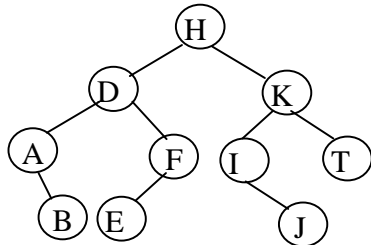


1. Trong danh sách đặc, khi thêm phần tử vào vị trí p trong danh sách ta cần phải:

- Luôn luôn dịch chuyển các phần tử từ p đến L.Last ra sau một vị trí.
- Có thể dịch chuyển các phần tử từ p đến L.Last ra sau một vị trí.**
- Luôn luôn dịch chuyển các phần tử từ p+1 đến L.Last ra sau một vị trí.
- Có thể dịch chuyển các phần tử từ p+1 đến L.Last ra sau một vị trí.

2. Cho cây nhị phân sau:



Cây nhị phân đã cho là cây tìm kiếm nhị phân

a. **Đúng**

b. Sai

3. Danh sách duyệt NLR của cây nhị phân ở câu 2 là:

- A, B, D, E, F, H, I, J, K, T
- H, D, A, B, E, F, K, I, J, T
- H, D, A, B, F, E, K, I, J, T**
- B, A, E, F, D, J, I, T, K, H

4. Danh sách duyệt LRN của cây nhị phân ở câu 2 là:

- A, B, D, E, F, H, I, J, K, T
- H, D, A, B, E, F, K, I, J, T
- B, A, D, H, F, E, K, I, J, T
- B, A, E, F, D, J, I, T, K, H**

5. Danh sách duyệt trung tự tổng quát của cây nhị phân ở câu 2 là:

- B, A, D, E, F, H, J, I, K, T**
- B, A, D, E, F, H, I, J, K, T
- A, B, D, E, F, H, I, J, K, T
- A, B, D, E, F, H, J, I, K, T

6. Giá trị biểu thức tiền tố: /, *, -, 2, 1, +, 3, 4, *, 7, -, 6, 5 là

- 7
- 1
- 1**
- 7

7. Khi thêm phần tử đầu tiên vào danh sách liên kết L rỗng có ô đầu mục thì

- L luôn luôn thay đổi
- L có thể thay đổi
- L luôn luôn không đổi**

8. Vị trí để thêm phần tử vào hàng là

- Vị trí đầu hàng
- Vị trí cuối hàng**
- Vị trí bất kỳ trong hàng

9. Khi xóa phần tử ra khỏi danh sách đặc L không rỗng thì ta phải luôn luôn thực hiện câu lệnh:

- L.Last--;
- L->Last--;
- L->Last++;
- Tất cả đều sai**

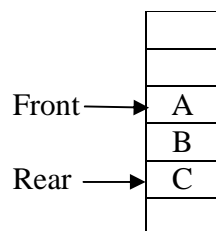
10. Giá trị biểu thức hậu tố 1, 2, -, 3, 4, +, *, 7, 6, 5, -, *, / là:

- 1**
- 1
- 7
- Một giá trị khác

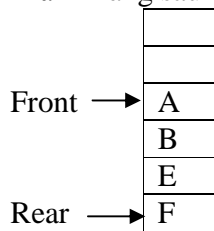
11. Lấy nội dung phần tử tại vị trí p trong danh sách liên kết kép L không có ô đầu mục ta thực hiện lệnh:

- return L->Element;
- return L->Next->Element;
- return p->Element;**

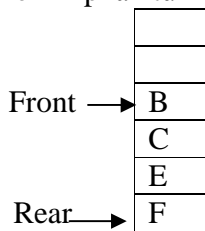
- d. return p->Next->Element;
12. Sau khi thêm phần tử X vào vị trí p trong danh sách đặc L, ta luôn luôn có mệnh đề sau là đúng:
- L.Elements[p-1]==X;
 - L.Elements[p]==X;
 - L.Elements[p+1]==X;
 - Tất cả đều sai.**
13. Khi thêm phần tử X vào bảng băm mở, ta sẽ xen phần tử đó vào đầu của danh sách ở vị trí:
- Bucket 0.
 - Bucket B-1
 - Bucket bất kỳ
 - Bucket H(X)**
14. Cho hàng cài đặt bằng mảng tĩnh tiến như sau:



