô màu nút x màu đỏ và đấu thủ thứ hai tô màu nút y màu xanh. Sau đó, các đấu thủ lần lượt thay nhau chơi. Trong mỗi lượt, cầu thủ chọn một nút với màu của họ (đỏ nếu là cầu thủ 1, xanh nếu là cầu thủ 2) và tô màu nút lân cận chưa được tô của nút được chọn (hoặc là con trái, con phải hoặc cha cha của nút được chọn).
- Nếu (và chỉ nếu) một đấu thủ không thể chọn nút theo cách như vậy, họ bị mất lượt. Nếu cả hai đấu thủ bị mất lượt, trò chơi kết thúc và người thắng là đấu thủ tô được nhiều nút hơn. Bạn là đấu thủ thứ hai. Nếu có thể chọn một giá trị y như vậy để đảm bảo rằng bạn thắng trò chơi, trả lại giá trị *true*. Nếu không thể, trả lại giá trị *false*.

Ví dụ 1:



Đầu vào: root = [1,2,3,4,5,6,7,8,9,10,11], n = 11, x = 3

Đầu ra: true

Diễn giải: Đầu thủ thứ hai có thể chọn nút với giá trị 2 thì sẽ thắng.

Các ràng buộc:

- root là gốc của cây nhị phân với n nút và có các giá trị khác nhau từ 1 đến n.
- n là lẻ.
- $1 \leq x \leq n \leq 100$

Bài 33

- Xây dựng lớp cây tìm kiếm nhị phân.

- Viết chương trình quản lý lịch công tác trong tháng đơn giản: cho phép nhập vào nội dung công việc cần làm theo ngày, theo giờ. Trong một ngày có thể có nhiều công việc, mỗi công việc có giờ bắt đầu, tên công việc, nội dung công việc, tính chất công việc {*rất quan trọng, quan trọng, bình thường, không cần thiết*}, ...

Chương trình có các chức năng chính như sau:

- Nhập nội dung công việc cần làm theo ngày, theo giờ.
- Xem lịch công tác theo ngày yêu cầu.
- Xem các công việc theo tính chất: rất quan trọng, quan trọng, ...
- Xem các công việc đã hoàn tất.
- Xem các công việc chưa thực hiện.
- Xem các công việc từ ngày a đến ngày b.
- Xóa hay điều chỉnh lịch công tác. Nếu sau khi điều chỉnh, ngày nào không còn việc phải làm sẽ xóa khỏi lịch công tác.

Yêu cầu: Cấu trúc dữ liệu được sử dụng là cây nhị phân tìm kiếm (BST), trong đó:

- Mỗi nút trên cây BST là một ngày của lịch công tác.
- Trong mỗi nút ngày trên cây lại chứa một danh sách liên kết lưu thông tin các công việc.
- Khi thêm một công việc vào một ngày đã tồn tại trên cây thì công việc này sẽ được đưa vào danh sách liên kết chứa các công việc theo thứ tự tăng dần của giờ bắt đầu.

Bài 34

- Cài đặt lớp cấu trúc dữ liệu ngăn xếp bằng danh sách liên kết đơn mẫu.

- Ứng dụng giải bài toán sau:

Trong một văn bản HTML thì mỗi thẻ mở phải đi kèm với một thẻ đóng, ví dụ:

```
<div class="header-right">
```

```
<form method="get" id="searchform" action="http://wordpress.com/" >
```

```

<div>< label class ="hidden" for="s"> Tìm kiếm:</label >
<input type="text" value ="" name="s" id="s" />
<input type="submit" id ="searchsubmit" value="Go" /></ div>
</form >
</ div>

```

Hãy viết chương trình đọc vào 1 file .html và kiểm tra xem trong file đó có thẻ nào bị lỗi hay không?

Bài 35

- Cài đặt lớp cấu trúc dữ liệu danh sách liên kết đơn mẫu, bổ sung phương thức sắp xếp các phần tử theo thứ tự tăng dần hoặc dần tùy thuộc vào tham số
- Ứng dụng giải các bài toán sau:
 1. Viết chương trình con nhận vào từ bàn phím một dãy số nguyên, lưu trữ nó trong một danh sách có thứ tự tăng dần không có hai phần tử trùng nhau, theo cách sau: với mỗi phần tử được nhập vào chương trình con phải tìm kiếm xem nó có trong danh sách chưa, nếu chưa có thì xen nó vào danh sách cho đúng thứ tự.
 2. Viết chương trình con trộn hai danh sách liên kết chứa các số nguyên theo thứ tự tăng dần để được một danh sách cũng có thứ tự tăng dần.
 3. Viết chương trình con tách một danh sách chứa các số nguyên thành hai danh sách: một danh sách gồm các số chẵn còn danh sách kia chứa các số lẻ.

