

문제 1 – 참조변수, 복사생성자

- MyIntStack 클래스는 다음과 같이 선언되어 있다. 메인 함수가 오류 없이 동작하도록 MyIntStack을 구현해 보시오.

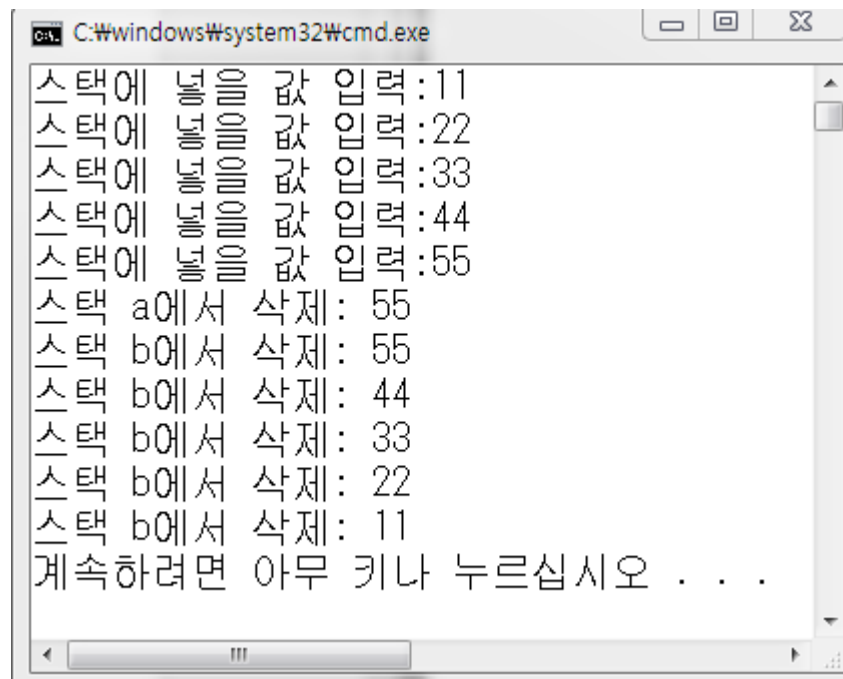
```
class MyIntStack {  
    int *p; // 스택 메모리로 사용할 포인터  
    int size; // 스택의 최대 크기  
    int tos; // 스택의 탑을 가리키는 인덱스  
public:  
    MyIntStack(int size=10);  
    MyIntStack(const MyIntStack& s);  
    ~MyIntStack();  
    bool push(int n); // 정수 n을 스택에 푸시한다.  
        // 스택이 꽉 차 있으면 false를, 아니면 true 리턴  
    bool pop(int &n); // 스택의 탑에 있는 값을 n에 팝한다.  
        // 만일 스택이 비어 있으면 false를, 아니면 true 리턴  
};
```

문제 1

- MyIntStack 클래스는 다음과 같이 선언되어 있다. 메인 함수가 오류 없이 동작하도록 MyIntStack을 구현해 보시오.

```
int main() {
    MyIntStack a(5);
    int value;
    for(int i=0; i < 5; i++ )
    {
        cout << "스택에 넣을 값 입력:";
        cin >>value;
        if(!a.push(value)) cout << "스택 full \n";
    }
    MyIntStack b = a;

    if( a.pop( value ))
        cout << "스택 a에서 삭제: " << value << endl;
    for( int i=0; i < 6;i++ )
        if( b.pop( value ))
            cout << "스택 b에서 삭제: " << value << endl;
    return 0;
}
```



```
C:\windows\system32\cmd.exe
스택에 넣을 값 입력:11
스택에 넣을 값 입력:22
스택에 넣을 값 입력:33
스택에 넣을 값 입력:44
스택에 넣을 값 입력:55
스택 a에서 삭제: 55
스택 b에서 삭제: 55
스택 b에서 삭제: 44
스택 b에서 삭제: 33
스택 b에서 삭제: 22
스택 b에서 삭제: 11
계속하려면 아무 키나 누르십시오 . . .
```