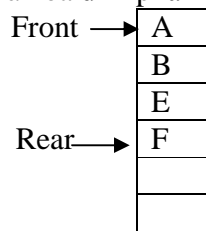
Hình ảnh hàng sau khi thêm 2 phần tử E, F và xóa đi 1 phần tử là:



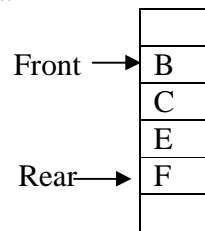
a.



b.



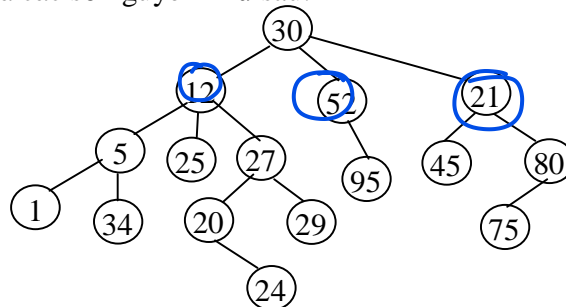
c.



d.

15. Khi thêm phần tử vào danh sách đặc không đầy, ta luôn luôn phải thực hiện lệnh
- L.Last++;
 - L->Last ++;
 - L->Last--;
 - Tất cả đều sai**
16. Khi xóa phần tử ra khỏi hàng cài đặt bằng mảng vòng thì
- Front luôn luôn thay đổi và Rear luôn luôn không đổi
 - Front luôn luôn thay đổi và Rear có thể thay đổi
 - Front có thể thay đổi và Rear có thể thay đổi**
 - Front có thể thay đổi và Rear luôn luôn không đổi
17. Vị trí phần tử cần xóa ra khỏi danh sách đặc trong phép toán Delete_List (p,L) chỉ hợp lệ khi:
- $0 \leq p \leq L.Last + 1$
 - $0 < p \leq L.Last + 1$
 - $1 \leq p \leq L.Last + 1$
 - $1 \leq p < L.Last + 1$**

18. Cho cây với nhãn là các số nguyên như sau:



Bậc của cây là:

- a. 1 b. 2 c. 3 d. 4

19. Danh sách duyệt theo mức của cây ở câu 18 là:

- a. 24, 1, 34, 20, 29, 75, 5, 25, 27, 95, 45, 80, 12, 52, 21, 30
b. 1, 34, 5, 25, 24, 20, 29, 27, 12, 95, 52, 45, 75, 80, 21, 30
c. 30, 12, 52, 21, 5, 25, 27, 95, 45, 80, 1, 34, 20, 29, 75, 24
d. Tất cả đều sai

20. Hãy minh họa cây ở câu 18 theo phương pháp cài đặt bằng mảng

Cách 1 :

Parent	-1	0	0	0	1	1	1	2	3	3	4	4	6	6	9	12				
Data	30	12	52	21	5	25	27	95	45	80	1	34	20	29	75	24				
Chỉ số mảng	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...	50

Maxnode=16

Cách 2 :

Parent	-1	0	1	2	2	1	1	6	7	6	0	10	0	12	12	14				
Data	30	12	5	1	34	25	27	20	24	29	52	95	21	45	80	75				
Chỉ số mảng	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	...	50

