객체지향프로그래밍

3-1

1. [Operation Overloading] Class Person은 사람이름, 나이, 거주지 정보를 가지고 있는 Class이다. 아래 main함수가 예시와 같이 동작하도록 Class를 구현하고 동작시키시오.

```
int main()
{
    Person P;
    cin >> P;
    cout << P;
    return 0;
}
```

main 함수

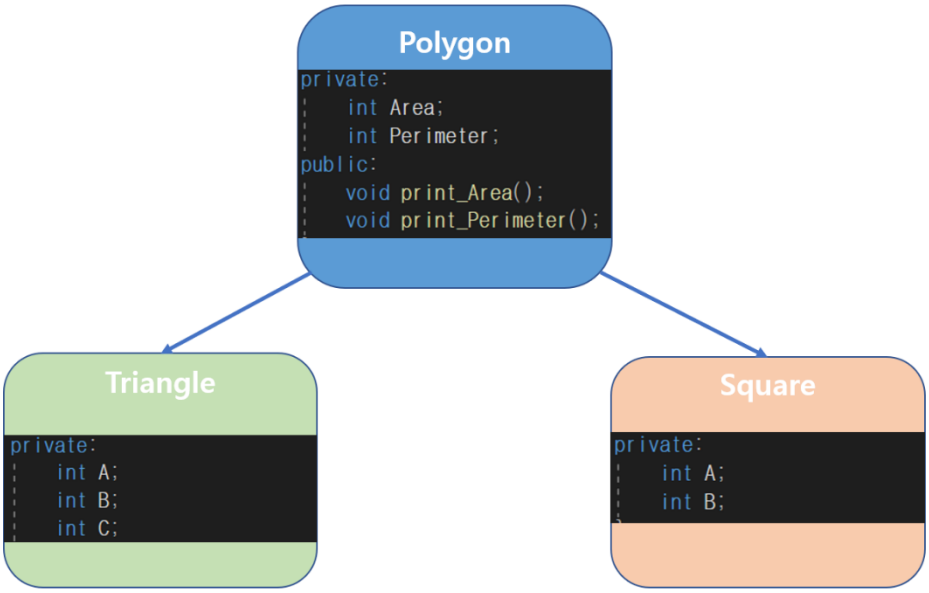
```
박우혁 28 부천
```

입력 예시

```
박우혁 28 부천
이름 :박우혁
나이 :28
거주지 :부천
```

출력 예시

2. [Inheritance] Class Polygon과 Triangle, Square는 아래그림(Feature 1)과 같은 상속관계를 가진다. 각각의 Class는 Feature 1에 있는 멤버 변수, 멤버 함수만 가지고 있으며 생성자와 소멸자를 추가할 수 있다. 프로그램의 main함수가 정상적으로 동작해서 예시와 같이 Area(넓이)와 Perimeter(둘레)가 출력되도록 나오도록 Class들을 구현하시오. (Triangle은 헤론의 공식을 사용하면 Area를 구할 수 있음)



<Feature 1>

```
class Square A(5,4);
class Triangle B(13, 14, 15);

cout << "--------Square--------"<<endl;
A.print_Area();
A.print_Perimeter();
cout << "----------------------" << endl;
cout << "-------Triangle-------" << endl;
B.print_Area();
B.print_Perimeter();
cout << "----------------------" << endl;
```

```
--------Square--------
Area is 20
Perimeter is 18
----------------------
-------Triangle-------
Area is 84
Perimeter is 42
----------------------
```

<main함수 및 결과화면>

3. [Operation Overloading] mystring class는 private권한의 char *형 멤버변수 string과 public권한의 print함수 그리고 생성자와 소멸자만 멤버로 가지고 있으며 +=, -=, ==, && operator에 대해서 table 1과 같은 동작을 하는 class이다. 프로그램은 table 2와 같은 command를 가지며 exit명령을 받을 때 까지 반복해서 command를 입력받는다.

