

데이터구조론_HW_1

: student의 정보를 linked list로 실습하여 구현하기



학번: 2020231144

소속: 경제학과

이름: 이 정 우

과제_1

과제는 Visual Studio(버전 2019 또는 2022 권장)로 작성되어야 합니다.
프로젝트를 만들어 작업하세요.

과제_1 은 첨부된 파일로 대체합니다.

과제_2

- 파일을 읽어 Linked list에 데이터를 입력하세요

```
// (과제 2) 파일을 읽어 Linked list 데이터 삽입
// 출처: https://blockdmask.tistory.com/392 코드 참고
FILE* student_pFile= fopen("student.txt", "r");
if(student_pFile==NULL){
    printf("파일이 없습니다. 프로그램을 종료합니다.");
    return 0;
}
else {
    printf("student파일을 읽었습니다.\n");
}
FILE* gpa_pFile = fopen("gpa.txt", "r");
if (gpa_pFile== NULL) {
    printf("파일이 없습니다. 프로그램을 종료합니다.");
    return 0;
}
else {
    printf("gpa파일을 읽었습니다.\n");
}

// student, gpa 파일 받아와서 linked list에 넣어야겠지?
while(!feof(student_pFile) && !feof(gpa_pFile)) { //feof 파일 끝에 도달 > !feof 파일끝 도달이
아님 > 두 파일 다 끝 도달이 아니라면 계속 진행
    char tmp_name[30];
    int tmp_id=0;
    char tmp_phone[20];
    char tmp_province[30];
    float tmp_gpa;

    fscanf(student_pFile, "%s %d %s %s", tmp_name, &tmp_id, tmp_phone, tmp_province);
    fscanf(gpa_pFile, "%f", &tmp_gpa);

    listData newData;
    strcpy(newData.name, tmp_name);
    newData.id = tmp_id;
    strcpy(newData.phone, tmp_phone);
    strcpy(newData.province, tmp_province);
    newData.gpa = tmp_gpa;
```

```

        insertLast(L, newData);
    }
    fclose(student_pFile);

```

과제_3

- 코드에서 아래의 함수는 실제로 구현되어 있지 않습니다. 코드가 정상적으로 작동할 수 있도록 내용을 채워 넣으세요

INSERT , INSERT_FIRST , INSERT_LAST

```

void insert(linkedList* L, listNode* pre, listData x) {
    //빈 노드 먼저 생성
    listNode* newNode;
    newNode = (listNode*)malloc(sizeof(listNode));
    newNode->data = x;

    //case 1) list가 비어있는 경우
    if (L->head == NULL) {
        newNode->link = NULL;
        L->head = newNode;
    }
    //case 2) list의 첫번째에 넣는 경우
    else if (pre == NULL) {
        newNode->link = L->head;
        L->head = newNode;
    }
    //case 3) 일반적으로 사용되는 경우 (중간)
    else {
        newNode->link = pre->link;
        pre->link = newNode;
    }
    //if문들 탈출하면 L길이 하나 늘리기
    L->length++;
}

void insertFirst(linkedList* L, listData x) {
    listNode* newNode;
    newNode = (listNode*)malloc(sizeof(listNode));
    // newNode의 data값과 link값 넣기!! 위에 case 2 참고
    newNode->data = x;
    newNode->link = L->head;
    L->head = newNode;
    L->length++;
}

```

```

void insertLast(linkedList* L, listData x) {
    listNode* newNode, * temp;
    newNode = (listNode*)malloc(sizeof(listNode));
    // newNode의 data 와 link 지정 마지막에 넣는거 고정이니까 link = NULL
    newNode->data = x;
    newNode->link = NULL;

    //case 1) list가 비어있는 경우
    if (L->head == NULL) {
        L->head = newNode;
    }
    //case 2) 일반적인 경우
    else {
        temp = L->head;
        //temp가 끝(NULL)에 도달하지 않는다면 앞으로 앞으로~
        while (temp->link != NULL)
            temp = temp->link;
        temp->link = newNode;
    }
    L->length++;
}