Maxnode=16

21. Cây tìm kiếm nhị phân là cây nhị phân với

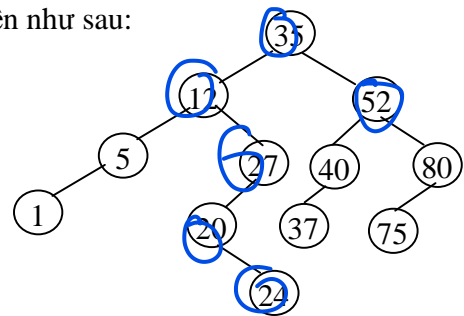
- a. Nhãn của nút lớn hơn nhãn của nút con trái và nhỏ hơn nhãn của nút con phải
b. Nhãn của nút lớn hơn nhãn của nút con phải và nhỏ hơn nhãn của nút con trái
c. Nhãn của nút lớn hơn nhãn của tất cả các nút ở cây con trái và nhỏ hơn nhãn của tất cả các nút ở cây con phải
d. Nhãn của nút lớn hơn nhãn của tất cả các nút ở cây con phải và nhỏ hơn nhãn của tất cả các nút ở cây con trái

22. Trong chương trình chính, để thêm phần tử x vào cuối danh sách L ta thực hiện lời gọi hàm như sau:

- a. InsertList (x, Last(L), L);
b. InsertList (x, Last(*L),L);
c. InsertList (x, EndList(&L),&L);
d. InsertList (x, EndList(L),&L);

23. Khi xóa nút 27 ra khỏi cây tìm kiếm nhị phân, ta thực hiện như sau:

- Thay nút 27 bởi nút 24 và quay về xóa nút 24.
- Thay nút 27 bởi nút 20 và quay về xóa nút 20.
- Cho con phải của nút 12 trở xuống nút 20.**
- Cho con phải của nút 12 trở xuống nút 24.



24. Số bước cần duyệt để tìm nút 25 trên cây ở câu 23 theo giải thuật Search(x,T) là: **0.6**

- 3
 - 4
 - 5
 - 6**
25. Xóa nút 52 trên cây ở câu 23, ta cần:
- Thay nút 52 bởi nút 37 và quay về xóa nút 37.
 - Thay nút 52 bởi nút 40 và quay về xóa nút 40.**
 - Thay nút 52 bởi nút 80 và quay về xóa nút 80.
 - Cả 2 câu b,c đều đúng

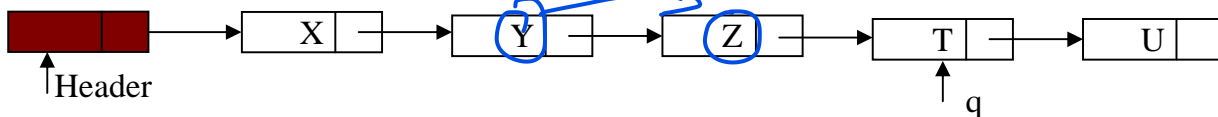
26. Thêm nút 42 vào cây ở câu 23, ta thực hiện:

- Không thêm được
 - Gắn nút 42 vào con phải của 37.
 - Gắn nút 42 vào con trái của 75.
 - Gắn nút 42 vào con phải của 40.**
27. Hãy cho biết thao tác nào không được phép dùng trên cấu trúc ngăn xếp
- Thêm một phần tử vào đỉnh ngăn xếp.
 - Xóa một phần tử ở vị trí bất kì khỏi ngăn xếp.
 - Thêm một phần tử vào vị trí bất kì trong ngăn xếp.
 - Cả b và c.**

28. Cho biểu thức trung tố: $(a * (5 + b) - 2 * c) + 4$, biểu thức nào sau đây là dạng **hậu tố** của nó ?

- $a * b + 2 - c * 4 +$
- $a * b + * 2 - c * 4 +$
- $a * b + * 2 * - 4 +$**
- Tất cả đều sai

29. Cho danh sách liên kết như hình



Kết quả trên màn hình sau khi thực hiện đoạn lệnh là

```
p=First(Header);
while (p->next!=q){
    printf("%c ", p->next->data);
    p=p->next;
}
```