operator	사용 방법	동작 설명
+=	mystring += char *	string에 char*를 이어붙임
-=	mystring -= char	string에서 char를 모두 제거
--	mystring--	string의 마지막 문자를 제거 마지막 문자가 없을 경우 동작하지않음
&&	mysting && char	string에서 char외에 모두 0으로 변환

<table 1>

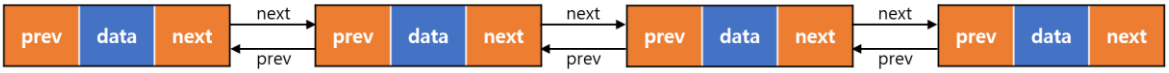
Command number	Command name	사용법	동작 설명
1	add	1 string2	기존 문자열에 string2을 추가(string2의 길이는 최대 100이며 기존 문자열(string)의 길이는 제한없이 동작해야함)
2	delete_char	2 character	기존 문자열에서 character를 모두 제거 (character가 w일 경우 예시 : kwangwoon -> kangoon)
3	delete_lastchar	3	기존 문자열의 마지막문자를 제거, 문자열에 문자가 없을 경우 동작하지않음
4	and_operator	4 character	문자열에서 문자 외에 모두 0으로 변환 (character가 w일 경우 예시 : kwangwoon -> 0w000w000n)
5	print	5	현재 문자열 출력
6	exit	6	프로그램 종료

<table 2>

```
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 1 Kwang
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 1 woon
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 2 w
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 5
Kangoon
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 3
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 5
Kangoo
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 1 operator
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 5
Kangoooperator
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 4 a
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 5
0a00000000a000
Please Enter command(1 : add, 2 : delete_char, 3 : delete_lastchar, 4 : and_operator, 5 : print, 6 : exit) : 6
```

<프로그램 예시>

4. [Linked List]번호와 이름을 저장, 출력, 정렬, 삭제하는 프로그램을 구현하시오. 번호와 이름은 Node class에 저장되어 있어야 하며 doubly linked list에 저장되어 관리되어야 한다. Doubly linked list란 그림과 같이 Node가 양방향으로 연결되어 있는 구조를 말한다. 프로그램은 table 1과 같은 command를 가지며 exit 명령을 받을 때까지 반복해서 command를 입력받는다.



<doubly linked list 예시>

Command number	Command name	사용법	동작 설명
1	insert	1 ID name	Linked list에 새로운 정보를 추가한다. 정보는 ID를 기준으로 처음부터 시작해 자신보다 ID가 큰 Node를 만나면 그 자리에 추가된다. (예 : 1 7 5 8에 6을 추가할 경우 1 6 7 5 8) 단 ID가 중복될 경우 저장하지 않는다
2	print	2	저장되어있는 정보를 정방향으로 출력한다
3	print_reverse	3	저장되어있는 정보를 역순으로 출력한다
4	sort_by_name	4	저장되어 있는 정보를 이름기준 오름차순으로 정렬한다.
5	sort_by_ID	5	저장되어 있는 정보를 ID기준 오름차순으로 정렬한다
6	delete	6 ID	정보 중 ID가 같은 정보를 삭제한다
7	exit	7	프로그램 종료

<table 1>

```
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 1 75 Park
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 1 54 Kim
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 1 42 Lee
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 1 88 Hong
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 1 75 An
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 2
42 Lee
54 Kim
75 Park
88 Hong
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 3
88 Hong
75 Park
54 Kim
42 Lee
Please Enter command(1 : insert, 2 : print, 3 : prin_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 4
88 Hong
54 Kim
42 Lee
75 Park
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 1 32 Han
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 2
32 Han
88 Hong
54 Kim
42 Lee
75 Park
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 5
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 2
32 Han
42 Lee
54 Kim
75 Park
88 Hong
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 6 75
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 2
32 Han
42 Lee
54 Kim
88 Hong
Please Enter command(1 : insert, 2 : print, 3 : print_reverse, 4 : sort_by_name, 5 : sort_by_ID, 6 : delete, 7 : exit) : 7
```

<프로그램 예시>