Exercise 5

- Inherit `Unit` class below to solve problem 1~3.
- Use `ArrayList<Unit>` to save each problem's units.
- All problems should print out all `Unit`s with `toString()` method when you entered `list` as an input.
  - See input & output example of each problems for detail.
- All problems can read input endlessly, but when they reads `QUIT`, they quit.
- All integer calculation results do not exceed range of `int`.

```
class Unit {
 protected String operator;
 protected int left;
 protected int right;

 public Unit(int left, iint right, String operator) {
 // TODO: implement me

 calculate(); // Dynamic binding
 }

 protected void calculate() {
 // TODO: implement me
 }
}
```

Method description

- Constructor should save parameter values, and print out calculation results like examples below.
- `calculate()` will be called by constructor, calculate the result value, and save it in the variable `result`.
- `toString()` prints out the result from `calculate()` like output examples below.

1. Arithmetic Unit

1-1. Description

```
class ArithmeticUnit extends Unit {
 protected int result;

 public ArithmeticUnit(int left, int right, String operator) {
 super(left, right, operator);
 }

 @Override
 protected void calculate() {
 // TODO: implement me
 }

 @Override
 public String toString() {
 // TODO: implement me
 }
}
```

- A `Unit` that calculates arithmetic operation(`+`, `-`, `/`, `%`, `*`, `^` (power)).
- Implement `ArithmeticUnit` that reads 2 integer and calculate them.

1-2. Input example

```
2 4 +
3 5 *
list
6 4 /
list
16 5 %
3 10 -
3 6 ^
QUIT
```

- Two `int` range numbers and an operator is entered with intervals to ` ` (space).

1-3. Output example

```
2+4=6
3*5=15
2+4=6
3*5=15
6/4=1
2+4=6
3*5=15
6/4=1
16%5=1
3-10=-7
3^6=729
```

2. Compare Unit

2-1. Description

```
class CompareUnit extends Unit {
 protected boolean result;

 public CompareUnit(int left, int right, String operator) {
  // TODO: implement me
 }

 @Override
 protected void calculate() {
  // TODO: implement me
 }

 @Override
 public String toString() {
  // TODO: implement me
 }
}
```

- A `Unit` that compares(`==` , `!=` , `<=` , `<` , `>` , `>=`) two integers.
- Implement `CompareUnit` that reads 2 integer and calculate them.

2-2. Input example

```
2 4 <=
list
3 5 ==
6 4 >
6 15 >=
3 10 !=
QUIT
```

- Two `int` range numbers and an operator is entered with intervals to ` ` (space).

2-3. Output example

```
2<=4 : true
2<=4 : true
3==5 : false
6>4 : true
6>=15 : true
3!=10 : true
```

3. Bit Unit

3-1. Description

```
class BitUnit extends Unit {
 protected int result;

 public BitUnit(int left, int right, String operator) {
  // TODO: implement me
 }

 @Override
 protected void calculate() {
  // TODO: implement me
 }

 @Override
 public String toString() {
  // TODO: implement me
 }
}
```

- A `Unit` that calculates bit operation(`|` , `&` , `>>` , `<<`).
- Implement `CompareUnit` that reads 2 integer and calculate them.

3-2. Input example

```
1 1 &
2 1 |
16 2 >>
8 2 <<
list
QUIT
```

- Two `int` range numbers and an operator is entered with intervals to ` ` (space).

3-3. Output example

```
1&1=1
2|1=3
16>>2=4
8<<2=32
1&1=1
2|1=3
16>>2=4
8<<2=32
```