Exercise 1  JML Queue Specifications (12p)

Please give a JML specification of a queue by completing the specification (not implementation) of the following class, including invariants, preconditions, and postconditions:

```
public class Queue {
    Object [] array;
    int size;
    int first;
    int next;

    public Queue(int max) {
        // ...
    }

    public int size() {
        // ...
    }

    public void enqueue(Object x) {
        // ...
    }

    public Object.dequeue() {
        // ...
    }
}
```

Exercise 2  JML List Specifications (8p)

Please give a JML specification for the following class. In particular, specify the method size().

```
public class List {
    Object first;
    List rest;
```
```java
public List(Object first, List rest) {
    this.first = first;
    this.rest = rest;
}

public int size() {
    // ...
}
}
```

**Exercise 3   JML Basics (10p)**

1. JML only allows pure functions in expressions. Is there a good reason for this? Why?
2. Can you change JML so that it would allow arbitrary functions in expressions? What is the effect of this change on runtime monitoring and on formal verification?
3. Are pure methods really pure? Give an example of a pure method and a JML specification such that the pure method changes the result of one of the JML expressions, or explain why there is no such example.

**Exercise 4   Formal Semantics of Java (20p)**

Develop a formal semantics for a Java fragment that includes all relevant detail to give a semantics to the following code snippet

```java
try {
    if (++x < x+1) {
        x = 2*x;
        throw e;
    } else {
        x = 0;
    }
} catch (SomeException t) {
    x = x*x;
}
```

The semantics does not need to capture all of Java but only programs that include the elements that you find in the above example program.