

**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**  
**BỘ MÔN KHOA HỌC MÁY TÍNH**

\*\*\*

Mã đề  
**DH 20133 - 01**



<b>Họ tên:</b> .....	<b>ĐỀ THI MÔN: CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT</b>	Hà nội, ..... / ..... / .....
<b>Lớp:</b> .....	<b>Ngày thi:</b> ...../...../.....	<b>Trường bộ môn</b>
<b>SHSV:</b> .....	<b>Thời gian 90'</b> (Sinh viên được sử dụng tài liệu)	

**Bài 1.**

a) Cho **biểu thức trung tố** sau

$$(3 + a)/2 - c + 2/a$$

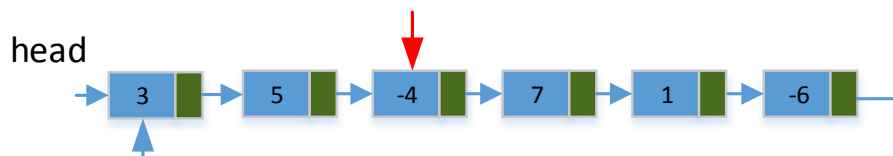
Hãy xây **dựng cây biểu thức** tương ứng

b) Giả sử bạn có một câu nhị phân chỉ gồm có các **nút trong có 2 con hoặc nút lá**. Khi **duyet cây theo thứ tự** trước từ gốc ta sẽ ký hiệu là L nếu đi theo con trái và R nếu đi theo con phải.  
Với thứ tự **duyet trước** thu được của một cây là **LLRLRLRLR** thì **chiều cao** của cây này là?

**Bài 2.** Danh sách liên kết đơn chứa n số nguyên

```
struct Node
{
    int data;
    struct Node *pNext;
}
```

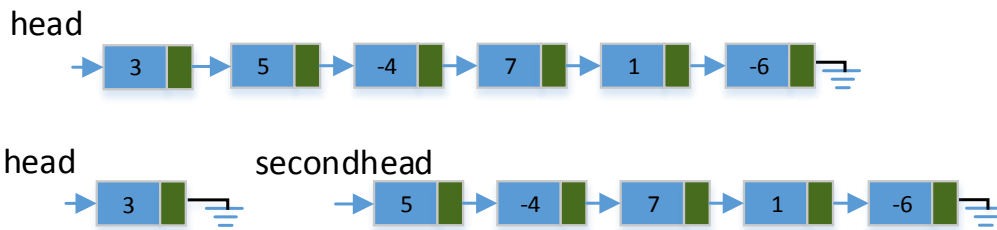
a) Giả sử danh sách liên kết đơn này được nối vòng. Hãy viết hàm tìm và trả về phần tử giữa danh sách một cách nhanh nhất



```
struct Node findMid(struct Node *head)
```

b) Hoàn thiện hàm **tìm vị trí cắt danh sách thành 2 nửa** sao cho tổng độ chênh lệch giữa 2 nửa là nhỏ nhất

```
void splitLinkedList(struct Node **head, struct Node **secondhead)
```



### Bài 3. STACK QUEUE

- Khi biểu diễn ngăn xếp trong trường hợp tổng quát ta nên dùng Mảng hay danh sách liên kết? Vì sao?
- Cho tập hợp gồm  $n$  điểm đen và  $n$  điểm trắng cách đều nhau. Hãy mô tả thuật toán kết nối một điểm đen với một điểm trắng sao cho tổng khoảng cách là nhỏ nhất.



- Cho một danh sách gồm thời gian của  $n$  buổi họp trong một ngày. Hãy xây dựng hàm tìm số phòng cần sử dụng **ít nhất** để đảm bảo các cuộc họp trùng nhau về thời gian sẽ được bố trí ở các phòng khác nhau.

```

struct Meeting
{
    int start, finish;
}

int findNumberOfRooms(struct Meeting A[], int n)

```

Để cho đơn giản, ta tính thời gian bắt đầu và kết thúc (start, finish) bằng số phút chênh lệch so với 00:00 của ngày hôm đó.

- Bài 4.** Cho một mảng chứa  $n$  số nguyên bất kỳ, hãy viết hàm tìm và in ra màn hình **dãy con tăng lớn nhất**. Đánh giá theo  $O$  lớn về thời gian thực hiện và bộ nhớ sử dụng trong trường hợp tồi nhất.

```

void maxIncrementalSequence(int A[], int n)

```

- Bài 5.** Hãy xây dựng hàm tính  $x^n$  với ( $n \geq 0, x > 0$ ) với thời gian cỡ  $O(\log n)$

- Bài 6.** Cây nhị phân

```

struct BNode
{
    int data;
    struct BNode *left, *right, *parent;
}

```

- Viết hàm in ra các nút lá trên cây mà **không dùng đệ quy**

```

void printLeaves(struct BNode *root)

```

- Viết hàm đếm số nút trong có đầy đủ 2 con trên cây nhị phân

```

int countNodes(struct BNode *root)

```