- X Y Z T U
- X Y Z T
- X Y Z**
- Y Z T

30. Cho bảng băm đồng với số bucket $B=10$ và hàm băm $h(x)=x \bmod B$ với phương pháp giải quyết độ là băm lại bình phương ($h(x)=(x+i) \bmod B$). Kết quả bảng băm sau khi thực hiện các thao tác thêm 3, 5, 9, 12, 15, 16, 25 là:

a)	0		b)	0		c)	0		d)	0	
	1			1			1			1	
	2	12		2	12		2	12		2	12
	3	3		3	3		3	3		3	3
	4			4			4	25		4	
	5	5		5	5		5	5		5	5
	6	16		6	15		6	15		6	15
	7	15		7	16		7	16		7	25
	8	25		8	25		8			8	16
	9	9		9	9		9	9		9	9

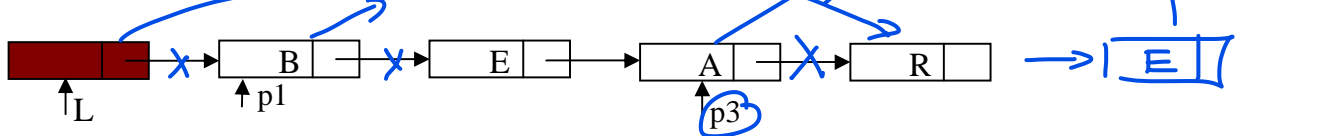
31. Danh sách duyệt LRN của cây nhị phân cho bởi cặp danh sách duyệt sau
~~NLR: A, B, C, D, G, E, F~~ và ~~LNR: C, B, D, G, A, E, F~~ là:

- C, B, G, E, D, A, F
- C, G, D, B, F, E, A**
- C, G, D, B, E, F, A
- Cặp danh sách đã cho không hợp lệ.

32. Danh sách duyệt một cây nhị phân theo phương pháp duyệt trung tự nhị phân và trung tự tổng quát luôn luôn khác nhau khi và chỉ khi

- Có tồn tại nút bị khuyết con trái trên cây
- Có tồn tại nút bị khuyết con phải trên cây
- Có tồn tại nút có con phải mà bị khuyết con trái trên cây.**
- Có tồn tại nút có con trái mà bị khuyết con phải trên cây

33. Cho danh sách liên kết sau



$p3 \rightarrow \text{Next} \rightarrow \text{Next} = p1 \rightarrow \text{Next};$

$L \rightarrow \text{Next} = p3 \rightarrow \text{Next};$

$p1 \rightarrow \text{Next} = \text{NULL};$

$p3 \rightarrow \text{Next} = p1;$

$p1 \rightarrow \text{Element} = \text{Retrieve}(p3, L);$

PrintList(L); // Hàm in danh sách ra màn hình

Kết quả hiển thị trên màn hình là:

- REAB**
- BEAR
- REAR
- Một danh sách khác

34. Khi thêm phần tử vào hàng cài đặt bằng mảng vòng không rỗng thì:

- Front luôn luôn không đổi và Rear luôn luôn thay đổi
- Front luôn luôn không đổi và Rear có thể thay đổi**
- Front có thể thay đổi và Rear luôn luôn thay đổi
- Front có thể thay đổi và Rear có thể thay đổi

