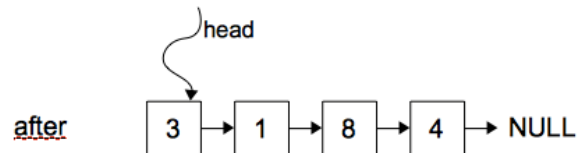
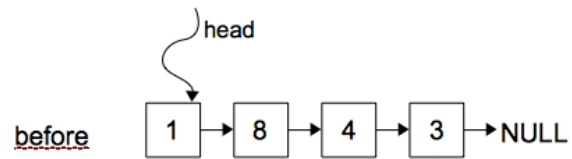


3. The insertBeginning(node* head) method inserts the last node of the given linked list to the beginning of the list (see the figure below). Please find the logical error(s) and correct them on the code given below:

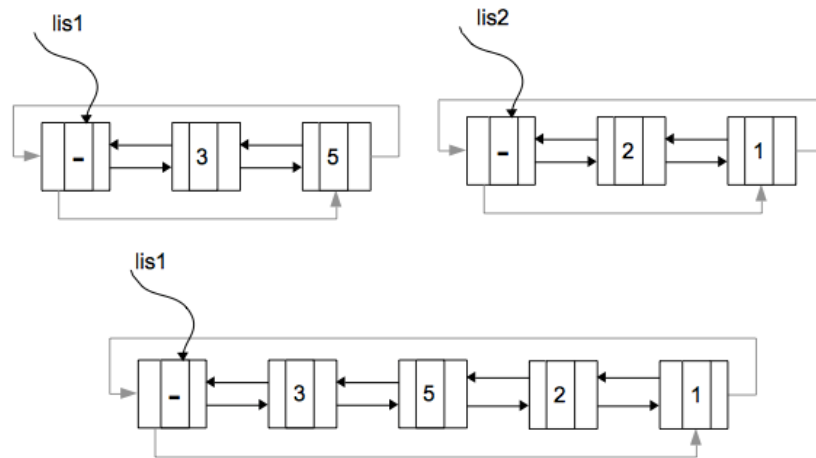


```
typedef struct node{
    int x;
    struct node *next;
}node;

void insertBeginning(node* head){
    node *pre, *old_head=head;
    for(;;head->next;pre=head,head=head->next);
    pre->next=NULL;
    head->next=old_head;
}

int main()
{
    node* head;
    .....
    insertBeginning(head);
    .....
}
```

4. Complete the following method that concatenates two circular doubly linked lists by removing the header of the second linked list (see the figures below).



```

typedef struct node{
    int x;
    struct node *pre; //previous link
    struct node *fol; //following link
}node;

node* concatLists(node* list1, node* list2){

if(list2->pre==list2){ // or if(list2->fol==list2)
    free(list2);
    return list1;
}

list1->pre=list2->pre;
list2->pre->fol=list1;
list1->pre->fol=list2->fol;
list2->fol->pre=list1->pre;

}

```

5. Please convert the following prefix expression into infix and postfix notations:

$/*-6/7+5/3+1*24-15-28$

a) Infix notation:

b) Postfix notation:

6. Draw the linked list implementation of the given sparse matrix according to the given structure definitions. You do not need to write the tag fields.

```
typedef enum {head, entry} tagfield;
typedef struct matrix_node * matrix_pointer;
typedef struct entry_node{
    int col;
    int row;
    int value;
};
typedef struct matrix_node{
    matrix_pointer down;
    matrix_pointer right;
    tagfield tag;
    union {
        matrix_pointer next;
        entry_node entry;
    }u;
};
```

0	1	0	0
2	0	1	0
0	0	0	1