

BÀI TẬP LỚP

HỌC PHẦN CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT

- Mỗi sinh viên làm bài tập gồm 2 phần

PHẦN A.

BÀI TOÁN QUẢN LÝ

Sinh viên làm bài tập về quản lý (đề tài sinh viên tự đề xuất). Sinh viên sử dụng (không xây dựng chỉ sử dụng cấu trúc vector hoặc list có sẵn trong STL làm bài toán quản lý theo các yêu cầu sau:

+ Tên do sinh viên tự đề xuất (ví dụ quản lý sinh viên, quản lý điểm, quản lý giảng viên, quản lý hàng hóa, quản lý thời khóa biểu, quản lý ô tô ...) mỗi sinh viên một đề tài

+ Xây dựng ít nhất 03 lớp gồm:

- Xây dựng lớp đối tượng quản lý có ít nhất 04 trường dữ liệu (ví dụ sinh viên gồm mã, tên, tuổi, điểm) với ít nhất 03 toán tử >> và << và 01 toán tử so sánh < (theo 1 trường nào đó ví dụ điểm)

- Xây dựng lớp danh sách đối tượng quản lý (dùng cấu trúc vector **nếu số cuối cùng của mã sinh viên của bạn là số chẵn**, dùng cấu trúc list **nếu số cuối cùng của mã sinh viên của bạn là số lẻ**) cho phép nhập, xuất danh sách, sắp xếp danh sách (dùng hàm sort có sẵn đối với vector và phương thức sort có sẵn đối với list) và ít nhất 03 thao tác, ví dụ: tìm max, min, tìm kiếm, xóa, thêm ...

- Xây dựng lớp app để quản lý có menu và thực hiện các thao tác ở lớp danh sách

- Viết hàm main chạy chương trình

PHẦN B.

SINH VIÊN LÀM 1 TRONG CÁC BÀI SAU (CÓ DANH SÁCH **PHÂN CÔNG** KÈM THEO)

Chú ý: Yêu cầu các cấu trúc dữ liệu ngăn xếp, hàng đợi, vector, list, tree, ... phải tự code không sử dụng thư viện có sẵn

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:
 - a. Nhập vào một biểu thức dạng trung tố
 - b. Chuyển biểu thức đó sang dạng hậu tố
 - c. Tính giá trị của biểu thức

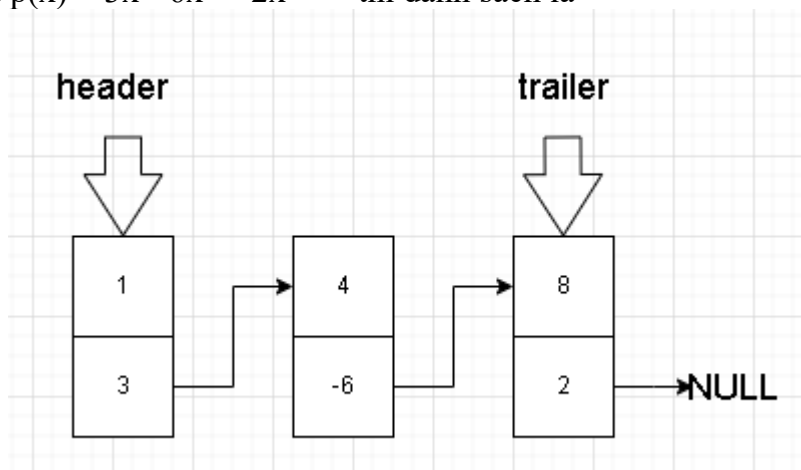
Bài 2

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

- a. Toán tử nhập, xuất
- b. Tính giá trị của đa thức
- c. Toán tính cộng, trừ, nhân 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 $p(x) = 3x - 6x^4 + 2x^8$ thì danh sách là



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

Bài 3

1. Xây dựng lớp chuỗi ký tự có các phương thức:
 - a. Cắt các ký trắng (dấu cách) ở hai đầu của chuỗi

- Ví dụ: $s = \text{ " Ha Noi "}$, sau khi cắt ký tự trắng $s = \text{ "Ha Noi"}$
- b. Cắt toàn bộ ký tự trắng thừa có trong xâu (giữa hai từ mà có n ($n > 2$) ký tự trắng thì số ký tự trắng thừa là $n-1$.
Ví dụ: $s = \text{ "Ha Noi"}$, sau khi cắt ký tự trắng $s = \text{ "Ha Noi"}$
 - c. Chuyển đổi toàn bộ xâu ký tự thành xâu ký tự hoa
 - d. Chuyển đổi toàn bộ xâu ký tự thành xâu ký tự thường
 - e. Thực hiện nối thêm một xâu ký tự vào sau xâu ký tự hiện có
 - f. Phương thức trích ra n ký tự ở phía phải của xâu
 - g. Phương thức trích ra n ký tự ở phía trái của xâu
 - h. Phương thức trích ra n ký tự kể từ vị trí thứ i của xâu
 - i. Nhập
 - j. In
2. Viết chương trình kiểm tra sự hoạt động của lớp đã xây dựng

