

단순 연결 리스트 프로그램

5671144 강범창

1. 단순 연결리스트를 메뉴로 조작하는 프로그램을 작성한다.

1.1 단순 연결리스트를 메뉴로 조작하는 프로그램을 작성하기 위해 리스트에 사용할 구조체와 `element`를 선언합니다.

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

1.2 단순 연결리스트를 구현합니다.

`insert first` 함수입니다. 연결리스트의 `head`에 값을 넣어 가장 앞에 넣는 것을 구현합니다.

```
ListNode* insertFirst(ListNode* head, int item)
{
    ListNode* node = (ListNode*)malloc(sizeof(ListNode));
    node->data = item;
    node->link = head;
    head = node;
    return head;
}
```

Insert 함수입니다. Head 와 이전의 node추가할 값을 받아와서 사용하며, 이전의 node 에 새로 만든 node 를 넣어 이전에 바꾸자 한 node에 위치에 값을 넣어줍니다.

```
ListNode* insert(ListNode* head, ListNode* pre, element item)
{
    ListNode* node = (ListNode*)malloc(sizeof(ListNode));

    node->data = item;
    node->link = pre->link;
    pre->link = node;
    return head;
}
```

Delete first 함수입니다. Head를 받아 가장 앞에 있는 node 를 삭제하고 그 다음 연결되어 있는 link를 넣어뒤 리스트의 가장 앞에 것을 삭제하는 것을 구현합니다.

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

Delete 함수입니다. Head 와 바꿀 위치에 있는 node , pre를 가져와 사용하고, pre 에 있는 값들을 지우고 다음위치에 있는 node를 연결 시켜줍니다.

```

ListNode* delete(ListNode* head, ListNode* pre)
{
    ListNode* removed;
    removed = pre->link;
    pre->link = removed->link;
    free(removed);
    return head;
}

```

printList 함수입니다. Head 를 받아 head에서부터 마지막 노드까지 출력을합니다.

```

void printList(ListNode* head)
{
    printf("List: ");
    for (ListNode* p = head; p != NULL; p = p->link)
    {
        printf("%d->", p->data);
    }
    printf("\n");
}

```

1.3 메인에서 입력을 받아 처리하는 부분입니다.

입력을 받아 0라면 프로그램을 종료하고, 아니라면 그에 맞는 메뉴를 조건에 맞춰 실행 시켜줍니다.

0의 경우 - 프로그램을 종료한다는 메시지와 종료를 합니다.

1의 경우 - insert 할 값과 위치를 입력 받고 head가 없거나 0인경우에 는 그에 맞는 insertfirst를 통해 가장 앞에 값을 넣어줍니다. 그것이 아니라면 pre 의 위치를 찾아 그 위치에 값을 추가하고 이후 찾을 때 사용된 횟수를 출력합니다.

2의 경우 - delete 를 수행을 하고, head가 없을 경우 리스트가 비어 있다는 메시지를 출력하고 메뉴를 다시 수행합니다. Head 가 있을 경우 지울 위치를 입력을

받고, 0일경우 가장 앞에 값을, 아니라면 이전과 동일한 방식으로 pre의 위치를 찾아 삭제를 하고 삭제를 위해 돈 횟수를 출력합니다.

3의 경우 - 리스트를 출력합니다.

나머지의 경우 - 메뉴에 포함되지 않는 번호라는 메시지를 출력하고 다시 메뉴를 실행시킵니다.

```
int main(void)
{
    int menu;
    int num1, num2;

    ListNode* head = NULL;
    while (1)
    {
        printf("Menu\n");
        printf("(1) Insert\n");
        printf("(2) Delete\n");
        printf("(3) Print\n");
        printf("(0) Exit\n");
        printf("Enter the menu: ");
        scanf("%d", &menu);

        switch (menu)
        {
            case 0:
                printf("Exit the Program.\n");
                exit(1);
                break;
            case 1:
                printf("Enter the number and position: ");
                scanf("%d %d", &num1, &num2);
                if (head == NULL)
```

```

    {
        printf("List is empty. Insert at position 0..\n\n");
        head = insertFirst(head, num1);
        // 처음에는 움직이지 않고 입력이 가능하기에
0만을 출력합니다.

        printf("Move along the link: 0\n");
    }
    else if (num2 == 0) {
        head = insertFirst(head, num1);
        printf("Move along the link: 0\n");
    }
    else
    {
        ListNode* pre = head;
        ListNode* lastNode = NULL;
        int lastIndex = 0;

        while (pre != NULL && lastIndex < num2)
        {
            lastNode = pre;
            pre = pre->link;
            lastIndex++;
        }

        if (num2 >= lastIndex)
        {
            printf("The last index is %d. Insert at the
end of the list.\n\n", lastIndex);

            head = insert(head, lastNode, num1);
            printf("Move along the link: %d\n",
lastIndex);
        }
    }

```

```

    }
    break;
case 2:
    if (head == NULL)
    {
        printf("List is empty.\n");
        break;
    }
    ListNode* pre = head;
    ListNode* lastNode = NULL;
    int lastIndex = 0;

    printf("Enter the position to delete: ");
    scanf("%d", &num1);
    if (num1 == 0) {
        head = deleteFirst(head, num1);
        printf("\nMove along the link: 0\n");
    }
    else {
        while (pre!=NULL && lastIndex < num1)
        {
            lastNode = pre;
            pre = pre->link;
            lastIndex++;
        }
        head = delete(head, lastNode);
        printf("\nMove along the link: %d\n" , lastIndex);
    }
    break;
case 3:
    printList(head);
    break;

```

```
default:
    printf("Invalid Menu. Please select again..\n");
    break;
}

printf("\n");
```

아래는 실행 결과입니다.

스크린샷으로 담기에는 너무 보기가 힘들어져, 영상으로 대체하였습니다.

<https://youtu.be/dp5PyKlQy1A>