```

DELETE

```

int delete(linkedList* L, listNode* p) {
    listNode* pre;

    // case 1) list가 비어있는 경우
    if (L->head == NULL) return FALSE;
    // case 2) 찾는 값이 list에 없는 경우
    if (p == NULL) return FALSE;
    // case 3) 찾는 값이 첫번째인 경우 > head를 p link로 옮기고 비워주기
    if (L->head == p) {
        L->head = p->link;
        free(p);
    }
    else { // case 4) 찾는 값이 첫번째가 아닌 경우 > 일반적인 경우
        pre = L->head;
        while (pre->link != p) { //이전 노드가 향하고 있는 것이 p가 아니라면 앞으로
            pre = pre->link;
            if (pre == NULL) return FALSE; //NULL 즉 끝에 도달하면 FALSE --> CASE
        }
        pre->link = p->link;
        free(p);
    }
    L->length--;
    return TRUE;
}

```

전진

2랑 같은거 아닌가?

SEARCH

```
listNode* search(linkedList* L, int x) {
    //temp를 설정 후 head로 정의
    listNode* temp = L->head;
    // int x값을 id라고 가정하고 설정
    while (temp != NULL) { //찾으면 return 못찾으면 전진
        if (temp->data.id == x)
            return temp;
        else
            temp = temp->link;
    }
    return temp; //이건 NULL이라면 NULL값 출력하라고 있는건가?
}
```

CLEAR

```
void clear(linkedList* L) {
    // free 써야되는거같음
    listNode* temp = L->head;
    listNode* link_bridge; //temp의 다음 다리 연결용

    if (temp != NULL) {
        link_bridge = temp->link; //중간 다리에 temp_link값 잠시 넣어두고
        free(temp);               //temp free 시키고
        temp = link_bridge;        //temp에 아까 보관해둔 다리 다시 부르기
    }
    L->length = 0;
}
```

REVERSE

```
void reverse(linkedList* L) {
    listNode *prev, *curr, *next; //prev 전꺼 가리키기, curr 중간꺼, next 다음꺼 길
    //앞지않게 미리 보내기
    prev = NULL;
    curr = L->head;

    while (curr != NULL) {
        next = curr->link;
        curr->link = prev;
        prev = curr;
        curr = next;
    }
    L->head = prev;
}
```

과제_4

'과제 2' 및 '과제 3'에서 완성된 코드를 실행 후 명령문을 입력하여 작업을 수행 하세요.

print

```
student파일을 읽었습니다.  
gpa파일을 읽었습니다.  
명령어를 입력하세요: print  
L=([김연세 2023312151 010-3542-6538 경남 4.20]  
[홍길동 2020253013 010-6451-3876 경기 4.30]  
[박연세 2020245001 010-4653-2222 충북 3.10]  
[고길동 2019253004 010-6453-1111 경기 3.70]  
[장연세 2022243007 010-7897-7897 강원 2.80]  
[최연세 2024253003 010-9871-2542 서울 3.50]  
[장홍길 2022245081 010-5234-3257 강원 3.80]  
[김태현 2022253078 010-8764-7899 전남 3.50]  
[고정규 2018253088 010-3977-4762 제주 3.80]  
[한민승 2014253008 010-4797-7878 경남 2.80]  
[박상화 2020245006 010-6534-3452 전남 3.30]  
[한재원 2023265120 010-5656-8989 서울 3.60]  
[진상현 2020245150 010-7899-8777 전북 3.10]  
[우시현 2024192008 010-7988-1466 경북 3.80]  
)
```

search 2018253088

```
명령어를 입력하세요: search 2018253088  
2018253088 노드를 찾았습니다. 이름: 고정규  
명령어를 입력하세요:
```

