#include <stdio.h>

#include <stdlib.h>

// 定义链表节点结构

typedef struct Node {

int data;

struct Node\* next;

} Node;

// 初始化链表头节点

Node\* createList() {

return NULL;

}

// 向链表中添加节点

void addNode(Node\*\* head, int data) {

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

if (newNode == NULL) {

printf("内存分配失败\n");

exit(1);

}

newNode->data = data;

newNode->next = \*head;

\*head = newNode;

}

// 输出链表数据

void printList(Node\* head) {

Node\* current = head;

while (current != NULL) {

printf("%d ", current->data);

current = current->next;

}

printf("\n");

}

// 反转链表

Node\* reverseList(Node\* head) {

Node\* prev = NULL;

Node\* current = head;

Node\* next = NULL;

while (current != NULL) {

next = current->next;

current->next = prev;

prev = current;

current = next;

}

return prev;

}

// 销毁链表

void destroyList(Node\* head) {

Node\* current = head;

Node\* next = NULL;

while (current != NULL) {

next = current->next;

free(current);

current = next;

}

}

int main() {

Node\* head = createList();

int choice, data;

while (1) {

printf("1. 添加节点\n");

printf("2. 输出链表\n");

printf("3. 反转链表\n");

printf("4. 输出反转后链表\n");

printf("5. 销毁链表并退出\n");

printf("请输入选择: ");

scanf("%d", &choice);

switch (choice) {

case 1:

printf("请输入要添加的数据: ");

scanf("%d", &data);

addNode(&head, data);

break;

case 2:

printf("链表数据: ");

printList(head);

break;

case 3:

head = reverseList(head);

printf("链表已反转\n");

break;

case 4:

printf("反转后链表数据: ");

printList(head);

break;

case 5:

destroyList(head);

printf("链表已销毁，程序退出\n");

return 0;

default:

printf("无效的选择，请重新输入\n");

}

}

return 0;

}