Bài 36

- Cài đặt lớp cấu trúc dữ liệu danh sách liên kết kép mẫu, bổ sung phương thức sắp xếp các phần tử theo thứ tự tăng dần hoặc dần tùy thuộc vào tham số.
- Ứng dụng giải bài toán sau:

Một buổi họp mặt đại gia đình nhân dịp cụ già Ted tròn 100 tuổi, người ta muốn sắp xếp con cháu của cụ theo thứ tự từ tuổi cao xuống thấp. Giả sử ta có thông tin về giấy khai sinh của từng người đó. Mỗi giấy khai sinh chỉ viết ba thông tin đơn giản gồm: *Tên người cha*, *Tên người con*, *Tuổi của người cha lúc sinh con*.

Hãy giúp đại gia đình trên tính ra tuổi của từng người con cháu cụ Ted và viết ra danh sách theo thứ tự từ tuổi cao xuống thấp.

Đầu vào:

Dòng đầu ghi số bộ test (không quá 100). Với mỗi bộ test:

- Dòng đầu tiên ghi số X ($0 < X < 100$) là số người con cháu cần sắp xếp.
- Tiếp theo là X dòng, mỗi dòng ghi thông tin về một giấy khai sinh của từng người (thứ tự ngẫu nhiên) gồm 3 thành phần, mỗi thành phần cách nhau một khoảng trống:

o Tên người cha: không quá 20 ký tự và không chứa khoảng trống

o Tên người con: không quá 20 ký tự và không chứa khoảng trống

o Tuổi của người cha khi sinh con: 1 số nguyên dương, không quá 100.

Đầu ra:

- Với mỗi bộ test, in ra màn hình thứ tự bộ test (xem thêm trong bộ test ví dụ), sau đó lần lượt là từng người trong danh sách tuổi từ cao xuống thấp (không tính cụ Ted). Mỗi người viết ra hai thông tin: tên, một khoảng trống rồi đến tuổi của người đó.
- Nếu hai người có cùng tuổi thì xếp theo thứ tự từ điển.

Ví dụ:

Input:

2

1

Ted Bill 25

4

Ray James 40

James Beelzebub 17

Ray Mark 75

Ted Ray 20

Output:

DATASET 1

Bill 75

DATASET 2

Ray 80

James 40

Beelzebub 23

Mark 5

Bài 37

1. Cài đặt cấu trúc dữ liệu trừu tượng Vector. Viết hàm main minh họa các phương thức của vector.

2. Ứng dụng Vector viết chương trình quản lý sinh viên. Mỗi đối tượng sinh viên có các thuộc tính sau: id, tên, giới tính, tuổi, điểm toán, điểm lý, điểm hóa, điểm trung bình và học lực.

Id là mã sinh viên tự động tăng.

Khi khởi tạo, chương trình sẽ đọc file "student.txt" để lấy ra danh sách sinh viên đã được lưu trước đó.

Điểm trung bình là giá trị trung bình của 3 môn toán, lý và hóa.

Học lực được tính như sau:

- Giỏi: nếu điểm trung bình lớn hơn hoặc bằng 8.
- Khá: nếu điểm trung bình nhỏ hơn 8 và lớn hơn hoặc bằng 6.5.
- Trung Bình: nếu điểm trung bình nhỏ hơn 6.5 và lớn hơn hoặc bằng 5.
- Yếu: nếu điểm trung bình nhỏ hơn 5.

Yêu cầu: tạo ra một menu với các chức năng sau:

- 1. Thêm sinh viên.
- 2. Cập nhật thông tin sinh viên bởi ID.
- 3. Xóa sinh viên bởi ID.
- 4. Tìm kiếm sinh viên theo tên.
- 5. Sắp xếp sinh viên theo điểm trung bình (GPA).
- 6. Sắp xếp sinh viên theo tên.
- 7. Hiện thị danh sách sinh viên.
- 8. Ghi danh sách sinh viên vào file "student.txt"