Bài 4

1. Xây dựng lớp cây tìm kiếm nhị phân
 2. Xây dựng một lớp quản lý một 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 phương thức:
 - a. Đọc từ điển từ file để xây dựng cây
 - b. 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*)
 - c. Thêm một từ mới vào từ điển đang lưu trong cây
 - d. Xóa bỏ một từ của từ điển đang lưu trong cây
 - e. Cập nhật lại nghĩa một từ đang lưu trong cây
 - f. Tìm kiếm một từ trong cây.
2. Viết chương trình kiểm tra sự hoạt động của lớp từ điển đã xây dựng

Bài 5

1. Xây dựng lớp biểu diễn một bảng băm sử dụng một trong những hàm băm đã học
 - 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:
 - a. Nạp từ điển từ file vào bảng băm
 - b. Tìm kiếm bằng phương pháp tìm kiếm trên bảng băm
 - c. Sửa đổi các từ
 - d. Thêm từ mới
 - e. Xóa từ
 - f. Lưu từ điển vào file
2. Xây dựng chương trình có các chức năng
 - a. Nạp từ điển từ file (có thể nạp tự động khi chạy chương trình)
 - b. Tra từ điển (nhập vào một từ, hiển thị nghĩa của nó nếu có trong từ điển)
 - c. Sửa đổi từ
 - d. Thêm từ mới

- e. Xóa từ
- f. Lưu từ điển vào file

Bài 6

1. Xây dựng một lớp quản lý một dãy số, xây dựng cấu trúc dữ liệu danh sách liên kết đơn để lưu trữ. Với các phương thức:
 - a. Tạo danh sách số, quá trình nhập sẽ dừng lại khi nhập dấu “#”.
 - b. Thêm một phần tử vào danh sách, vị trí thêm vào do người dùng chọn.
 - c. Nhập vào một số k ($k \neq 0$), đếm xem trong dãy có bao nhiêu số có giá trị = k ?
 - d. 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.
 - e. Sắp xếp danh sách theo thứ tự tăng dần.
 - f. Xóa tất cả các số nguyên tố trong danh sách.
 - g. Xóa tất cả các phần tử có giá trị trùng nhau trong danh sách, chỉ giữ lại 1 số.
2. Xây dựng hàm main để kiểm tra các chức năng của lớp.

Bài 7

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.
- Ghi từ điển vào file.

Bài 8

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 9

Xây dựng lớp biểu diễn một bảng băm sử dụng một trong những hàm băm đã học. 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 sách 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 khi biết tên
- 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, số lượng*

Bài 10

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 (**dùng cấu trúc dữ liệu nào thì tự cài đặt cấu trúc đó**). Mỗi phần tử của danh sách liên kết lưu một chữ số thập phân của số lớn. Ví dụ số 4728

Có các phương thức:

- a. Nhân một số lớn với một số kiểu int
 - b. Nhân hai số lớn
 - c. Cộng hai số lớn
 - d. 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 11

1. Xây dựng các lớp biểu diễn danh sách liên kết kép.
2. Ứng dụng lớp List xây dựng lớp 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. Lớp có các phương thức sau:
 - a. 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.
 - b. Tìm kiếm 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:
 - Xóa đi một khách hàng
 - Sửa đổi thông tin họ tên, SĐT của khách hàng.
 - c. Hiện thị toàn bộ danh sách khách hàng hiện có.

Bài 12

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) dùng stack không dùng đệ quy để tìm 1 đường đi giữa 2 đỉnh bất kỳ
3. Viết hàm main thực hiện các công việc trên

Bài 13

1. Xây dựng cấu trúc dữ liệu hàng đợi
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 rộng (BFS) để tìm đường đi ngắn nhất giữa 2 đỉnh bất kỳ
- 4. Viết hàm main thực hiện các công việc trên

Bài 14

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 15

1. Xây dựng lớp Vector
 2. Xây dựng 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.
- Với các phương thức:
- a. Đọc danh thông tin quản lý của cửa hàng từ file vào chương trình
 - b. Ghi thông tin quản lý của cửa hàng vào file
 - c. Nhập xuất các mặt hàng
 - d. Hiển thị toàn bộ danh sách mặt hàng hiện có
 - e. Thống kê số lượng mặt hàng tồn của mỗi mặt hàng
2. Viết hàm main thực hiện các chức năng trên

Bài 16

1. Xây dựng danh sách liên kết đơn để giải quyết bài toán đồ thị
 - Xây lớp đồ thị (Graph) vô hướng không có trọng số, đồ thị được mô tả bằng danh sách kề, có các phương thức:
 - a. Đọc đồ thị từ file
 - b. Ghi đồ thị ra file
 - c. Dùng thuật toán DFS (sử dụng stack có sẵn trong STL) đếm số thành phần liên thông của đồ thị
 - d. Dùng thuật toán BFS (sử dụng queue có sẵn trong STL) để kiểm tra có tồn tại đường đi giữa 2 điểm nào đó không.
2. Viết hàm main, thực hiện các chức năng của lớp Graph

