

2020-2

데이터구조및실습

1차 Homework

- 소스코드 명: hw1_1_각자이름.cpp

- 단순연결리스트로 구현한 다항식을 사용하여 사용자가 다항식정보를 입력하고 x값을 입력하면 결과를 출력하는 코드를 작성하세요.
- 조건
 - 사용자가 각 항의 계수와 차수를 반복적으로 입력(마지막 항은 차수를 0으로 입력하기)
 - 미지수 x값을 입력하여 다항식의 결과값 출력
 - 각 항의 차수는 0 이상으로 가정함
 - 파랑색네모: 사용자 입력

Input coef and exponent of polynomial<마지막항: exponent를 0으로 입력>

4 6
2 4
-3 2
5 0

<Polynomial>

$4x^6 + 2x^4 + -3x^2 + 5x^0$

Input the value x: 5

Result of polynomial with x: 63680

Input coef and exponent of polynomial<마지막항: exponent를 0으로 입력>

2 9
-5 7
3 6
-8 4
7 3
6 2
9 0

<Polynomial>

$2x^9 + -5x^7 + 3x^6 + -8x^4 + 7x^3 + 6x^2 + 9x^0$

Input the value x: 3

Result of polynomial with x: 30222

- 소스코드 명: hw1_2_각자이름.cpp

- 이중원형연결리스트에 사용자가 입력한 양의 정수를 리스트의 마지막에 계속 insert한 후, 이를 역순으로 출력하는 코드를 작성하세요.
- 조건
 - 이중원형연결리스트를 사용할 것
 - 각 노드의 데이터는 양의 정수로 가정하고 0을 입력하면 입력종료(0은 제외)
 - 리스트의 끝에 새 노드가 입력되는 함수와 이 리스트를 역순으로 출력하는 함수를 각각 구현할 것

```
Input data in dual linkedlist(0 to quit)
```

```
4 6 9 3 0
```

```
4 <-> 6 <-> 9 <-> 3
```

```
Print the list in reverse order
```

```
3 <-> 9 <-> 6 <-> 4
```

- 소스코드 명: hw1_3_각자이름.cpp(1)

- 단순연결리스트를 사용하여 새로 입력되는 숫자들이 항상 오름차순을 유지할 수 있도록 하는 코드를 작성하세요.
- 조건
 - 단순연결리스트를 사용할 것
 - 각 노드의 데이터는 양의 정수로 가정하고 0을 입력하면 입력종료(0은 제외)
 - 함수 프로토타입: `ListNode* insert_sort(ListNode* head, int num)`
- 힌트
 - head부터 링크를 증가시키며 값을 비교하여 insert할 위치를 찾되 insert할 위치의 포인터를 저장해두기
 - 새로 정의하는 함수에서 기존의 `insert_first`와 `insert` 함수를 적절히 호출해서 사용

• 소스코드 명: hw1_3_각자이름.cpp(2)

문제3

```
Input data to insert to linkedlist(0 to quit)
6
6 -> NULL
3
3 -> 6 -> NULL
1
1 -> 3 -> 6 -> NULL
4
1 -> 3 -> 4 -> 6 -> NULL
9
1 -> 3 -> 4 -> 6 -> 9 -> NULL
2
1 -> 2 -> 3 -> 4 -> 6 -> 9 -> NULL
0
1 -> 2 -> 3 -> 4 -> 6 -> 9 -> NULL
Program finished.
```

```
Input data to insert to linkedlist(0 to quit)
2
2 -> NULL
7
2 -> 7 -> NULL
9
2 -> 7 -> 9 -> NULL
1
1 -> 2 -> 7 -> 9 -> NULL
8
1 -> 2 -> 7 -> 8 -> 9 -> NULL
3
1 -> 2 -> 3 -> 7 -> 8 -> 9 -> NULL
1
1 -> 1 -> 2 -> 3 -> 7 -> 8 -> 9 -> NULL
5
1 -> 1 -> 2 -> 3 -> 5 -> 7 -> 8 -> 9 -> NULL
0
1 -> 1 -> 2 -> 3 -> 5 -> 7 -> 8 -> 9 -> NULL
Program finished.
```

