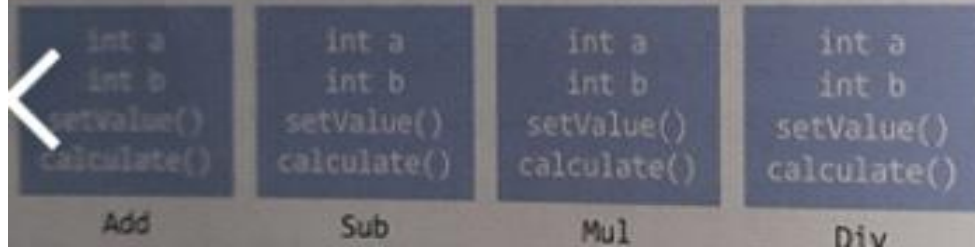


다수의 클래스를 선언하고 활용하는 간단한 문제이다. 더하기(+), 빼기(-), 곱하기(*), 나누기(/)를 수행하는 4개의 클래스 Add, Sub, Mul, Div를 만들고자 한다. 이들은 모두 공통으로 다음 멤버를 가진다.

문제를 보면 Add Sub Mul Div라는 클래스를 4개 만들라고 한다.

은 모두 공통으로 다음 멤버를 가진다.

- int 타입 변수 a, b : 피연산자
- void setValue(int x, int y) 함수 : 매개 변수 x, y를 멤버 a, b에 복사
- int calculate() 함수 : 연산을 실행하고 결과 리턴



각 클래스에는 int a, int b, setValue(), calculate()라는 멤버를 가졌다고 명시를 해주었다

```

#include <iostream>
using namespace std;

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

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

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

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

```

따라서 보기와 같이 총 4개의 클래스를 만들어 줬다. 정수타입 변수 a, b는 접근을 보호하기 위해 private를 선언하였고(접근 지정의 디폴트가 private이긴 하다) setValue와 calculate는 접근을 위해 public을 선언하였다.

```

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;
}

```

그다음 구현부를 보기와 같이 만들어주었다. setValue()함수로 x, y값을 받아와 a, b에 저장하고. calculate로 각 클래스마다의 계산 방법으로 결과값을 리턴 시키게 하였다.

```
두 정수와 연산자를 입력하세요>>3 4 *  
12  
두 정수와 연산자를 입력하세요>>5 2 /  
2  
두 정수와 연산자를 입력하세요>>8 2 -  
6
```

- (1) 클래스의 선언부와 구현부를 분리하고, 모든 코드를 `Calculator.cpp` 파일에 작성하라. **도움말 6**
- (2) 클래스의 선언부와 구현부를 헤더 파일과 `cpp` 파일로 나누어 프로그램을 작성하라. **도움말 7**

결과창과 같이 출력하기 위해 메인 함수를 짜줬다.

```

int main() {
    int x, y;
    char c;
    Add a;
    Sub s;
    Mul m;
    Div d;

    cout << "두 정수의 연산자를 입력하세요>>";
    cin >> x >> y >> c;
    if (c == '+') {
        a.setValue(x, y);
        cout << a.calculate() << endl;
    }
    else if (c == '-') {
        s.setValue(x, y);
        cout << s.calculate() << endl;
    }
    else if (c == '*') {
        m.setValue(x, y);
        cout << m.calculate() << endl;
    }
    else if (c == '/') {
        d.setValue(x, y);
        cout << d.calculate() << endl;
    }
}

#include <iostream>
using namespace std;

#include "calculator.h"

int main() {
    int x, y;
    char c;

    cout << "두 정수의 연산자를 입력하세요>>";
    cin >> x >> y >> c;
    if (c == '+') {
        Add a;
        a.setValue(x, y);
        cout << a.calculate() << endl;
    }
    else if (c == '-') {
        Sub s;
        s.setValue(x, y);
        cout << s.calculate() << endl;
    }
    else if (c == '*') {
        Mul m;
        m.setValue(x, y);
        cout << m.calculate() << endl;
    }
    else if (c == '/') {
        Div d;
        d.setValue(x, y);
        cout << d.calculate() << endl;
    }
}

```



메인 함수에는 숫자를 입력 받을 지역 변수 x, y를 선언하고,
연산자를 구별할 문자변수 c를 선언 하였다.

그리고 각 클래스의 객체 a, s, m, d를 선언하였다.

cout으로 문자와 연산자를 입력하라 한 다음 cin으로 받은 숫자와 연산자를 각각 x y c에 저장하고 받은 연산자를 비교하여 각 연산자에 맞는 클래스 객체를 불러와 계산을 진행하였다.

나중에 객체를 먼저 생성할 필요가 없다고 생각하여 if문에 추가해 필요한 연산객체를 생성하도록 바꾸었다

```
두 정수와 연산자를 입력하세요>>8 5 -  
3
```

코드를 실행하면 위와같이 뜨는 것을 확인할 수 있다.

이제 문제 2번을 위해 헤더파일과 구현부 메인 으로 코드를 나누어 주겠다

```

#ifndef CALCULATOR_H
#define CALCULATOR_H

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

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

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

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

```

헤더 파일에 클래스 선언부를 넣어주고 #ifndef CALCULATOR_H, #define CALCULATOR_H, #endif 를 넣어 헤더파일을 중복 include할때 발생하는 문제를 해결 하였다.


```

#include "calculator.h"

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;
}

```

다른 cpp에 #include "calculator.h"로
헤더파일을 불러오고 구현부를 옮겨
주었다.

```

#include <iostream>
using namespace std;

#include "calculator.h"

int main() {
    int x, y;
    char c;

    cout << "두 점수와 연산자를 입력하세요>>";
    cin >> x >> y >> c;
    if (c == '+') {
        Add a;
        a.setValue(x, y);
        cout << a.calculate() << endl;
    }
    else if (c == '-') {
        Sub s;
        s.setValue(x, y);
        cout << s.calculate() << endl;
    }
    else if (c == '*') {
        Mul m;
        m.setValue(x, y);
        cout << m.calculate() << endl;
    }
    else if (c == '/') {
        Div d;
        d.setValue(x, y);
        cout << d.calculate() << endl;
    }
}

```

마지막으로 다른 cpp에 메인 함수를 옮겨주었고
마찬가지로 #include "calculator.h"로 헤더파일을
불러올 수 있게 하였다.

두 정수와 연산자를 입력하세요 >> 77 5 /
15

결과또한 잘 나온다.