This exercise gives you practice with writing iterators and visitors. Go to office hours if you have trouble with this exercise.

1. Implement IntRepIterator, an iterator that produces the decimal digits of an IntRep. IntRepIterator would be created by IntRep’s iterator() method, shown below:

```java
public class IntRep implements BigInt {
    private int n;
    /**
     * @return iterator that yields the digits of the decimal representation of this number, starting with the least significant digit. If this == 0, the iterator yields no digits at all. For example, if this==3115, then the iterator yields 5, 1, 1, 3.
     */
    public Iterator<Integer> iterator() {
        return new IntRepIterator(n);
    }
}
```

Recall that an Iterator should implement the following interface:

```java
interface Iterator<T> {
    boolean hasNext();
    T next();
    // java.util.Iterator has a remove() method too, but don’t implement that
}
```

2. Implement IsConstant, a visitor that tests whether an Expr expression contains only constants and operators (but no variables). IsConstant would be created by a static method like the following:

```java
public class Exprs {
    /**
     * @return true iff e has no variables in it, so that e.eval() is guaranteed to return a value
     */
    public static boolean isConstant(Expr e) {
        return e.accept(new IsConstant());
    }
}
```

Recall that a visitor for Expr should implement the following interface:

```java
interface Visitor<T> {
    T onConst(Const e);
    T onVar(Var e);
    T onPlus(Plus e);
    T onPow(Pow e);
    // ...and other methods, but don’t worry about implementing them
}
```