insert 2018253088, 2023324319, 김정규, 010-6625-2129, 제주, 4.2

```
명령어를 입력하세요: insert 2018253088, 2023324319, 김정규, 010-6625-2129, 제주, 4.2  
2018253088 노드를 찾았습니다. 이름: 고정규  
L=([김연세 2023312151 010-3542-6538 경남 4.20]  
[홍길동 2020253013 010-6451-3876 경기 4.30]  
[박연세 2020245001 010-4653-2222 충북 3.10]  
[고길동 2019253004 010-6453-1111 경기 3.70]  
[장연세 2022243007 010-7897-7897 강원 2.80]  
[최연세 2024253003 010-9871-2542 서울 3.50]  
[장홍길 2022245081 010-5234-3257 강원 3.80]  
[김태현 2022253078 010-8764-7899 전남 3.50]  
[고정규 2018253088 010-3977-4762 제주 3.80]  
[김정규, 2023324319 010-6625-2129, 제주, 4.20]  
[한민승 2014253008 010-4797-7878 경남 2.80]  
[박상화 2020245006 010-6534-3452 전남 3.30]  
[한재원 2023265120 010-5656-8989 서울 3.60]  
[진상현 2020245150 010-7899-8777 전북 3.10]  
[우시현 2024192008 010-7988-1466 경북 3.80]  
)  
명령어를 입력하세요: 
```

delete 2023324319

```
명령어를 입력하세요: delete 2023324319
2023324319 노드를 찾았습니다. 이름: 김정규,
성공적으로 삭제하였습니다.
L=([우시현 2024192008 010-7988-1466 경북 3.80]
[진상현 2020245150 010-7899-8777 전북 3.10]
[한재원 2023265120 010-5656-8989 서울 3.60]
[박상화 2020245006 010-6534-3452 전남 3.30]
[한민승 2014253008 010-4797-7878 경남 2.80]
[고정규 2018253088 010-3977-4762 제주 3.80]
[김태현 2022253078 010-8764-7899 전남 3.50]
[장홍길 2022245081 010-5234-3257 강원 3.80]
[최연세 2024253003 010-9871-2542 서울 3.50]
[장연세 2022243007 010-7897-7897 강원 2.80]
[고길동 2019253004 010-6453-1111 경기 3.70]
[박연세 2020245001 010-4653-2222 충북 3.10]
[홍길동 2020253013 010-6451-3876 경기 4.30]
[김연세 2023312151 010-3542-6538 경남 4.20]
```

Reverse

```
명령어를 입력하세요: reverse
L=([김연세 2023312151 010-3542-6538 경남 4.20]
[홍길동 2020253013 010-6451-3876 경기 4.30]
[박연세 2020245001 010-4653-2222 충북 3.10]
[고길동 2019253004 010-6453-1111 경기 3.70]
[장연세 2022243007 010-7897-7897 강원 2.80]
[최연세 2024253003 010-9871-2542 서울 3.50]
[장홍길 2022245081 010-5234-3257 강원 3.80]
[김태현 2022253078 010-8764-7899 전남 3.50]
[고정규 2018253088 010-3977-4762 제주 3.80]
[한민승 2014253008 010-4797-7878 경남 2.80]
[박상화 2020245006 010-6534-3452 전남 3.30]
[한재원 2023265120 010-5656-8989 서울 3.60]
[진상현 2020245150 010-7899-8777 전북 3.10]
[우시현 2024192008 010-7988-1466 경북 3.80]
)
```

getLength

```
명령어를 입력하세요: getLength
리스트에 저장된 데이터 개수: 14
```

Exit

```
리스트에 저장된 데이터 개수: 14
명령어를 입력하세요: exit

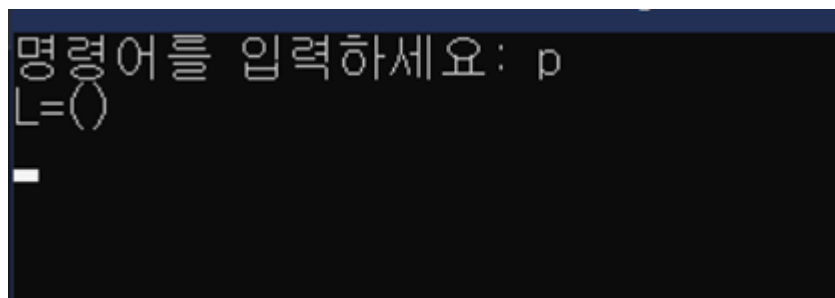
C:\#24_1#HW1#x64#Debug#HW1.exe(프로세스 6168개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면
자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...■
```

과제_5

- Case) 파일은 읽은 것 같은데 왜 list에 들어가지 않을까?

