

원형리스트 프로그램

5671144 강범창

1.1 원형 리스트를 가지고 메뉴로 조작하는 프로그램을 작성하기 위해 리스트에 사용할 구조체와 element를 선언합니다.

```
typedef char* element;

typedef struct
{
    element data;
    struct ListNode* link;
} ListNode;
```

1.2 원형리스트를 구현합니다.

insert first 함수입니다. 리스트의 head 와 넣을 data를 받으며, 가장 앞에 값을 넣는(head 바로 다음) 함수이기에 head가 없다면 새로 만든 node 를 head 로 만들어 주며, 아니라면 새롭게 만든 node 의 link에 head 의 link를 넣어 줌으로 head 가 가리키고 있던 곳을 가리키게 되며, 이후 head 가 새로운 node 를 가리키게 함으로 head 바로 뒤 즉 바로 앞에 값을 넣을 수 있게 해줍니다.

```
ListNode* insertFirst(ListNode* head, element data)
{
    ListNode* node = (ListNode*)malloc(sizeof(ListNode));
    node->data = data;
    if (head == NULL) {
        head = node;
        node->link = head;
    }
    else {
```

```

        node->link = head->link;
        head->link = node;
    }
    return head;
}

```

InsertLast 함수입니다. Head와 넣을 값인 data를 입력 받으며, 위의 함수와 달리 원형 큐의 마지막에 값을 넣는 것을 구현하기 위해 새로운 노드를 만든 다음 이 노드의 다음을 head 가 가리키고 있던 것을 가리키게 하여 한 칸을 미루게 됩니다. 이후 head의 link 가 node를 가리키게 함으로 가장 뒤에 값을 넣어줍니다. 이후 head를 node 로 변경해주면서 가장 원형 큐에서 가장 뒤에 있게 되는 값인 head를 교체해 줍니다. (head 가 가장 마지막에 있다 생각하고 그것으로 구현).

```

ListNode* insertLast(ListNode* head, element data)
{
    ListNode* node = (ListNode*)malloc(sizeof(ListNode));
    node->data = data;
    if (head == NULL) {
        head = node;
        node->link = head;
    }
    else {
        node->link = head->link;
        head->link = node;
        head = node;
    }
    return head;
}

```

printList 함수입니다. Head를 입력 받으며, head에서부터 head의 마지막까지 출력합니다. 조작은 p라는 ListNode를 만들어 받은 head를 초기값으로 준 다음, 다음위치를 계속 저장하고 다음위치를 가는 식으로 진행이 됩니다.

```

void printList(ListNode* head) {

```

```

ListNode* p;

if (head == NULL) return;

p = head->link;
if (head->link == head) {
    printf("%d->\n", p->data);
    return;
}
do {
    printf("%d-> ", p->data);
    p = p->link;
} while (p != head);

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

```

Delete first 함수입니다. Head를 받아 가장 앞에 있는 node를 선택한후 ListNode를 선언하여 담아 둔 다음 이 node가 가리키고 있던 node를 새로운 node라고 하고 head가 이것을 가리키게 함으로 가장 앞에 있는 노드를 삭제하는 것을 구현합니다.

```

ListNode* deleteFirst(ListNode* head) {
    if (head == NULL) {
        printf("리스트가 비어있습니다.\n");
        return head;
    }

    if (head->link == head) {
        head->link = NULL;
        return head;
    }
    else {
        ListNode* removed = head->link;

```

```

        ListNode* newFrist = removed->link;
        free(removed);
        head->link = newFrist;
        return head;
    }
}

```

DeleteLast 함수입니다. 리스트의 head를 입력 받으며 비어 있거나 head가 자기 자신을 가르키고 있는 경우는 head를 바로 return 합니다. 아니라면 마지막에 있는 노드인 head의 값을 없애고 그 이전의 노드를 head로 만들기위해 head의 이전 노드를 찾습니다 이후 만약 지웠을 때 혼자 남을 경우에는 자기 자신을 이어주고 아니라면 그 이전의 노드가 head가 가리키고 있던 곳을 가리키게 한 후 그 이전 노드를 반환함으로 head를 변경합니다.

```

ListNode* deleteLast(ListNode* head) {
    if (head == NULL) {
        printf("리스트가 비어있습니다.\n");
        return head;
    }

    if (head->link == head) {
        head->link = NULL;
        return head;
    }
    else {
        ListNode* removePrevNode = head->link;
        while (1)
        {
            if (removePrevNode->link == head) {
                break;
            }
            else {
                removePrevNode = removePrevNode->link;
            }
        }
    }
}

```

```

        if (head->link == removePrevNode) {
            removePrevNode->link = removePrevNode;
            return removePrevNode;
        }
        else {
            removePrevNode->link = head->link;
            return removePrevNode;
        }
    }
}

```

1.3 메인에서 입력을 받아 처리하는 부분입니다.

1의 경우 – 숫자(data)를 입력 받아 insert Frist 함수를 실행시킵니다

2의 경우 – 숫자(data)를 입력 받아 insert Last 함수를 실행시킵니다.

3의 경우 – delete Fast 함수를 실행시킵니다. 이후 삭제됐다는 메시지를 출력합니다.

4의 경우 – delete Last 함수를 실행시킵니다. 이후 삭제됐다는 메시지를 출력합니다.

0의 경우 – 프로그램을 종료합니다.

나머지의 경우 – 따로 처리를 하지 않고 다시 메뉴를 출력합니다.

```

ListNode* head = NULL;

int seletedMenu;
int inputNum;
while (1)
{
    printf("***** Menu *****\n");
    printf("(1) Insert Frist\n");
    printf("(2) Insert Last\n");
}

```

```
printf("(3) Delte FristWn");
printf("(4) Delte LastWn");
printf("(5) Print ListWn");
printf("(0) ExitWn");
printf("Enter the menu: ");
scanf("%d", &seletedMenu);
switch (seletedMenu)
{
case 0:
    printf("Exit the program.Wn");
    exit(0);
    break;
case 1:
    printf("Input a number: ");
    scanf("%d", &inputNum);
    head = insertFirst(head, inputNum);
    break;
case 2:
    printf("Input a number: ");
    scanf("%d", &inputNum);
    head = insertLast(head, inputNum);
    break;
case 3:
    head = deleteFirst(head);
    printf("First node has been deleted.Wn");
    break;
case 4:
    head = deleteLast(head);
    printf("Last node has been deleted.Wn");
    break;
case 5:
    printList(head);
    break;
default:
    break;
```

```

    }
    printf("Wn");
}
}

```

아래는 실행 결과입니다.

<https://youtu.be/poK2x9RAn00>

