**Circular Linked List의 ADT 구현**

5645640 제갈건 0321

1. 실행코드

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <stdlib.h>

typedef int element;

//head->link는 첫번째 노드를 가리킴

typedef struct ListNode {

element data;

struct ListNode\* link;

} ListNode;

//리스트 출력 함수

void print\_list(ListNode\* head){

ListNode\* p;

if (head == NULL) return;

p = head->link; //첫번째 노드

do {

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

p = p->link;

} while (p != head);// // 마지막 노드에 도착하면 순환을 끝내라

printf("%d->\n", p->data); // 마지막 노드 출력

}

//첫번째 노드에 insert하는 함수

ListNode\* insert\_first(ListNode\* head, element data){

ListNode\* node = (ListNode\*)malloc(sizeof(ListNode));

node->data = data;

//head가 NULL이면 head에 insert

if (head == NULL) {

head = node;

node->link = head;

}

else {

node->link = head->link;//node->link를 첫번째 노드로 연결

head->link = node; // head의 포인터를 노드에 연결하여 노드를 첫번째로 변경

}

return head;

}

//마지막 노드에 insert하는 함수

ListNode\* insert\_last(ListNode\* head, element data){

ListNode\* node = (ListNode\*)malloc(sizeof(ListNode));

node->data = data;

//head가 NULL이면 head에 insert

if (head == NULL) {

head = node;

node->link = head;

}

//head가 있으면

else {

node->link = head->link; //node->link를 첫번째 노드로 연결

head->link = node; // head의 포인터를 노드에 연결하여 노드를 첫번째로 변경

head = node; //노드를 head로 전환(마지막 노드로 전환?)

}

return head;

}

// 첫번째 노드를 삭제하는 함수

ListNode\* delete\_first(ListNode\* head) {

ListNode\* node = (ListNode\*)malloc(sizeof(ListNode));

if (head == NULL) return head;

//head가 있으면

node = head->link;// 첫번째 노드 들고오기

head->link = node->link; // 마지막 노드를 두번째 노드와 연결시켜 첫번째 노드 분리하기

free(node); // 첫번째 노드 삭제

return head;

}

//마지막 노드를 삭제하는 함수

ListNode\* delete\_last(ListNode\* head) {

ListNode\* node = (ListNode\*)malloc(sizeof(ListNode));

if (head == NULL) return head;

//head가 있으면

ListNode\* p;

if (head == NULL) return head;

// 마지막 직전의 노드로 이동(마지막 직전의 노드 들고오기)

p = head->link;

do {

p = p->link;

} while (p->link != head);

node = p->link; // 마지막 노드

p->link = head->link; // 마지막 노드 분리

head = p; //head를 뒤로 전환

free(node); // 마지막 노드 삭제

return head;

}

int main() {

ListNode\* head = NULL;

for (int i = 0; i < 5; i++) {

head = insert\_last(head, i \* 10);

print\_list(head);

}

head = delete\_first(head);

print\_list(head);

head = delete\_last(head);

print\_list(head);

free(head);

return 0;

}

1. 실행화면

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 고찰

처음 ADT를 만들때는 head의 위치를 고려하지 않아 많이 어려웠습니다. 하지만 head의 위치가 마지막 노드에 위치한다는 것을 알고 head의 위치를 고려하여 구현하였습니다.