

자료구조응용

09. 연결 리스트 : 스택, 큐

1. [Linked Stacks] 다음과 같이 과목별 스택을 생성하고 실행하는 프로그램을 작성하라. 이를 위해, push, pop, stackEmpty 함수를 구현하여야 한다.

(1) 실행 순서

① 학번 순으로 미리 정렬된 (과목번호, 학번, 성적)의 쌍으로 구성된 입력파일("input.txt")을 입력받으면서 입력되는 순서대로 각 과목 별로 Linked Stack에 저장 하시오. (과목이 3개이므로 3개의 스택이 필요함)

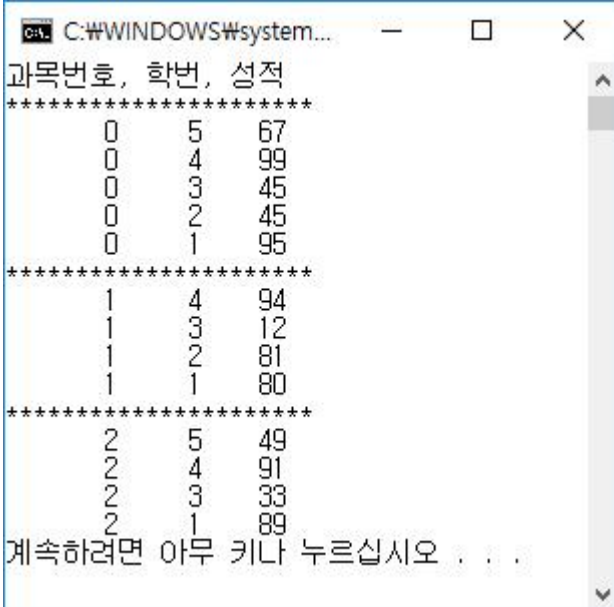
0	1	95
1	1	80
2	1	89
0	2	45
1	2	81
0	3	45
1	3	12
2	3	33
0	4	99
1	4	94
2	4	91
0	5	67
2	5	49

② 각 과목 별로 학번의 역순으로 노드의 데이터(학번, 성적)를 출력하라.

(2) 데이터 선언문 참고

```
#define MAX_STACKS 3
typedef struct {
    int id;           //학번
    int grade;        //성적
} element;
typedef struct stack *stackPointer;
typedef struct stack {
    element data;
    stackPointer link;
}Node;
stackPointer top[MAX_STACKS];
/*****
top[i] = NULL, 0 ≤ i < MAX_STACKS // initial condition
top[i] = NULL, iff the ith stack is empty // boundary condition
*****/
```

실행 예



A screenshot of a Windows command prompt window. The title bar shows the path "C:\WINDOWS\system...". The window contains a table of student scores with columns for subject number, student ID, and score. The table is separated into three sections by lines of asterisks. The first section shows scores for subject 0, the second for subject 1, and the third for subject 2. At the bottom, there is a prompt "계속하려면 아무 키나 누르십시오 . . .".

과목번호	학번	성적

0	5	67
0	4	99
0	3	45
0	2	45
0	1	95

1	4	94
1	3	12
1	2	81
1	1	80

2	5	49
2	4	91
2	3	33
2	1	89

계속하려면 아무 키나 누르십시오 . . .

2. [Linked Queues] 다음과 같이 과목별 큐를 생성하고 실행하는 프로그램을 작성하라.

(1) 실행 순서

① 입력파일("input.txt")로 부터 학번 순으로 미리 정렬된 데이터를 입력받으면서 순서대로 Linked Queue를 구현한다. (과목번호, 학번, 성적)의 쌍으로 데이터들이 입력되며 각 과목별로 큐에 저장된다.

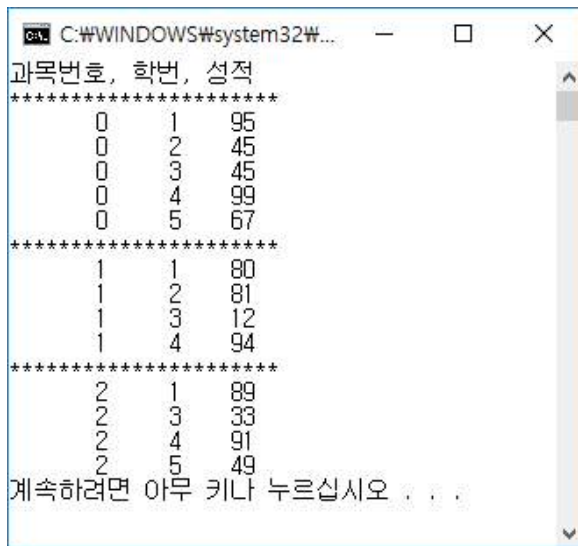
0	1	95
1	1	80
2	1	89
0	2	45
1	2	81
0	3	45
1	3	12
2	3	33
0	4	99
1	4	94
2	4	91
0	5	67
2	5	49

② 각 과목 별로 학번 순으로 노드의 데이터(학번, 성적)를 출력하라.