Bài 38

1. Cài đặt cấu trúc dữ liệu trừu tượng danh sách liên kết kép có bộ duyệt xuôi và ngược.
2. Lập chương trình quản lý danh sách học sinh, chương trình có những chức năng sau: (Hồ sơ một học sinh giả sử có: Tên, lớp, số điện thoại, điểm TB)
 - a. Cho phép nhập danh sách học sinh từ bàn phím hay từ file.
 - b. Cho phép in ra danh sách học sinh gồm có tên và xếp loại
 - c. Cho phép in ra danh sách học sinh gồm các thông tin đầy đủ
 - d. Cho phép nhập vào từ bàn phím một tên học sinh và một tên lớp, tìm xem có học sinh có tên nhập vào trong lớp đó không? Nếu có thì in ra số điện thoại của học sinh đó
 - e. Cho phép vào một hồ sơ học sinh mới từ bàn phím, bổ sung học sinh đó vào danh sách học sinh, in ra danh sách mới.
 - f. Cho phép nhập vào từ bàn phím tên một lớp, loại bỏ tất cả các học sinh của lớp đó khỏi danh sách, in ra danh sách mới.
 - g. Có chức năng sắp xếp danh sách học sinh theo thứ tự giảm dần của điểm trung bình.
 - h. Cho phép nhập vào hồ sơ một học sinh mới từ bàn phím, chèn học sinh đó vào danh sách mà không làm thay đổi thứ tự đã sắp xếp, in ra danh sách mới.
 - i. Cho phép lưu trữ lại trên đĩa danh sách học sinh khi đã thay đổi.

Bài 39

1. Cài đặt Stack bằng 2 cách: Mảng và danh sách liên kết
2. Mô phỏng hoạt động của stack bằng cách xây dựng chương trình sau:

Bạn được cho một ngăn xếp rỗng và một số truy vấn với ngăn xếp này. Các truy vấn là những truy vấn cơ bản của ngăn xếp: Đẩy vào, lấy ra, in ra phần tử ở đỉnh, các truy vấn có dạng:

- 1 n : Đẩy số nguyên n vào ngăn xếp
- 2 Loại bỏ phần tử ở đầu ngăn xếp (nếu ngăn xếp rỗng thì thao tác này không có hiệu lực)
- 3 In ra phần tử ở đỉnh ngăn xếp (không lấy ra khỏi ngăn xếp, nếu ngăn xếp rỗng thì in ra Empty!)

Dữ liệu vào

- Dòng đầu chứa số nguyên dương TT là số truy vấn;
- TT dòng tiếp theo, mỗi dòng chứa một truy vấn.

Giới hạn:

- $1 \leq T \leq 10^5; |n| \leq 10^9; 1 \leq T \leq 105; |n| \leq 109$.

Dữ liệu ra

- Ứng với mỗi truy vấn loại 3, in ra kết quả tương ứng trên một dòng.

Ví dụ

Input #1

6

1 15

1 20

2

3

2

3

Output #1

15

Empty!

3. Ứng dụng stack lập trình giải bài toán sau:

Cho dãy ngoặc đúng gồm n dấu mở ngoặc (và n dấu đóng ngoặc). Các dấu ngoặc được đánh số thứ tự từ 1 đến $2n$. Hãy liệt kê chỉ số của các cặp dấu mở ngoặc và đóng ngoặc tương ứng. Dữ liệu vào

- Gồm một dòng duy nhất chứa xâu ký tự biểu diễn dãy ngoặc.

Giới hạn:

- $1 \leq n \leq 10^5$.

Dữ liệu ra

- Gồm n dòng, mỗi dòng gồm hai số u, v là chỉ số của các cặp ngoặc tương ứng nhau, thứ tự liệt kê tăng dần theo chỉ số của dấu ngoặc đóng.

Ví dụ

Input #1

()(())

Output #1

1 2

4 5

6 7

3 8

Bài 40

1. Cài đặt cấu trúc dữ liệu queue bằng 2 cách: Mảng và danh sách liên kết.

2. Mô phỏng hoạt động của hàng đợi bằng cách xây dựng chương trình sau:

Bạn được cho một hàng đợi rỗng và một số truy vấn với hàng đợi này. Các truy vấn là những truy vấn cơ bản của hàng đợi: Đẩy vào cuối hàng đợi, lấy ra ở đầu hàng đợi, in ra phần tử ở đầu hàng đợi, các truy vấn có dạng:

- 1 n : Đẩy số nguyên n vào cuối hàng đợi
- 2: Loại bỏ phần tử ở đầu hàng đợi (nếu hàng đợi rỗng thì thao tác này không có hiệu lực)
- 3: In ra phần tử ở đầu hàng đợi (không lấy ra khỏi hàng đợi, nếu hàng đợi rỗng thì in ra Empty!)

