The goal of this project is to extend the base *j*-- language by adding some basic Java operations (on primitive integers) to the language. Supporting these operations requires studying the *j*-- compiler in its entirety, if only cursorily, and then making slight modifications to it. Notice that many of the operations have different levels of precedence, just as `*` has a different level of precedence in *j*-- than does `+`. These levels of precedence are captured in the Java grammar (see Appendix C of our text); for example, the parser uses one method to parse expressions involving `*` and `/`, and another to parse expressions involving `+` and `-`.

**Problem 1.** (*Division Operation*) Download and unzip the base *j*-- compiler under some directory (we'll refer to this directory as `$j`) and make sure you are able to compile and run it — see Appendix A for information on what's in the *j*-- distribution. Next, follow the process outlined in Section 1.5 to implement the Java division operator `/`.

```
$ $j/j--/bin/j-- tests/Division.java
$ java Division 42 6
7
```

**Problem 2.** (*Remainder Operation*) Implement the Java remainder operator `%`.

```
$ $j/j--/bin/j-- tests/Remainder.java
$ java Remainder 42 13
3
```

**Problem 3.** (*Shift Operations*) Implement the Java shift operators: arithmetic left shift `<<`, arithmetic right shift `>>`, logical right shift `>>>`.

```
$ $j/j--/bin/j-- tests/LeftShift.java
$ java LeftShift 1 5
32
```

```
$ $j/j--/bin/j-- tests/RightShift.java
$ java RightShift 32 5
1
$ java RightShift -32 5
-1
```

```
$ $j/j--/bin/j-- tests/LogicalRightShift.java
$ java LogicalRightShift 32 5
1
$ java LogicalRightShift -32 5
134217727
```

**Problem 4.** (*Bitwise Operations*) Implement the Java bitwise operators: unary complement `~`, inclusive or `|`, exclusive or `^`, and `&`.

```
$ $j/j--/bin/j-- tests/Not.java
$ java Not 42
-43
```

```
$ $j/j--/bin/j-- tests/InclusiveOr.java
$ java InclusiveOr 3 5
7
```

```
$ $j/j--/bin/j-- tests/ExclusiveOr.java
$ java ExclusiveOr 3 5
6
```

```
$ $j/j--/bin/j-- tests/And.java
$ java And 3 5
1
```

**Problem 5.** (*Unary Plus Operation*) Implement the Java unary plus operaor `+`.

```
$ $j/j--/bin/j-- tests/UnaryPlus.java
$ java UnaryPlus -42
-42
```

### Files to Submit

1. `j--.zip` (*j*-- source tree as a single zip file)
2. `report.txt` (project report)

**Before you submit:**

- Make sure you create the zip file `j--.zip` such that it only includes the source files and not the binaries, which can be done on the terminal as follows:

  ```
  $ cd $j/j--
  $ ant clean
  $ cd ..
  $ tar -cvf j--.tar j--/*
  $ gzip j--.tar
  ```

- Make sure the files `$j/j--/lexicalgrammar` and `$j/j--/grammar` are updated with the syntactic changes you have made to the *j*-- language.
- Make sure your report isn't too verbose, doesn't contain lines that exceed 80 characters, and doesn't contain spelling/grammatical mistakes