#include <stdio.h>

#include <string.h>

#include <malloc.h>

#define MAX 100

typedef char element;

typedef struct listNode

{

element data[MAX];

struct listNode \*next;

}listNode;

listNode \*setNode(element data[])

{

listNode \*newNode = (listNode\*)malloc(sizeof(listNode));

strcpy(newNode->data, data);

newNode->next = NULL;

return newNode;

}

void insertLast(listNode \*\*head, element data[])

{

listNode \*newNode = setNode(data);

if(\*head == NULL)

{

\*head = newNode;

}

else

{

for(listNode \*curr = \*head; curr != NULL; curr = curr->next)

{

if(curr->next == NULL)

{

curr->next = newNode;

break;

}

}

}

}

void insertAfter(listNode \*head, element pre[], element data[])

{

int count = 0;

listNode \*newNode = setNode(data);

for(listNode \*curr = head; curr != NULL; curr = curr->next)

{

if(strcmp(curr->data, pre) == 0)

{

count++;

newNode->next = curr->next;

curr->next = newNode;

break;

}

}

if(count == 0)

printf("Insert Failed(%s does not exist)\n", pre);

}

void printList(listNode \*head)

{

for(listNode \*curr = head; curr != NULL; curr = curr->next)

{

if(curr->next == NULL)

printf("%s\n", curr->data);

else

printf("%s-", curr->data);

}

}

void freeList(listNode \*head)

{

printf("\n");

listNode \*curr;

listNode \*temp;

curr = head;

while(curr != NULL)

{

printf("%s freed\n",curr->data);

temp = curr->next;

free(curr);

curr = temp;

}

}

int main(int argc, char \*argv[])

{

listNode \*head = NULL;

insertLast(&head, "Apple"); printList(head);

insertLast(&head, "Durian"); printList(head);

insertAfter(head, "Apple", "Banana"); printList(head);

insertAfter(head, "Tomato", "Grape"); printList(head);

insertAfter(head, "Banana", "Carrot"); printList(head);

freeList(head);

return 0;

}