```
#ifndef __CALCULATOR_HPP__
#define __CALCULATOR_HPP__

#include<iostream>
/*
이 파일은 calculator.hpp 파일입니다.
이 파일에는 정수 연산(Add, Sub, Div, Mul)클래스와
각각의 클래스 구성 요소들(메소드 및 멤버 변수)이 선언되어 있습니다.
이 클래스들은 각각 impl.cpp에서 최종 정의되며, main.cpp에서 각각의 클래스들의
메소드가 불려집니다(Add a; Sub b; ..이후 a.setValue(int x, int y); 그리고
a.calculate()).
위에서 setValue(int x, int y) 메소드는 앞서 말한 초기화되지 않은 변수(a, b)에
x와 y의 값을 각각 a와 b의 값에 대입하는 역할을 합니다.
그리고 int calculate()란 메소드는 정수 연산을 하여 setValue(int x, int y)에서
정의된 a와 b의 값을 연산하는 역할을 합니다.
여기에서 생성자의 역할이 중요합니다.하지만, 이 코드에서는 명시적인 생성자가
보이지 않습니다. 그렇다면 이 코드를 메인 함수에서 실행할 때에는 어떻게
a.setValue(int x, int y)라든지 a.calculate() 함수가 어떻게 실행될 수
있는 것인가요? 그 해답은 바로 이 코드를 해석하여 기계어 코드로 변환하는
컴파일러에 있습니다.
 C++ 컴파일러는 클래스에 생성자가 하나도 없으면 디폴트 생성자를 집어넣습니다.
따라서 우리가 컴파일 할 때에 아무런 오류가 나지 않는 것이지요.
예를 들어 Add a;로 선언할 때에 a는 Add 클래스의 멤버들을 부를 수 있는 매개변수입니다.
따라서 멤버들을 부를 수 있는 매개변수가 통로가 되어 Add 클래스의 요소들을 부를 수
있는 것입니다.
```

```
*/
class Add{ //클래스 Add 선언
public:
int a, b; //정수 타입 a, b 변수 선언(초기화되지 않은)
void setValue(int x, int y);
int calculate();

};
class Sub{ //클래스 Sub 선언
public:
int a, b;
void setValue(int x, int y);
int calculate();

};

class Mul{
public:
int a, b;
void setValue(int x, int y);
int calculate();

};
```

```
class Div{

public:

int a, b;

void setValue(int x, int y);

int calculate();


};



#endif




#include"calculator.hpp"

/*

이 파일은 calculator.cpp입니다.

Calculator.hpp를 include하여 정의한 것입니다.

*/

void Add::setValue(int x, int y){

a=x;

b=y;



}

int Add::calculate(){
```

```
    return a+b;
}
void Sub::setValue(int x, int y){
a=x;
b=y;



}
int Sub::calculate(){

    return a-b;
}
void Mul::setValue(int x, int y){
a=x;
b=y;



}
int Mul::calculate(){

    return a+b;
}
void Div::setValue(int x, int y){
a=x;
b=y;
```

```
}
int Div::calculate(){

	return a+b;
}
```

```
#include "impl.cpp"
/*이 파일은 main.cpp입니다.
이 파일에서는 Add 클래스와 Sub 클래스 , Mul 클래스, Div 클래스를 호출하고 있습니다.
그리고 while(true) 구문을 통하여 main 함수 내의 기능들을 반복하고 있습니다.
여기에서 주목할 것은 char c; 선언입니다.
문자 변수 c는 아직 초기화 되지 않았습니다(while 구문 이전에).
만약 변수 'c'가 정수 산술연산을 나타내는 기호이면 클래스 Add 혹은 Sub 혹은 Mul
혹은 Div의 setValue(int x, int y)를 부르고 해당 클래스의 멤버 변수(a,b)에 우리가
입력한 초기값(x, y)을 할당합니다.
이후 해당 클래스의 calculate()를 부른 뒤 정수 연산을 한 후에 결과값을 리턴합니다.
이 메인 함수는 Ctrl+C를 하기 전까지는 멈추지 않습니다.

다음은 실행 예시 입니다.
enter the two integers and operators : 9 8 -
1
enter the two integers and operators : 9 7 (
unknown operator
```

...

이런 식으로 반복이 됩니다.

```
*/
int main(int argc, char *argv[]){
Add a;
Sub s;
Mul m;
Div d;
char c;
int x;
int y;
while(true){
std::cout<<"enter the two integers and operators : ";

std::cin>>x>>y>>c;


if(c=='+'){
	a.setValue(x,y);
	std::cout<<a.calculate()<<std::endl;
}
if(c=='*'){
```

```
m.setValue(x,y);
std::cout<<m.calculate()<<std::endl;
}
if(c=='/')
{
d.setValue(x,y);
std::cout<<d.calculate()<<std::endl;

}
if(c=='-')
{
        s.setValue(x,y);
      std::cout<<s.calculate()<<std::endl;
}
else{
        std::cout<<"unknown operator"<<std::endl;
        continue;

}
}
}
```