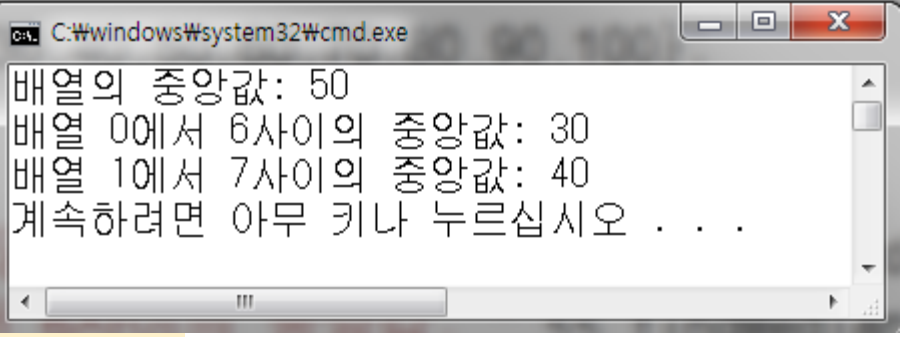
문제 2 - 함수 중복

- 함수 findMedian 는 배열에서 중앙값을 찾는 함수이다. 다음과 같은 실행 결과가 나오도록 findMedian 함수를 구현하시오.

```
int main()
{
    int a[]={0,10,20,30,40,50,60,70,80,90,100};
    int n=11;

    cout<< "배열의 중앙값: " << findMedian(a,n) << endl;
    cout<< "배열 0에서 6사이의 중앙값: " << findMedian(a,n,6) << endl;
    cout<< "배열 1에서 7사이의 중앙값: " << findMedian(a, n, 1, 7) << endl;

    return 0;
}
```



C:\windows\system32\cmd.exe

```
배열의 중앙값: 50
배열 0에서 6사이의 중앙값: 30
배열 1에서 7사이의 중앙값: 40
계속하려면 아무 키나 누르십시오 . . .
```

문제 3 –연산자 중복

- GameItem 클래스의 속성은 string name, int power, double duration으로 구성되어 있다. 메인 함수가 오류 없이 동작하도록 GameItem 을 구현해 보시오.

```
int main()
{
    GameItem item1("sword", 5,0.5), item2("magic"), item3("shield", 2, 10.1);

    cout << "#### 생성된 게임 아이템 종류\n";
    cout << item1 << item2 << item3 ;

    item2 = 10+item2;
    cout<< "\n#### 변경된 게임아이템\n" << item2;

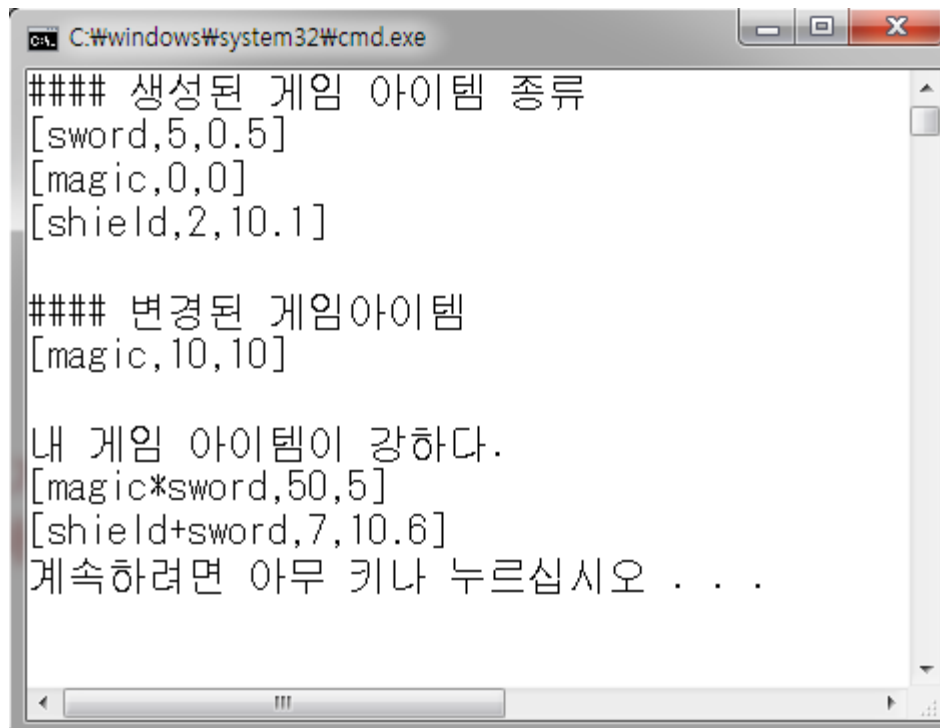
    GameItem myitem = item2*item1;
    GameItem youritem = item3+item1;

    if( myitem > youritem ) cout << "\n내 게임 아이템이 강하다.\n" << myitem << youritem;
    else cout << "\n너의 게임 아이템이 강하다.\n" << youritem << myitem ;

    return 0;
}
```

문제 3 –연산자 중복

- GameItem 클래스의 속성은 string name, int power, double duration으로 구성되어 있다. 메인 함수가 오류 없이 동작하도록 GameItem 을 구현해 보시오.



```
C:\windows\system32\cmd.exe
#### 생성된 게임 아이템 종류
[sword,5,0.5]
[magic,0,0]
[shield,2,10.1]

#### 변경된 게임아이템
[magic,10,10]

내 게임 아이템이 강하다.
[magic*sword,50,5]
[shield+sword,7,10.6]
계속하려면 아무 키나 누르십시오 . . .
```