• 소스코드 명: hw1_4_각자이름.cpp

- 입력된 문자열이 대칭을 이루는지 체크하는 코드를 1. 배열이용, 2. 스택이용 두가지 사용자 정의함수로 작성하세요.
- 조건
 - main함수에서 사용자가 "0"을 입력할 때까지 계속 반복하기
 - 배열의 index를 이용한 함수 처리결과와 스택의 push/pop을 이용한 함수 처리결과를 각각 출력
 - 두 함수를 각각 구현해서 각각 호출하기

```

알파벳으로 구성된 문자열 입력
abcdcba
대칭<배열>
대칭<스택>

알파벳으로 구성된 문자열 입력
abcdef
비대칭<배열>
비대칭<스택>

알파벳으로 구성된 문자열 입력
abbba
대칭<배열>
대칭<스택>

알파벳으로 구성된 문자열 입력
abba
대칭<배열>
대칭<스택>

알파벳으로 구성된 문자열 입력
abcdcbcdcba
대칭<배열>
대칭<스택>

알파벳으로 구성된 문자열 입력
0
  
```

- 소스코드 명: hw1_5_각자이름.cpp(1)

- 사용자가 입력하는 양수 N 이 주어질 때 다음과 같은 규칙에 의해 마지막으로 남는 카드의 번호를 찾는 코드를 작성하세요.
- 조건
 - 각 카드는 차례로 1부터 N 까지의 번호가 붙어 있으며, 1번 카드가 제일 위, N 번 카드가 제일 아래인 상태에서 아래 동작을 카드가 1장 남을 때까지 반복
 - 반복할 동작(1과 2 동작을 계속 반복)
 1. 제일 위에 있는 카드를 버림
 2. 다음으로 제일 위에 있는 카드를 제일 아래에 있는 카드 밑으로 옮김
- (예) $N=3$ 인 경우
 - 단계1: 123
 - 단계2: 23 (맨위 1을 버림)
 - 단계3: 32 (맨위 2를 맨 아래로 옮김)
 - 단계4: 2 (맨위 3을 버림 → 최종 결과 2)
- 조건
 - 연결리스트로 구현된 큐를 활용하기
 - 연결리스트 시작주소를 받아 노드의 개수를 반환하는 함수 구현 및 사용

• 소스코드 명: hw1_5_각자이름.cpp(2)

• 실행예

```
Input number of cards: 5
1 : 2 : 3 : 4 : 5 :
2 : 3 : 4 : 5 :
3 : 4 : 5 : 2 :
4 : 5 : 2 :
5 : 2 : 4 :
2 : 4 :
4 : 2 :
2 :
Final element is 2
```

```
Input number of cards: 10
1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 : 9 : 10 :
2 : 3 : 4 : 5 : 6 : 7 : 8 : 9 : 10 :
3 : 4 : 5 : 6 : 7 : 8 : 9 : 10 : 2 :
4 : 5 : 6 : 7 : 8 : 9 : 10 : 2 :
5 : 6 : 7 : 8 : 9 : 10 : 2 : 4 :
6 : 7 : 8 : 9 : 10 : 2 : 4 :
7 : 8 : 9 : 10 : 2 : 4 : 6 :
8 : 9 : 10 : 2 : 4 : 6 :
9 : 10 : 2 : 4 : 6 : 8 :
10 : 2 : 4 : 6 : 8 :
2 : 4 : 6 : 8 : 10 :
4 : 6 : 8 : 10 :
6 : 8 : 10 : 4 :
8 : 10 : 4 :
10 : 4 : 8 :
4 : 8 :
8 : 4 :
4 :
Final element is 4
```

```
Input number of cards: 0
queue empty
```


Homework 결과 제출

1. 각 소스코드에 **주석** 30%이상 포함
2. 학과,학번,성명을 출력하는 **함수를 호출**하여 결과화면에 출력
3. 각 소스 코드와 결과화면 캡처한 그림 파일을 하나의 파일로 압축하여
제출

→ 압축파일명: HW1_각자이름.zip