<자료구조>

4장 연습문제



|  |  |
| --- | --- |
| 학부 | 컴퓨터공학과 |
| 학번 | 20171327 |
| 이름 | 조민아 |

▪ 교재 4.2절 Exercises의 8번 문제 – Source Code

#include <stdio.h>

#include <stdlib.h>

typedef struct listNode\* listPointer;

typedef struct listNode {

int data;

listPointer link;

}listNode;

void createNode(listPointer\* r);

void Rmove(listPointer \*l, listPointer \*r,int m);

void Lmove(listPointer \*l, listPointer \*r,int m);

void printList(listPointer r,listPointer l);

int main() {

int num=0; //이동에 필요한 노드 수

listPointer left = NULL, right = NULL;

int select\_num=0;

while (1) {

printf("\*\*\*\*\*\*\*\*\*\*메뉴\*\*\*\*\*\*\*\*\*\*\n");

printf("1) 리스트 추가\n");

printf("2) 오른쪽으로 이동\n");

printf("3) 왼쪽으로 이동\n");

printf("4) 리스트 출력\n");

printf("5) 프로그램 종료\n");

printf("메뉴를 선택하세요: \n");

scanf\_s("%d", &select\_num,sizeof(select\_num));

switch (select\_num)

{

case 1:

createNode(&right);

break;

case 2:

printf("오른쪽으로 몇 칸 이동하겠습니까? :");

scanf\_s("%d", &num,sizeof(num));

Rmove(&left, &right, num);

break;

case 3:

printf("왼쪽으로 몇 칸 이동하겠습니까? :");

scanf\_s("%d", &num,sizeof(num));

Rmove(&left, &right, num);

break;

case 4:

printList(right,left);

break;

case 5:

exit(0);

}

}

}

void createNode(listPointer\* r) {

int num; //노드 데이터 값 입력받는 변수

int listnum; //추가할 리스트의 개수

printf("몇 개의 리스트를 추가하겠습니까?: ");

scanf\_s("%d", &listnum,sizeof(listnum));

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

listPointer node = (listPointer)malloc(sizeof(listNode));

printf("리스트의 데이터를 입력하세요: ");

scanf\_s("%d", &num,sizeof(num));

node->data = num;

node->link = \*r; //노드를 리스트의 오른쪽에 넣어준 후,

\*r = node; //리스트를 가리키는 \*r 포인터를 리스트의 맨 앞으로 옮겨준다

}

}

void Rmove(listPointer\* l, listPointer\* r, int n) {

listPointer temp = NULL; //오른쪽 이동 시 현재의 위치를 나타내는 포인터

for (; n > 0 && \*r != NULL; n--) {

temp = (\*r)->link; //temp에 현재 위치의 오른쪽 노드를 저장

(\*r)->link = \*l; //\*r의 link에 \*l의 위치를 넣어줌

\*l = \*r; //\*l의 위치에 \*r을 넣어줌

\*r = temp; //\*r의 위치에 \*r 다음의 리스트를 넣어줌

}

}

void Lmove(listPointer \* l, listPointer \* r, int n) {

listPointer temp = NULL;

for (; n > 0 && \*l != NULL; n--) {

temp = (\*l)->link; //temp에 현재 위치의 왼쪽 노드를 저장

(\*l)->link = \*r; //\*l의 link에 \*r의 위치를 넣어줌

\*r = \*l; //\*r의 위치에 \*l을 넣어줌

\*l = temp;//\*l의 위치에 \*l 다음의 리스트를 넣어줌

}

}

void printList(listPointer r,listPointer l) {

printf("오른쪽 리스트: ");

for (; r; r = r->link) {

printf("%4d", r->data);

}

printf("\n");

printf("왼쪽 리스트: ");

for (;l; l = l->link) {

printf("%4d", l->data);

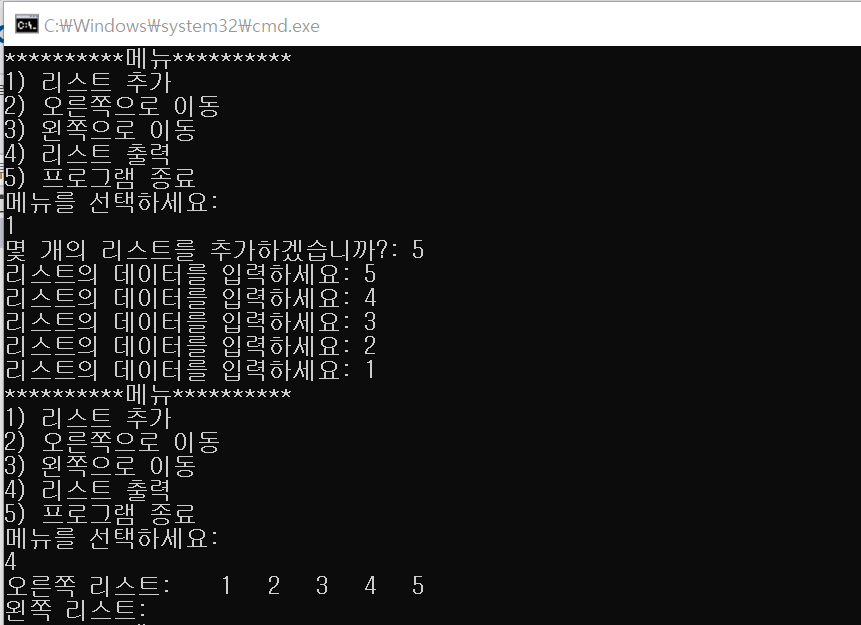
}

printf("\n");

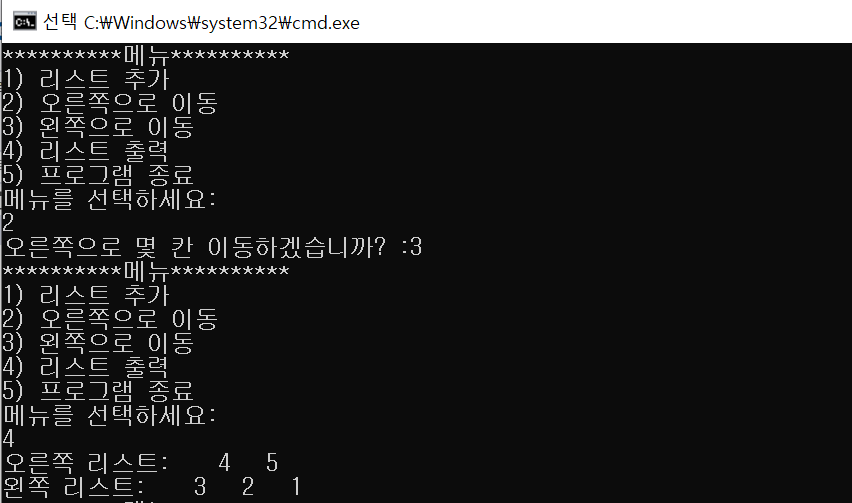
}

▪ 실행화면

* 리스트 추가 & 결과 출력



* 리스트 오른쪽으로 이동 & 결과 출력



* 리스트 왼쪽으로 이동 & 결과 출력