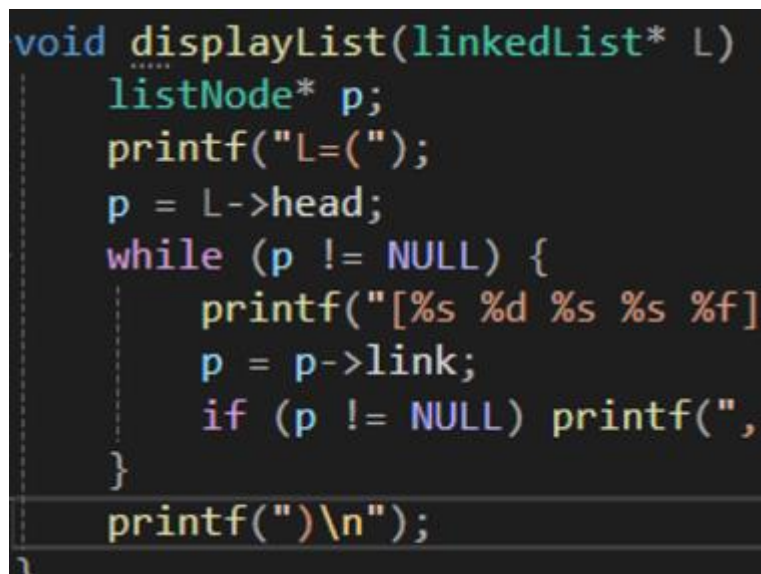
Reverse의 함수는 어려워서 주석처리를 하고 데이터를 먼저 넣어보고자 했습니다.

하지만 결과값이 계속 이렇게 나왔습니다. 처음에는 displayList에 문제가 있다고 생각했습니다.



```
명령어를 입력하세요: p
L=()
_
```

이를 보니 displayList code에서 while문 안에서의 문제라는 것을 알아냈습니다.



```
void displayList(linkedList* L) {
    ....
    listNode* p;
    printf("L=(");
    p = L->head;
    while (p != NULL) {
        printf("[%s %d %s %s %f]",
            p->data, p->count, p->link, p->next, p->val);
        p = p->link;
        if (p != NULL) printf(", ");
    }
    printf(")\n");
}
```

만약 L=(과)는 잘 수행되는 것을 보니 while 문의 문제를 확실해졌습니다.


```
// (과제 2) 파일을 읽어 Linked list 데이터 삽입
// 출처: https://blockdmask.tistory.com/392 코드 참고
FILE* student_pFile= fopen("student.txt", "r");
if(student_pFile==NULL){
    printf("파일이 없습니다. 프로그램을 종료합니다.");
    return 0;
}
else {
    printf("student파일을 읽었습니다.\n");
}
FILE* gpa_pFile = fopen("gpa.txt", "r");
if (gpa_pFile== NULL) {
    printf("파일이 없습니다. 프로그램을 종료합니다.");
    return 0;
}
else {
    printf("gpa파일을 읽었습니다.\n");
}
```

student파일을 읽었습니다.
gpa파일을 읽었습니다.
명령어를 입력하세요:

While문 안에서 출력이 안되는 것을 보면 L이 비어 있다는 의심을 하게 되었습니다.

그래서 파일이 잘 열어지는지 확인하기 위해서 fopen 함수 뒤에 파일이 없는 경우 뿐 아니라 있는 경우에는 읽었다고 printf 를 통해 나타냈습니다.

