

문제 1. Prob1.java 파일을 열어서 다음의 조건을 구현하여 완성하십시오.

조건1: main 메소드에서 MySum 객체를 출력하면 객체 생성시에 전달한 두 정수의 덧셈 결과를 출력하도록 toString 메소드를 오버라이딩합니다.

예)

```
MySum ms1 = new MySum(i, j);  
System.out.println(ms1);
```

호출시에 출력값은 i와 j의 산술 연산 덧셈 결과입니다.

(참고 : String.valueOf 메소드는 int를 String으로 변경합니다)

조건2: main 메소드에서 MySum 클래스의 equals 메소드 호출시 전달하는 매개변수가 MySum 객체를 참조하고 현재 객체의 toString 메소드와 전달 객체의 toString 메소드 내용이 동일하면 true를 리턴하고 나머지 경우에는 false 를 리턴하도록 equals 메소드를 오버라이딩합니다.

예)

```
MySum ms1 = new MySum(i, j);  
MySum ms2 = new MySum(j, i);  
String s1 = new String("30");
```

ms1.equals(ms2) 호출 결과는 true입니다.

ms1.equals(s1) 호출 결과는 false입니다.

실행 결과 예) java Exam02

30

30

ms1과 ms2의 합계는 동일합니다.

ms1과 s1의 값은 동일하지 않습니다.

문제 2. 주어진 Prob2클래스가 정상적으로 실행되기 위해 MyStack 클래스를 구현하시오.

MyStack 클래스에 대한 요구조건

1. MyStack클래스는 기본생성자로 생성하면 int 타입의 정수를 10개만 저장하는 Stack 클래스이다.
2. 객체 생성 시 생성자 매개변수로 배열의 크기를 지정할 수 있으나 음수가 매개변수로 들어올 경우는 기본적으로 10개의 정수를 저장하도록 한다.
3. push() 메서드로 Stack에 정수를 저장한다.
4. isEmpty() 메서드로 Stack이 비어있는지 확인할 수 있다.
5. **isFull()** 메서드로 Stack이 가득찼는지 확인할 수 있다.
6. top() 메서드로 Stack에서 최상위에 저장된 숫자가 뭔지 알 수 있다. top() 메서드를 호출했다고 해서 숫자가 삭제되는 것은 아니다. 꺼낼 숫자가 없는 경우 -1을 리턴한다.
7. pop() 메서드로 Stack에서 최상위에 저장된 숫자를 꺼낼 수 있다. 스택에서 숫자를 꺼내면 그 숫자는 Stack에서 삭제된다. 꺼낼 숫자가 없는 경우 -1을 리턴한다.
8. java.util.Stack class는 사용하지 않는다.

<< Prob2 실행 결과 >>

스택이 비어있습니다.

스택이 가득 찼습니다.

최상위 숫자 : 10

최상위에서 꺼낸 숫자 : 10

최상위에서 꺼낸 숫자 : 9

== 스택 리스트 ==

8
7
6
5
4
3
2
1

문제3. 주어진 Prob3.java 파일을 이용하여 결과대로 나올수 있게 Entry 클래스의 하위 클래스인 Directory 클래스와 File 클래스를 완성하시요.

<<수행결과>>

```
tetD  
testF(1000)
```

<<참고 및 주의사항>>

- main 메서드는 수정없이 그대로 사용합니다.
- 주어진 메소드의 signature는 변경하지 않습니다.
- Entry 클래스는 수정없이 그대로 사용합니다.
- File 클래스와 Directory 클래스내에 어떠한 변수도 선언되지 않아야 합니다.