

자료구조응용

09. 연결 리스트 : 스택, 큐

1. [Linked Stacks] 다음과 같이 3개 과목별 스택을 생성하고 실행의 예와 같이 처리하는 프로그램을 작성하라.

(1) 실행 순서

① 각 학반에 대한 3개 교과목별 성적 데이터가 input.txt 파일에 저장되어 있다. input.txt 파일의 각 라인은 (과목번호, 학반, 성적)에 대한 정보로 구성되어 있으며, 각 라인들은 학반에 대하여 정렬되어 있다. 입력파일("input.txt")의 데이터 순서대로 각 과목 별로 해당 교과목 Linked Stack에 저장 하시오.

(과목이 3개이므로 3개의 스택이 필요함)

② 각 과목 별 해당 스택으로부터 데이터를 출력하시오.

input.txt

```
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
```

실행 결과



```
선택 Microsoft Vis...
과목번호, 학반, 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
G:\2021년\2021-1\자료구조응용\실
습코드\DSA-09\Debug\1.exe( 프로세
스 23644개)이(가) 종료되었습니다(
코드: 0개).
이 창을 닫으려면 아무 키나 누르세
요...
```

2. [Linked Queues] 다음과 같이 3개 과목별 큐를 생성하고 실행하는 프로그램을 작성하라.

(1) 실행 순서

① 입력파일("input.txt")의 각 라인은 (과목번호, 학년, 성적)에 대한 정보를 가지고 있다. 입력파일("input.txt")로 부터 학년 순으로 미리 정렬된 데이터를 입력받으면서 순서대로, 해당 학년의 Linked Queue에 저장한다.

② 해당 학년의 Linked Queue에서 데이터(학번, 성적)를 출력하라.

input.txt

```
0 1 95
1 1 80
2 1 89
0 2 45
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
```

실행 결과

```
선택 Microsoft Visual Studio
과목번호, 학반, 성적
*****
0 1 95
0 2 45
0 3 45
0 4 99
0 5 67
*****
1 1 80
1 2 81
1 3 12
1 4 94
*****
2 1 89
2 3 33
2 4 91
2 5 49
*****
G:\2021년\2021-1\자료구조응용\실습
코드\DSA-09\Debug\2.exe(프로세스 9
052개)이(가) 종료되었습니다(코드:

```

3. 다음과 같이 정렬되지 않는 점수(정수 데이터)를 input.txt 파일에서 입력력 받아 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)

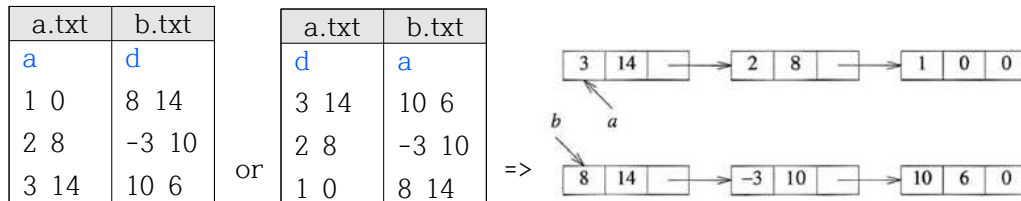
After deleting nodes with odd value

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 . . .
```

4. 다음과 같이 chain을 이용한 다항식 더하기 프로그램을 작성하라.

(1) 실행 순서

① 두 개의 입력파일("a.txt," "b.txt")로부터 데이터를 입력받아서 두 개의 다항식을 chain 형태로 구현한다. 아래 예는 $a = 3x^{14} + 2x^8 + 1$, $b = 8x^{14} - 3x^{10} + 10x^6$ 에 대한 입력 예이다.



※ 첫 줄 입력이 'a'이면 지수(exponent) 차수에 대해 오름차순(ascending order), 'd'이면 내림차순(descending order)으로 입력됨. 오름차순으로 입력되면 각 노드는 chain의 첫 노드로 삽입되어야 하며, 내림차순으로 입력되면 각 노드는 chain의 마지막 노드로 추가됨

- ② a, b 두 다항식의 정보를 실행 예와 같이 출력한다.
- ③ a+b의 결과를 c에 저장하는 다항식 더하기를 실행한다.
- ④ 다항식 c를 실행 예와 같이 출력한다.
- ⑤ 다항식 a, b, c를 모두 삭제한다.

(3) 실행 예

```

선택 Microsoft Visual Studio 디버그 콘솔
a :
(000002726A24BAC0 : +3x^14 : 000002726A24BA20)
(000002726A24BA20 : +2x^8 : 000002726A24B890)
(000002726A24B890 : +1x^0 : 0000000000000000)

b :
(000002726A2578D0 : +8x^14 : 000002726A257A10)
(000002726A257A10 : -3x^10 : 000002726A257560)
(000002726A257560 : +10x^6 : 0000000000000000)

a+b=c :
(000002726A2580F0 : +11x^14 : 000002726A257A60)
(000002726A257A60 : -3x^10 : 000002726A257970)
(000002726A257970 : +2x^8 : 000002726A258050)
(000002726A258050 : +10x^6 : 000002726A257B50)
(000002726A257B50 : +1x^0 : 0000000000000000)