35. Danh sách duyệt trung tự tổng quát của cây nhị phân cho bởi hai danh sách

LRN: 5, 6, 4, 2, 9, 10, 8, 7, 3, 1 và LNR: 2, 5, 4, 6, 1, 7, 9, 8, 10, 3 là

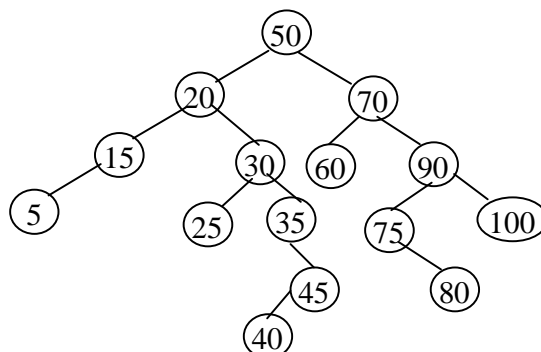
- 5, 4, 2, 6, 1, 9, 8, 10, 3, 7

- b. 5, 4, 6, 2, 1, 9, 8, 10, 7, 3
 c. 2, 5, 4, 6, 3, 1, 8, 7, 9, 10
 d. 4, 5, 6, 2, 1, 3, 7, 9, 8, 10
36. Cây ở câu 35 có số nút trung gian là: *no r hai cấu gồ và hđ.*
 a. 4 b. 5 c. 6 d. 7
37. Cây ở câu 35 có nút 7 là
 a. Nút lá
 b. Nút trung gian có 2 con
 c. Nút trung gian có 1 con phải
 d. Nút trung gian có 1 con trái
38. Chiều cao của cây ở câu 35 là:
 a. 2 b. 3 c. 4 d. 5
39. Hàm sau được thiết kế trong danh sách liên kết với nội dung là:
 Position **NONAME** (ElementType x, List L)
 { Position p=First(L);
 while (p!= Endlist(L))
 if (Retrieve (P,L)==x) return 1;
 else p=Next(p,L);
 return 0;
 }
 Hàm này thực hiện được không và làm nhiệm vụ gì?
 a. Hàm trả về tìm vị trí phần tử đầu có nội dung là x và không bị lỗi.
 b. **Hàm trên bị lỗi ở kiểu dữ liệu trả về cho hàm.**
 c. Hàm kiểm tra xem có phần tử X trong danh sách hay không và hàm không bị lỗi.
 d. Hàm luôn trả về giá trị 0.
40. Nguyên tắc làm việc của ngăn xếp là:
 a. FIFO b. **FILO**
 c. LILO d. Câu a và c cùng đúng.

II. Phần tự luận

Câu 1: (1.5 điểm)

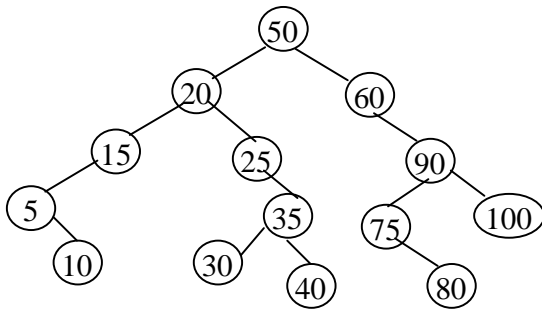
- a. Vẽ cây tìm kiếm nhị phân A từ danh sách đã cho: **(0.5 điểm)**
 50, 70, 20, 30, 90, 35, 15, 45, 30, 100, 60, 75, 80, 5, 25, 40



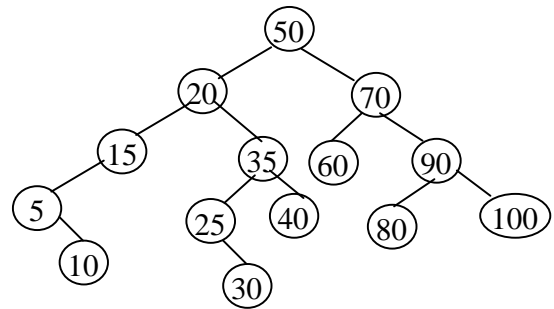
- b. Viết danh sách duyệt trung tự tổng quát của cây A **(0.25 điểm)**
 5, 15, 20, 25, 30, **40, 45, 35**, 50, 60, 70, **80, 75**, 90, 100

- c. Vẽ lại cây sau khi thực hiện các thao tác xóa nút 30, 70, 45 và thêm vào nút 30, 20, 10 (0.75 điểm)

Cách 1:



Cách 2:



Câu 2: (1.5 điểm)

Viết hàm xác định số nút trong của một cây nhị phân. Cho phép sử dụng các phép toán cơ bản trên cây.

(Sinh viên có thể làm nhiều cách khác nhau và được hưởng trọn điểm khi xây dựng hàm thỏa yêu cầu đặt ra)

- Xây dựng hàm tính tổng số nút trên cây: 0.5 điểm

- Xây dựng hàm xác định số nút lá trên cây: 0.5 điểm

- Tính số nút trung gian (có xét đến trường hợp cây rỗng, cây có 1 nút và trường hợp cây có nhiều nút : 0.5 điểm).