Dữ liệu vào

- Dòng đầu chứa số nguyên dương TT là số truy vấn;
- TT dòng tiếp theo, mỗi dòng chứa một truy vấn.

Giới hạn:

$$- 1 \leq T \leq 10^5; |n| \leq 10^9.$$

Dữ liệu ra

- Ứng với mỗi truy vấn loại 33, in ra kết quả tương ứng trên một dòng.

Ví dụ

Input #1

8
1 5
1 7
1 6
2
3
2
2
3

Output #1

7

Empty!

3. Ứng dụng queue lập trình giải bài toán sau:

Cho một dãy gồm N số nguyên: a_1, a_2, \dots, a_n và dãy B rỗng. Trên dãy B , bạn hãy thực hiện N phép biến đổi. Với phép biến đổi thứ i :

- Thêm a_i vào cuối dãy B .
- Đảo ngược thứ tự các phần tử của dãy B .

Bạn hãy tìm kết quả của dãy B sau khi thực hiện N phép biến đổi.

Dữ liệu vào

- Dòng đầu tiên, chứa số nguyên dương N .
- Dòng tiếp theo, chứa N số nguyên a_1, a_2, \dots, a_n .

Giới hạn:

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq a_i \leq 10^9$

Dữ liệu ra

- Gồm một dòng duy nhất là kết quả bài toán.

Ví dụ

Input #1

4
1 2 3 4

Output #1

4 2 1 3

Input #2

3

1 2 3

Output #2

3 1 2

Bài 41.

1. Cài đặt cấu trúc dữ liệu trừu tượng cây tìm kiếm nhị phân sử dụng danh sách liên kết.
2. Ứng dụng cây tìm kiếm nhị phân vào bài toán quản lý 1 lớp sinh viên, mỗi sinh viên gồm các thông tin: MSV (MSV là duy nhất), Họ tên, tuổi.

Xây dựng menu gồm các chức năng sau:

- a. Thêm 1 sinh viên vào lớp, nếu sinh viên đã tồn tại thì thông báo lỗi.
- b. Xóa 1 sinh viên trong lớp theo MSV. Nếu mã không có thì thông báo lỗi.
- c. Sửa thông tin 1 sinh viên trong lớp.
- d. Tìm kiếm sinh viên theo MSV
- e. Duyệt các sinh viên theo thứ tự trước
- f. Duyệt các sinh viên theo thứ tự giữa
- g. Duyệt các sinh viên theo thứ tự sau.

Bài 42

Ứng dụng lớp List để quản lý các đối tượng là các Khách hàng. Mỗi Khách hàng cần quản lý các thông tin sau: Mã KH là 1 số nguyên, Họ tên, SĐT.

1. Bổ sung vào lớp List phương thức sắp xếp theo thuật toán QuickSort, để sắp xếp các phần tử theo thứ tự một hàm so sánh được vào cho hàm Quicksort

2. Viết chương trình cho phép thực hiện các chức năng sau:

- Nhập thêm các khách hàng, sau khi hoàn thành nhập thông tin của một khách hàng, chương trình đưa ra câu hỏi có nhập nữa không (y/n)? Nếu người dùng nhập: Y thì tiếp tục nhập, nhập N thì kết thúc.
- Xóa đi một khách hàng
- Sửa đổi thông tin họ tên, SĐT của một khách hàng bất kỳ trong danh sách
- Sắp xếp danh sách khách hàng theo Mã KH
- Sắp xếp danh sách khách hàng theo họ tên.
- Hiện thị toàn bộ danh sách khách hàng hiện có.

Bài 43

Ứng dụng lớp List để quản lý các đối tượng là các Khách hàng. Mỗi Khách hàng cần quản lý các thông tin sau: Mã KH là 1 số nguyên, Họ tên, SĐT.

Viết chương trình cho phép thực hiện các chức năng sau:

Nhập thêm các khách hàng, sau khi hoàn thành nhập thông tin của một khách hàng, chương trình đưa ra câu hỏi có nhập nữa không (y/n)? Nếu người dùng nhập: Y thì tiếp tục nhập, nhập N thì kết thúc.

· Sử dụng thuật toán tìm kiếm nhị phân, tìm 1 khách hàng trong danh sách có mã nhập vào từ bàn phím. Nếu tìm thấy thì hiện ra 1 menu gồm các chức năng:

1. Xóa đi một khách hàng

2. Sửa đổi thông tin họ tên, SĐT của khách hàng

Hiển thị toàn bộ danh sách khách hàng hiện có.