(2) 데이터 선언문 참고

```
#define MAX_QUEUES 3
typedef struct {
    int id;           //학번
    int grade;        //성적
} element;
typedef struct queue *queuePointer;
typedef struct queue {
    element data;
    queuePointer link;
}Node;
queuePointer front[MAX_QUEUES], rear[MAX_QUEUES];
/*****
front[i] = NULL, 0 ≤ i < MAX_QUEUES // initial condition
front[i] = NULL, iff the ith queue is empty // boundary condition
*****/
```

실행 예



3. 다음과 같이 정렬되지 않는 점수(정수 데이터)를 입력에 대하여 Circularly Linked List를 만들고 실행 예와 같이 수행되는 프로그램을 작성하라.

(1) 실행 순서

① 입력파일("input.txt")로 부터 데이터를 입력받으면서 Circularly Linked List를 만든다.

input.txt
50 80 30 20 19 90
30 55 77 30 87 7

② Circularly Linked List의 처음부터 끝까지 노드의 데이터를 출력한다.

출력 형태 : (노드주소, 데이터 필드, 링크값)

③ 성적이 홀수인 노드를 Circularly Linked List에서 삭제한다.

④ Linked List의 처음부터 끝까지 노드의 데이터를 출력한다.

출력 형태 : (노드주소, 데이터 필드, 링크값)

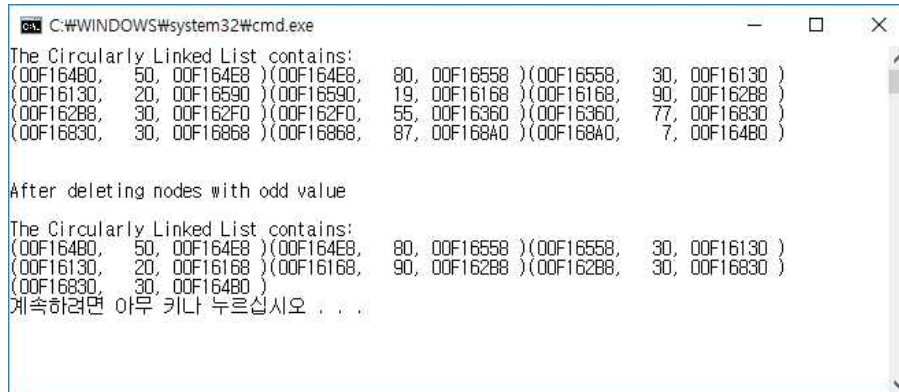
⑤ Linked List를 모두 삭제한다.

(2) 구현 세부사항

① 구조체 정의문은 다음과 같다.

```
typedef struct listNode *listPointer;
typedef struct listNode {
    int data;
    listPointer link;
} listNode;
listPointer first = NULL;
```

실행 예



```
C:\WINDOWS\system32\cmd.exe
The Circularly Linked List contains:
(00F164B0, 50, 00F164E8 )(00F164E8, 80, 00F16558 )(00F16558, 30, 00F16130 )
(00F16130, 20, 00F16590 )(00F16590, 19, 00F16168 )(00F16168, 90, 00F162B8 )
(00F162B8, 30, 00F162F0 )(00F162F0, 55, 00F16360 )(00F16360, 77, 00F16830 )
(00F16830, 30, 00F16868 )(00F16868, 87, 00F168A0 )(00F168A0, 7, 00F164B0 )

After deleting nodes with odd value

The Circularly Linked List contains:
(00F164B0, 50, 00F164E8 )(00F164E8, 80, 00F16558 )(00F16558, 30, 00F16130 )
(00F16130, 20, 00F16168 )(00F16168, 90, 00F162B8 )(00F162B8, 30, 00F16830 )
(00F16830, 30, 00F164B0 )
계속하려면 아무 키나 누르십시오 . . .
```

■ 제출 형식

- 솔루션 이름 : DS_09
- 프로젝트 이름 : 1, 2, 3
- 솔루션 폴더를 압축하여 제출할 것.
- 학습관리시스템에 과제를 올릴 때 제목:
1차 제출: 학번_이름_DS_09(1), 2차 제출: 학번_이름_DS_09(2)
제출은 2회걸쳐 가능(수정 시간 기준으로 처리)