문제 1. 다음과 같은 2가지 클래스가 제공된다. Prob1 클래스를 실행했을 때 실행결과와 같이 나올 수 있도록 MyBase클래스를 작성하시오.

Prob1.java

```
public class Prob1 {
	public static void main(String args[]){
		Base base = new MyBase();
		base.service("낮");
		base.service("밤");
		base.service("오후");
	}
}
```

Base.java

```
public class Base {
	public void service(String state){
		if(state.equals("낮"))
			day();
		else
			night();
	}

	public void day(){
		System.out.println("낮");
	}

	public void night(){
		System.out.println("night");
	}
}
```

<<실행결과>>

```
낮에는 열심히 일하자!
night
오후도 낮과 마찬가지로 일해야 합니다.
```

주의 사항 :

1. 주어진 main메소드를 이용하여 테스트 합니다.
2. 주어진 Class Base는 절대로 수정하면 안됩니다.
3. 출력결과가 하나만 다른 경우 C로 처리합니다.

문제2. JAVA Program 문제입니다.

문제 개요 : 아래 클래스 설명을 만족하고, 배포되는 Prob2.java 의 실행결과가 나오도록 BankCustomer 클래스를 작성하세요.

BankCustomer	
String name;	// 고객 이름
Int currentMoney;	// 현재 잔액
BankCustomer (String name, int currentMoney)	// 고객 이름과 잔액을 인자로 받는 생성자
Int getCurrentMoney()	// 현재 잔액을 리턴한다.
void withdraw(int money)	// 현재 잔액에서 일정 금액을 인출한다. // 현재 잔액이 인출하려는 금액보다 적은 경우, "잔액이 부족하여 인출할 수 없습니다."라는메시지를 출력한다.

<<실행결과>>

```
현재 잔액 : 100
현재 잔액 : 30
잔액이 부족하여 인출할 수 없습니다.
```

<<주의사항>>

1. main() 메서드는 수정 없이 그대로 사용합니다.
2. 문제에서 제시된 실행결과와 동일한 결과가 출력되어야 합니다.
3. Product 클래스는 별도의 java 파일로 작성하지 않고 Prob4.java 와 동일한 파일에 구현합니다.
4. 잔액 부족에 대한 예외 메시지가 출력되지 않고 마이너스 금액으로 차감된 경우, 부분점수(B) 를 부여하고 나머지 경우는 모두 오답(D)으로 처리합니다.

문제 1. 아래 클래스 설명을 만족하고, 배포되는 CellPhoneMain의 실행결과가 나오도록 CellPhone 클래스를 작성하세요.

CellPhone	
String model	// 핸드폰 모델 번호
double battery;	// 남은 배터리 양
CellPhone(String model)	// 모델 번호를 인자로 받는 생성자
void call(int time)	// 통화 시간(분)을 출력하고, 통화 시간에 따

	라 배터리 양을 감소시킨다.
	// **감소되는 배터리 양 = 통화시간(분) * 0.5**
	// 배터리 양은 0보다 작아지지 않는다.
	// 통화 시간(분)이 0보다 작은 경우에는 "통화시간입력오류"라는 메시지를 출력한다.

void charge(int time)	// 충전한 시간(분)을 출력하고, 충전한 시간에 따라 배터리 양을 증가시킨다.
	// **충전되는 배터리 양 = 충전시간(분) * 3**
	// 배터리 양은 100까지만 증가한다.
	// 충전 시간(분)이 0보다 작은 경우에는 "충전시간입력오류"라는 메시지를 출력한다.

void printBattery()	// 남은 배터리 양을 출력한다.
public boolean equals(Object otherObject)	// Object 타입의 객체를 입력받고, 입력받은 객체가CellPhone 타입의 클래스이면서 모델 번호가 같은 경우에 true를 리턴한다.

