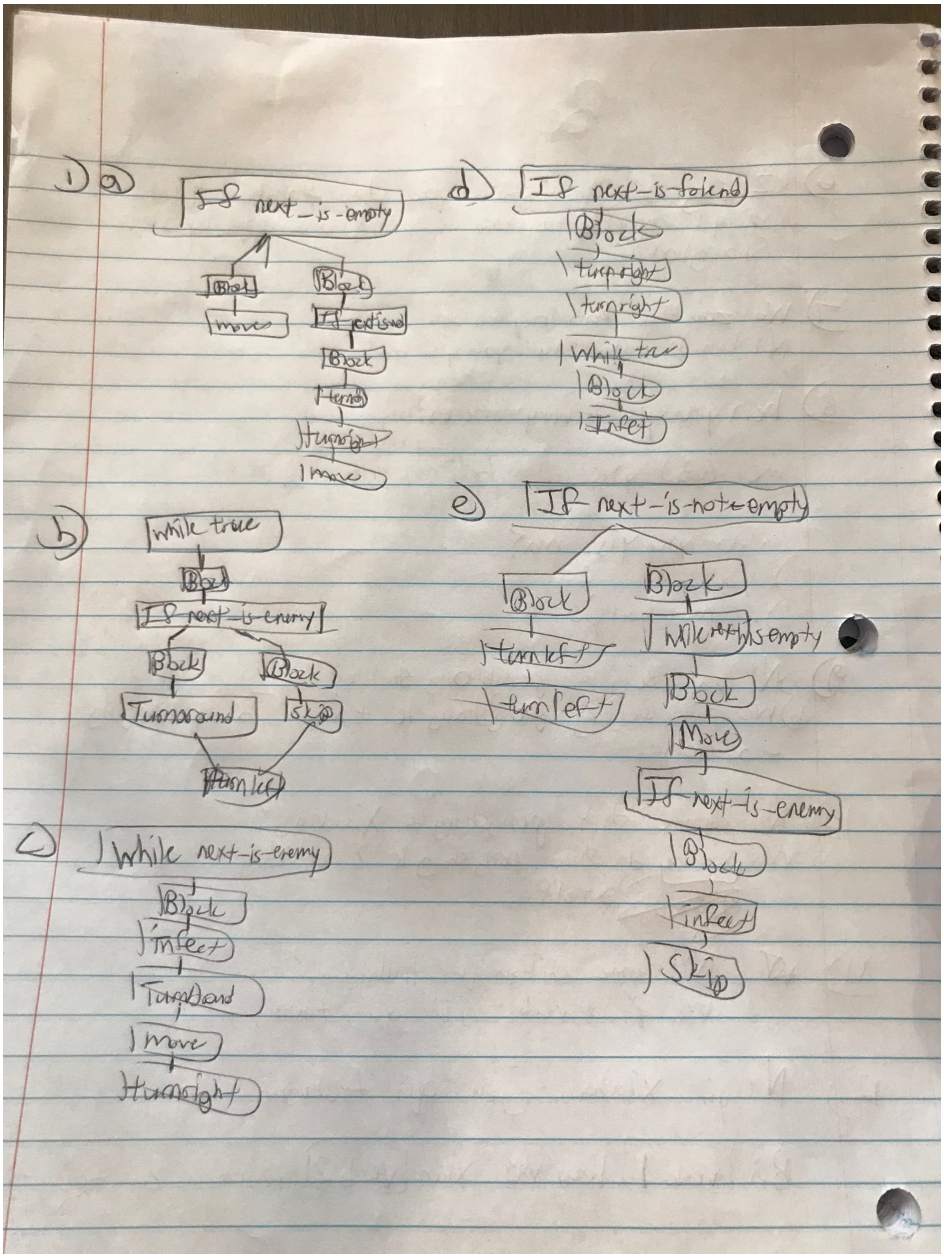
Homework 22

Shyam Sai Bethina

08 March 2022

1)



2)

```
public static int countOfPrimitiveCalls(Statement s) {
    int count = 0;
    switch (s.kind()) {
        case BLOCK: {
            /*
             * Add up the number of calls to primitive instructions in each
             * nested statement in the BLOCK.
             */

            // TODO - fill in case

            int length = s.lengthOfBlock();
            for (int i = 0; i < length; i++) {
                Statement child = s.removeFromBlock(i);
                count += countOfPrimitiveCalls(child);
                s.addToBlock(i, child);
            }

            break;
        }
        case IF: {
            /*
             * Find the number of calls to primitive instructions in the
             * body of the IF.
             */

            // TODO - fill in case
            Statement child = s.newInstance();
            Statement.Condition condition = s.disassembleIf(child);
            count = countOfPrimitiveCalls(child);
            s.assembleIf(condition, child);

            break;
        }
```

```
    case IF_ELSE: {
        /*
         * Add up the number of calls to primitive instructions in the
         * "then" and "else" bodies of the IF_ELSE.
         */

        Statement childIf = s.newInstance();
        Statement childElse = s.newInstance();

        Statement.Condition c = s.disassembleIfElse(childIf, childElse);
        count = countOfPrimitiveCalls(childIf)
                + countOfPrimitiveCalls(childElse);
        s.assembleIfElse(c, childIf, childElse);

        break;
    }
    case WHILE: {
        /*
         * Find the number of calls to primitive instructions in the
         * body of the WHILE.
         */

        Statement child = s.newInstance();
        Statement.Condition condition = s.disassembleWhile(child);

        count = countOfPrimitiveCalls(child);
        s.assembleWhile(condition, child);

        break;
    }
```

```
        }
        case CALL: {
            /*
             * This is a leaf: the count can only be 1 or 0. Determine
             * whether this is a call to a primitive instruction or not.
             */

            String label = s.disassembleCall();
            if (label.equals("turnright") || label.equals("move")
                    || label.equals("infect") || label.equals("turnleft")
                    || label.equals("skip")) {
                count++;
            }
            s.assembleCall(label);

            break;
        }
        default: {
            // this will never happen...can you explain why?
            break;
        }
    }
    return count;
}
```