G:\2020년\2020년1학기\자료구조\응용\WDSA-10\64\Debug\1.exe(23444 프로세스)이(가) 0 코드로 인해 종료되었습니다.
이 창을 닫으려면 아무 키나 누르세요.

```

5. 다음과 같이 헤더노드를 가진 단일 환형연결리스트 (singly linked circular list)을 이용한 다항식 더하기 프로그램을 작성하라.

(1) 실행 순서

① 입력파일(“a.txt,” “b.txt”)로부터 데이터를 입력받아서 두 개의 다항식 a, b를 각각 헤더노드를 가진 단일 환형연결리스트 형태로 구현하고, 두 다항식을 더하시오. 더한 결과도 역시 헤더노드를 가진 단일 환형연결리스트 형태로 구현하시오. 그리고 빈 노드를 관리하는 avail 리스트에 리스트 a, b, c 순으로 반환한 후 avail 리스트를 운영체제에 반환하시오.

a.txt	b.txt
d	a
8 14	1 0
-3 10	-7 5
5 8	2 8
10 6	3 10
7 5	3 14
3 0	5 15

※ 첫 줄 입력이 ‘a’이면 지수 차수에 대해 오름차순(ascending order), ‘d’이면 내림차순(descending order)으로 입력됨. 오름차순으로 입력되면 각 노드는 환형리스트의 첫 노드로 삽입되어야 하며, 내림차순으로 입력되면 각 노드는 환형리스트의 마지막 노드로 추가

- ② a, b 두 다항식의 정보를 실행 예와 같이 출력한다.
- ③ a+b의 결과를 c에 저장하는 다항식 더하기를 수행한다.
- ④ 다항식 c를 실행 예와 같이 출력한다.
- ⑤ 다항식 a, b, c를 **avail**에 반납한다.
- ⑥ avail 리스트를 실행 예와 같이 출력한다.
- ⑦ avail을 삭제한다.

(3) 실행 예

```
선택 Microsoft Visual Studio 디버그 콘솔
a :
(000001D08D7BFC30 : -842150451x^-1 : 000001D08D7BFAF0)
(000001D08D7BFAF0 : +8x^14 : 000001D08D7BFB40)
(000001D08D7BFB40 : -3x^10 : 000001D08D7BFE10)
(000001D08D7BFE10 : +5x^8 : 000001D08D7BF8C0)
(000001D08D7BF8C0 : +10x^6 : 000001D08D7BFCD0)
(000001D08D7BFCD0 : +7x^5 : 000001D08D7BF910)
(000001D08D7BF910 : +3x^0 : 000001D08D7BFC30)

b :
(000001D08D7BFBE0 : -842150451x^-1 : 000001D08D7BFEB0)
(000001D08D7BFEB0 : +5x^15 : 000001D08D7BFD70)
(000001D08D7BFD70 : +3x^14 : 000001D08D7BFD20)
(000001D08D7BFD20 : +3x^10 : 000001D08D7BF870)
(000001D08D7BF870 : +2x^8 : 000001D08D7BFC80)
(000001D08D7BFC80 : -7x^5 : 000001D08D7BF9B0)
(000001D08D7BF9B0 : +1x^0 : 000001D08D7BFBE0)

a+b=c :
(000001D08D7BFDC0 : -842150451x^-1 : 000001D08D7BFEB0)
(000001D08D7BFEB0 : +5x^15 : 000001D08D7BFB90)
(000001D08D7BFB90 : +11x^14 : 000001D08D7BF7D0)
(000001D08D7BF7D0 : +7x^8 : 000001D08D7BF780)
(000001D08D7BF780 : +10x^6 : 000001D08D7BF820)
(000001D08D7BF820 : +4x^0 : 000001D08D7BFDC0)

avail :
(000001D08D7BFEB0 : +5x^15 : 000001D08D7BFB90)
(000001D08D7BFB90 : +11x^14 : 000001D08D7BF7D0)
(000001D08D7BF7D0 : +7x^8 : 000001D08D7BF780)
(000001D08D7BF780 : +10x^6 : 000001D08D7BF820)
(000001D08D7BF820 : +4x^0 : 000001D08D7BFDC0)
(000001D08D7BFDC0 : -842150451x^-1 : 000001D08D7BFEB0)
(000001D08D7BFEB0 : +5x^15 : 000001D08D7BFD70)
(000001D08D7BFD70 : +3x^14 : 000001D08D7BFD20)
(000001D08D7BFD20 : +3x^10 : 000001D08D7BF870)
(000001D08D7BF870 : +2x^8 : 000001D08D7BFC80)
(000001D08D7BFC80 : -7x^5 : 000001D08D7BF9B0)
(000001D08D7BF9B0 : +1x^0 : 000001D08D7BFBE0)
(000001D08D7BFBE0 : -842150451x^-1 : 000001D08D7BFAF0)
(000001D08D7BFAF0 : +8x^14 : 000001D08D7BFB40)
(000001D08D7BFB40 : -3x^10 : 000001D08D7BFE10)
(000001D08D7BFE10 : +5x^8 : 000001D08D7BF8C0)
(000001D08D7BF8C0 : +10x^6 : 000001D08D7BFCD0)
(000001D08D7BFCD0 : +7x^5 : 000001D08D7BF910)
(000001D08D7BF910 : +3x^0 : 000001D08D7BFC30)
(000001D08D7BFC30 : -842150451x^-1 : 0000000000000000)

G:\₩2021년₩2021-1₩자료구조응용₩실습코드₩DSA-10₩x64₩Debug₩2.exe(프로세스 19332개)이(가) 종료되었습니다(코드: -1073741819개).
```

■ 제출 형식

- 솔루션 이름 : 학번_이름_DS_09
- 프로젝트 이름 : 1, 2, 3, 4, 5
- 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)