문제 **1**

두 개의 정수를 전달받아서, 두 수의 차의 절대값을 계산하여 출력하는 메소드와 이 메소드를 호 출하는 main메소드를 정의하라. 메소드 호출 시 전달되는 값의 순서에 상관없이 절대값이 계산 되어서 출력되어야 한다.

문제 **2**

전달된 값이 소수인지 아닌지를 판단하여, 소수인 경우 true, 소수가 아닌 경우 false 를 반환하 는 메소드를 정의하고, 이를 이용해서 1이상 100이하의 소수를 전부 출력할 수 있도록 main 메 소드를 정의하라.

문제 **3**
다음 조건을 만족하는 클래스를 정의하라
-어린아이가 소유하고 있는 구슬의 개수 정보를 담을 수 있다
- 생성자를 이용해서 구슬의 개수 정보를 초기화할 수 있는 클래스 - 놀이를 통한 구슬의 주고받음을 표현하는 메소드가 존재
- 어린이의 현재 구슬의 수를 출력하는 메소드가 존재

두 번째 구슬의 주고받음을 표현하는 메소드란,

```
class ChildProperty {
  ...
public void obtainBread(ChildProperty child, int obtainCount) { //child의 loseBead 호출

//내 구슬의 개수 증가 }

public int loseBead(int loseCount) {
//현재 개수보다 뺏기는 구슬(loseCount)의 수가 많으면 현재 개수 0

//아닌 경우엔 현재 개수에서 loseCount를 뺀만큼으로 }

}
```

클래스의 설계가 끝나면
main 메소드에서 구슬을 15개, 9개씩 가진 두 명의 아이를 생성하고 “1차 게임에서 어린이1가 어린이2의 구슬 2개를 뺏는다”
“2차 게임에서 어린이2가 어린이1의 구슬 7개를 뺏는다”
의 상황을 main 메소드 내에서 시뮬레이션하자

되도록이면 자바의 이름 규칙을 지켜줬으면

문제**4**

- The Employee class has three (3) instance variables
  - **String** *name*, **int** *employeeNum*, **String** *department*.
- Create a constructor that accepts the *name* and *employeeNum*. The *department* will be set to "No Dept."
- Create an **accessor** and **mutator** to get and set the *department*.
- Create an **equals(Object obj)** method that returns true if an Employee has the same *name* and *employeeNum* as another.
- Create a **toString()** method that return the employee's name and employeeNum.

- The Manager class will extend the Employee class.
- Create two (2) instance variables in the Manager class
  - **int officeNum, String team**
- Create a Constructor that accepts *name, employeeNum, officeNum* and *team* and then creates a Manager object.
- Create an **equals(Object obj)** method that returns true if a Manager has the same *name, employeeNum, officeNum* and *team* as another. (use the superclass's equals())
- Create a **toString()** method that return the Manager's name and employeeNum, location and what he does. (location is department and officeNum)

- The Engineer class will extend the Employee class.
- Create two (2) instance variables in the Engineer class
  - **String workZone, String project**
- Create a Constructor that accepts *name, employeeNum, workZone, project* and creates a Manager object.
- Create an **equals(Object obj)** method that returns true if an Engineer has the same *name, employeeNum, officeNum* and *workZone* as another. (use the superclass's equals())
- Create a **toString()** method that return the Engineer's name and employeeNum, location and what he does. (location is department and workZone)

- Test your classes with the following

```
public class Company {

    public static void main(String[] args) {
        Employee emp1 = new Manager("John Smith",1234,25,"door and panels");
        Employee emp2 = new Engineer("Peter Anderson",1852,"Fabrication #7","door and panels");
        Manager emp5 = new Manager("John Smith",1234,25,"team 7");

        Employee emp3 = new Employee("Jane Roberts",2345);
        Employee emp4 = new Employee("John Smith",1234);

        System.out.println(emp1);
        System.out.println(emp2);
        System.out.println(emp3);
        System.out.println(emp4);

        System.out.println(emp1.equals(emp2));
        System.out.println(emp1.equals(emp5));
        System.out.println(emp1.equals(emp4));
    }

}
```

```
<terminated> Company [Java Application] C:\Program Files\Java\jdk1.8.0_4
Name: John Smith
Emp#: 1234
Location: Management at officeNum
I manage the "door and panels" team.

Name: Peter Anderson
Emp#: 1852
Location: Engineering at zone fabrication #7
I work on the "door and panels" project.

Name: Jane Roberts
Emp#: 2345

Name: John Smith
Emp#: 1234

false
true
false
```