

# 실습과제 4.1.2

## [ 과제 설명 ]

기존의 원형 연결리스트는 삽입할 때, 맨 마지막의 노드까지 탐색하기 위해 반복문이 쓰였다. 따라서 원형 연결리스트를 구성하고 있는 노드의 수가  $n$ 일 경우에 수행시간은  $O(n)$ 이 된다. 수행시간을 개선하기 위해, 기존에 head가 첫 노드를 가리키고 있는 것을 tail이 마지막 노드를 가리키고 있는 것으로 개선한다. 마지막 노드를 가리키고 있으면, 탐색에 필요한 시간을 줄일 수 있다. 출력할 때는 원형이기 때문에 마지막 노드가 가리키고 있는 것이 첫 노드라서 첫 노드를 찾기 위해 따로 탐색할 필요가 없어진다.

## [ 코드 ]

헤더 파일을 분리할 정도로 함수를 많이 사용하지 않았기도 하고 보고서를 작성할 때 필요 이상으로 길어지는 것을 방지하기 위해 이번 실습에 쓰지 않은 함수를 제외하고 한 코드로 첨부한다.

또한 이전 실습에서도 사용한 함수는 주석을 생략한다.

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
    int data;
    struct Node* link;
}Node;

Node* Create_Node(int newData) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    newNode->data = newData;
    newNode->link = NULL;

    return newNode;
}

void Append_Node_T(Node** tail, Node* newNode) {
    if ((*tail) == NULL) {
        //현재의 tail이 비어있으면, 새로운 노드를 tail로 설정
```

```

    *tail = newNode;
    (*tail)->link = newNode;
}
else {
    //비어있지 않다면, 가장 끝부분에 새로운 노드를 추가하고 tail을 이동시킨다.
    newNode->link = (*tail)->link;
    //새 노드의 link에 tail이 가리키고 있던 노드를 가리키게 한다.
    (*tail)->link = newNode;
    (*tail) = newNode;
}
}

void Print_Circluar_Linked_List_T(Node* tail) {
    //tail을 기준으로 만들었기 때문에 기존 출력 함수와는 조금 다르다.
    if (tail == NULL) return; //tail이 비어있으면, 함수 종료

    Node* iter = tail->link;
    //tail 다음이 첫 노드기 때문에 tail 다음부터 시작
    int i = 0;

    do {
        printf("node[%d]: %d", i, iter->data);
        iter = iter->link;
        if (iter != NULL) printf(" -> ");
        i++;
    } while (iter != tail->link);
    //원점으로 돌아올 때까지 반복

    printf("head: %d", iter->data);
    //마지막 노드가 head로 연결되었는지 확인
    puts("");
}

int main(void) {
    Node* tail = NULL;

    //원형 연결리스트 구성
    Append_Node_T(&tail, Create_Node(15));
    Append_Node_T(&tail, Create_Node(25));
    Append_Node_T(&tail, Create_Node(31));
    Append_Node_T(&tail, Create_Node(24));

    //원형 연결리스트 출력
    Print_Circluar_Linked_List_T(tail);
}

```

## [ 실행 결과 ]

```
Microsoft Visual Studio 디버그 × + v
node[0]: 15 -> node[1]: 25 -> node[2]: 31 -> node[3]: 24 -> head: 15
C:\Users\seowo\OneDrive\바탕 화면\학교 문서\2학년_1학기\자료구조_예제_실
4개)이(가) 종료되었습니다(코드: 0개).
이 창을 닫으려면 아무 키나 누르세요...
```

### [ 과제에 대한 고찰 ]

이전에 일반 연결리스트를 구현했을 때는 계속 head로만 구현해봐서, 마지막 노드를 가리키는 tail을 사용하는 건 다소 생소하지만 원형인 경우에는 이런 것도 효율적이라는 새로운 방식을 알게 되어서 좋았다. 또한, head에서 tail로 변경해서 그런지 출력하는 함수가 달라졌는데, head를 가리킬 때와 tail을 가리킬 때 둘 다 한 함수로 구현할 수 있는 방법은 없는지 살펴보는 것도 괜찮을 것 같다는 생각이 들었다.