파일이 잘 읽혀지는 것은 확인했습니다.

```
// student, gpa 파일 받아와서 linked list에 넣어주겠다?
for (int i = 0; i < L->length; i++) {
    char tmp_name[30];
    int tmp_id=0;
    char tmp_phone[20];
    char tmp_province[30];
    float tmp_gpa;
    fscanf(student_pfile, "%s %d %s %s", tmp_name, &tmp_id, tmp_phone, tmp_province);
    fscanf(gpa_pfile, "%f", &tmp_gpa);
    listData newData;
    strcpy(newData.name, tmp_name);
    newData.id =tmp_id;
    strcpy(newData.phone, tmp_phone);
    strcpy(newData.province, tmp_province);
    newData.gpa = tmp_gpa;

    printf("Name: %s, ID: %d, Phone: %s, Province: %s, GPA: %f\n", newData.name, newData.id, newData.phone, newData.province, newData.gpa);
    insertLast(L,newData);
}
```

그러면 linked list 안에 넣는 제 함수에 문제가 있는 것을 알았습니다.

fscanf에서 변수 설정의 문제인지를 알기 위해 fscanf 이후 printf를 통해 확인했습니다.

하지만 아무 변화가 없었습니다. 그래서 for의 문제를 의심하여 제거하고 fscanf와 printf를 했습니다.

```
float tmp_gpa;

fscanf(student_pFile, "%s %d %s %s", tmp_name, &tmp_id, tmp_phone, tmp_province);
printf("Name: %s, ID: %d, Phone: %s, Province: %s\n", tmp_name, tmp_id, tmp_phone, tmp_province);
```

```
C:\> 선택 C:\W24_1\HW1\HW1\Debug\HW1.exe
student파일을 읽었습니다.
gpa파일을 읽었습니다.
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
명령어를 입력하세요: _
```

한 줄이 출력되는 것을 확인한 뒤 for을 수정했습니다.

```
for (int i = 0; i < 14 ; i++){
    char tmp_name[30];
    int tmp_id=0;
    char tmp_phone[20];
    char tmp_province[30];
    float tmp_gpa;

    fscanf(student_pFile, "%s %d %s %s",
    printf("Name: %s, ID: %d, Phone: %s,
```

```
student파일을 읽었습니다.
gpa파일을 읽었습니다.
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
Name: 김연세, ID: 2023312151, Phone: 010-3542-6538, Province: 경남
명령어를 입력하세요: _
```

For의 조건부가 length로 되어있어서 문제인 것 같아서 정수인 14로 교체하니 잘 작동 하였습니다.

하지만 위 student txt에는 무슨 값이 들어갈지 몰라야 합니다. 이번 과제는 제가 셀 수 있었기 때문에 14를 넣었지만 저 곳을 변수로 놓기 위해 노력했습니다.

```
// student, gpa 파일 받아와서 linked list에 넣어야겠지?
while(!feof(student_pFile) && !feof(gpa_pFile)) { //feof
    char tmp name[30];
```

다양한 방식을 시도했지만 결국에는 feof라는 기능에 not을 붙여서 교집합을 만들었습니다. Student파일도 끝이 아니어야 하며 GPA파일도 끝이 아니어야만 계속 진행되도록 했습니다. 이렇게 마무리하게 되었습니다.

과제를 하기 전 공부를 하며 내용은 이해가 되지만 C언어로 구현을 하지 못하는 답답함이 있었습니다. 또한 배운 코드를 printf 수준으로 출력이 되나 확인을 할 수 있는 방법도 몰라서 공부를 하기 힘들었습니다. 비전공자이기에 포인터의 개념도 가볍게 배우고 넘어왔는데 수업에서 진행되는 코드를 따라가기 벅했습니다. 하지만 이번 과제를 진행하면서 조교님께서 이번 과제의 벽만

넘으면 된다는 조언에 다른 과제보다 많은 시간을 투자하여 진행했던 것 같습니다. 이를 통해 C 언어에 기본적인 문법과 linked_list를 눈으로 구현해볼 수 있는 귀한 시간이었습니다. 앞으로 남은 과제들도 성실히 잘 수행해보겠습니다. 감사합니다.