Bài 17

1. Cho cây nhị phân trừu tượng (cây heap) 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 nào đó tự đề xuất hoặc bài nổi thanh kim loại <http://laptrinhonline.club/problem/tichpxnoidam>

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

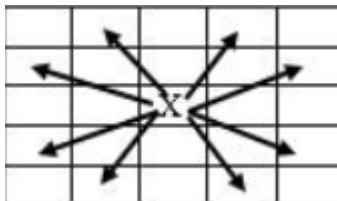
Bài 18

1. Xây dựng cấu trúc dữ liệu hàng đợi

2. Áp dụng cấu trúc dữ liệu hàng đợi vào thuật toán BFS giải quyết bài toán sau:

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

Trong luật cờ vua, mỗi nước đi của quân mã được quy định như sau: quân mã đang ở vị trí như hình bên dưới có thể đi đến một trong các ô mà mũi tên chỉ đến (theo đường chéo của hình chữ nhật 2×3).



Yêu cầu

Cho trước bàn cờ kích thước $n \times m$ ô. Hãy đếm số nước đi ít nhất để quân mã di chuyển từ ô có tọa độ $(x1, y1)$ đến ô có tọa độ $(x2, y2)$. Trong trường hợp không đến được thì xuất ra giá trị -1 .

Input

Dòng 1 ghi hai số nguyên dương n, m

Dòng 2 ghi hai số nguyên $x1, y1$

Dòng 3 ghi hai số nguyên $x2, y2$

Output

Ghi ra một số nguyên duy nhất cho biết số nước đi ít nhất để quân mã di chuyển từ ô $(x1, y1)$ đến ô $(x2, y2)$. Nếu không đến được thì ghi số -1 .

Ví dụ

Input:

```
4 6
1 1
2 4
```

Output:

```
2
```

Bài 19

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

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

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 tíchpx có các tiền tố t, ti, tic, tích, tíchp, tíchpx). 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 20

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 21

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 22

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 23

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 24

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:

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

Bài 25

Tìm hiểu về mã hóa Huffman, hàng đợi ưu tiên và thuật toán xây dựng cây Huffman. Xây dựng lớp biểu diễn cây Huffman có các chức năng sau:

- 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 26

Tìm hiểu về cấu trúc đồng nhị phân.

- 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.
- Ứng dụng cấu trúc hàng đợi ưu tiên ở trên để cài đặt thuật toán tìm cây khung nhỏ nhất Prim.

Bài 27

- Xây dựng lớp cây nhị phân.

- Ứng dụng cây nhị phân giải các bài toán sau:

- Cho một cây nhị phân. Hãy liệt kê tất cả các đường đi từ gốc đến lá.
- 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: $12 + 13 = 25$

- 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 28

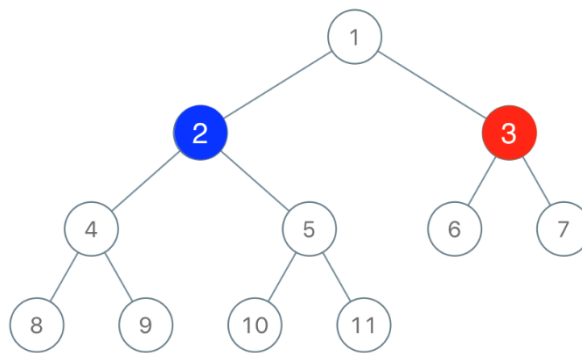
- Xây dựng lớp cây nhị phân.
- Ứng dụng cây nhị phân cho bài toán sau:

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 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: Đấu 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 29

- 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 30

1. Cài đặt lớp 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.
2. Ứng dụng giải các bài toán sau:
 - a. 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.
 - b. 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.
 - a. 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 31

1. Cài đặt lớp 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.
2. 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 32

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 33

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 34

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 xây dựng lớp quản lý các đối tượng sinh viên, mỗi sinh viên gồm các thông tin: MSV (MSV là duy nhất), Họ tên, tuổi. Lớp có các phương thức:
 - a. Đọc danh sách sinh viên từ tệp vào cây (*cấu trúc file lưu trữ danh sách sinh viên do người lập trình tự định nghĩa*)
 - b. Thêm 1 sinh viên vào cây, nếu sinh viên đã tồn tại thì thông báo lỗi.
 - c. Xóa 1 sinh viên trong lớp theo MSV. Nếu mã không có thì thông báo lỗi.
 - d. Cập nhật thông tin 1 sinh viên khi biết MSV và các thông tin cần cập nhật.
 - e. Tìm kiếm sinh viên theo MSV
 - f. Hiển thị danh sách sinh viên lên màn hình.
 - g. Ghi toàn bộ sinh viên của cây vào file.

HƯỚNG DẪN LÀM BÀI TẬP LỚN

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

Sinh viên thực hiện: **XX** - Lớp CNTT **YY-KZZ**

Đề bài số: AB

Hà Nội tháng 11 năm 20...

Nội dung báo cáo

PHẦN A.

1. Đề bài

2. Phân tích bài toán

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

3. Cài đặt các lớp và hàm main bằng C++

PHẦN B

1. Đề bài

2. 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.

3. Cài đặt các lớp và hàm main bằng C++

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