<<CellPhoneMain 실행결과>>

충전 시간 : 20분
남은 배터리 양 : 60.0
통화 시간 : 300분
남은 배터리 양 : 0.0
충전 시간 : 50분
남은 배터리 양 : 100.0
통화 시간 : 40분
남은 배터리 양 : 80.0
통화시간입력오류
동일 모델입니다.

문제4. Keyboard로부터 구매하려는 상품코드와 상품팩의 가격, 구매하려는 상품팩의 개수 등을 입력받아서 지불해야 할 금액을 계산하는 Purchase 프로그램을 작성하십시오.

1) 상품코드(productCode), 상품 가격(productCost), 구매할 상품팩의 개수(numOfPack)와 하나의 상품팩에 포함된 상품의 개수(numOfProduct)를 멤버 변수로 갖는 Product 클래스를 구현합니다. 각 멤버 변수에 대하여 getter, setter 메소드를 구현합니다.

2) 다음은 Purchase 클래스의 구현에 필요한 요구사항들입니다.

A. 아래와 같이 상품코드와 상품팩의 가격, 구매하려는 상품팩의 개수 등을 Keyboard로부터 입력받을 수 있어야 합니다.

구매하려는 상품코드를 선택하세요. [1:사과,2:오렌지,3:포도,4:딸기] ? 4
상품팩에 포함된 상품의 개수, 상품팩의 가격을 순서대로 입력하세요 (구분자:,) ? 2,4000
구매하려는 상품팩의 개수를 입력하세요 ? 3

B. 입력받은 값을 Product 클래스에 담아서 반환하는 getProduct 메소드를 작성해야 합니다.

```
public Product getProduct();
```

C. Product 클래스를 받아서 해당 상품팩의 재고량을 체크하고, 구매 가능한 경우 지불해야 할 총 금액을 계산하여 반환하는 calcTotalPrice 메소드를 작성해야 합니다. 재고량이 부족할 경우에는 -1을 반환합니다.

```
public double calcTotalPrice(Product product);
```

D. 아래와 같이 결과를 출력해야 합니다.

3개의 딸기 상품팩을 구매하셨습니다.
딸기 한 개의 가격은 2000원이며, 지불해야 할 총 금액은 12000원입니다.

해당 상품팩에 대한 재고량이 부족할 경우 결과는 다음과 같이 출력해야 합니다. (3점)

재고가 없어 3개의 딸기 상품팩에 대한 구매가 불가합니다. 죄송합니다.

<<처리 결과>>

구매하려는 상품코드를 선택하세요. [1:사과,2:오렌지,3:포도,4:딸기] ? 4
상품팩에 포함된 상품의 개수, 상품팩의 가격을 순서대로 입력하세요
(구분자:,) ? 2, 4000
구매하려는 상품팩의 개수를 입력하세요 ? 3
3개의 딸기 상품팩을 구매하셨습니다.
딸기 한 개의 가격은 2000원이며, 지불해야 할 총 금액은 12000원입니다.

<<참고 및 제한 사항>>

- 상품별 상품명을 정의한 nameOfProduct 배열을 사용합니다.

  private static String nameOfProduct[] = { “사과” ,” 오렌지” ,” 포도” ,” 딸기” };

- 상품별 재고 수량을 정의한 totalStockOfPack 배열을 사용합니다.

  private static int totalStockOfPack[] = {3, 5, 20, 1};

- 상품 한 개의 가격 : 상품팩의 가격 / 상품팩에 포함된 상품의 개수, 나누어 떨어지지 않는 경우 Math.round() 를 이용하여 반올림합니다.
- 지불 금액 : 상품팩 한개의 가격 * 구매 상품팩의 개수
- **Product 클래스는 새로 구현**하고, **Purchase 클래스는 주어진 소스의 주석 부분을 채워**서 완성합니다.

문제 5. Prob5.java 파일을 열어서 다음의 조건을 구현하여 완성하십시오.

```
조건1: main 메소드에서 MySum 객체를 출력하면 객체 생성시에 전달한 두 정수의 덧셈 결
    과를 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 오
    버라이딩합니다.

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 6. 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.

<< Prob6 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
```