

COP 3502 Section 4

Lab Assignment # 3

Suppose we have a linked list implemented with the structure below. Write a function that will take in a pointer to the head of a list and inserts a node storing -1 after each even value in the list. If the list is empty or there are no even values in the list, no modifications should be made to the list. (For example, if the initial list had 2, 6, 7, 1, 3, and 8, the resulting list would have 2, -1, 6, -1, 7, 1, 8, -1.)

```
#include <stdio.h>

typedef struct node {
    int data;
    struct node* next;
} node;

void markEven(node* head) {
}

void deallocateList(node** head)
{
    node* tmp = *head;

    while (tmp != NULL)
    {
        *head = tmp->next;
        free(tmp);
        tmp = *head;
    }

    *head = NULL;
}

int insertList(node** head, int data)
{
    node* newNode = (node*) malloc(sizeof(node));

    if (newNode == NULL)
        return -1;

    newNode->data = data;
    newNode->next = NULL;

    newNode->next = *head;
    *head = newNode;

    return 1;
}
```

```

void printList(node* head)
{
    node* tmp = head;

    while (tmp != NULL)
    {
        printf("%d",tmp->data);
        if (tmp->next != NULL)
        {
            printf(", ");
        }
        else
        {
            printf(".\n");
        }
        tmp = tmp->next;
    }
}

```

```

int main(void)
{
    node* head = NULL;

    insertList(&head, 8);
    insertList(&head, 3);
    insertList(&head, 1);
    insertList(&head, 7);
    insertList(&head, 6);
    insertList(&head, 2);

    printf("Initial List:\n");
    printList(head);
    markEven(head);
    printf("Marked List:\n");
    printList(head);

    deallocateList(&head);
}

```