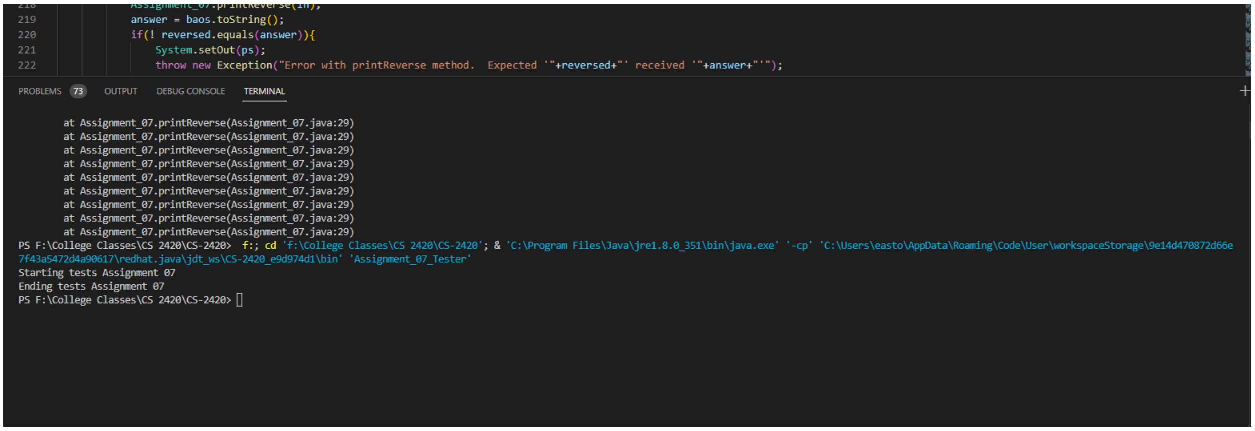
Easton Jackson

Assignment 7 tester

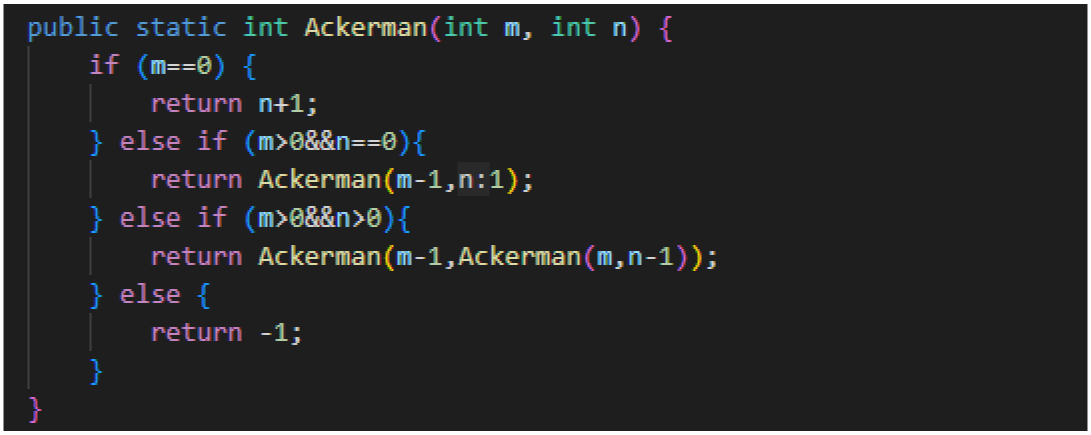


```
            Assignment_07.printReverse(in);
            answer = baos.toString();
            if(! reversed.equals(answer)){
                System.setOut(ps);
                throw new Exception("Error with printReverse method.  Expected '"+reversed+"' received '"+answer+"'");
```

```
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
        at Assignment_07.printReverse(Assignment_07.java:29)
PS F:\College Classes\CS 2420\CS-2420>  f:; cd 'f:\College Classes\CS 2420\CS-2420'; & 'C:\Program Files\Java\jre1.8.0_351\bin\java.exe' '-cp' 'C:\Users\easto\AppData\Roaming\Code\User\workspaceStorage\9e14d470872d66e
7f43a5472d4a90617\redhat.java\jdt_ws\CS-2420_e9d974d1\bin' 'Assignment_07_Tester'
Starting tests Assignment 07
Ending tests Assignment 07
PS F:\College Classes\CS 2420\CS-2420> 
```

7.21

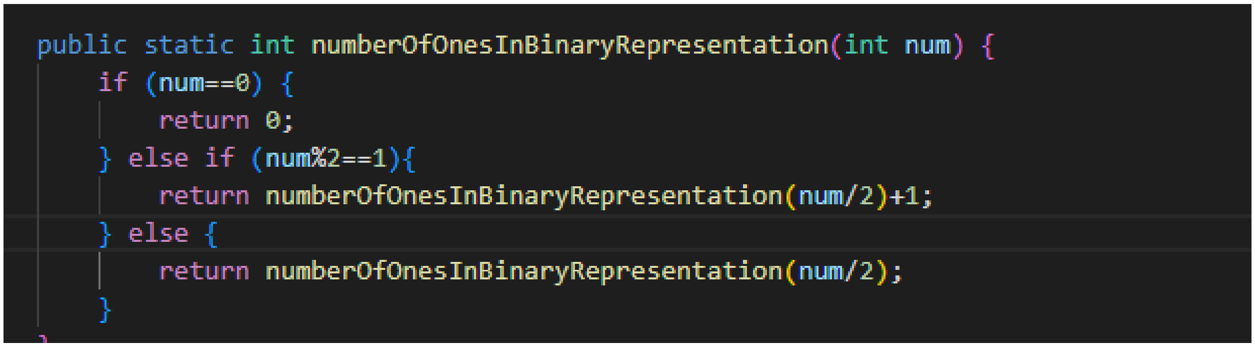
With this one, I did not struggle at all because the pseudo code is already in the book. It was pretty much just inserting it into the java syntax and hitting run.



```java
public static int Ackerman(int m, int n) {
    if (m==0) {
        return n+1;
    } else if (m>0&&n==0){
        return Ackerman(m-1,n:1);
    } else if (m>0&&n>0){
        return Ackerman(m-1,Ackerman(m,n-1));
    } else {
        return -1;
    }
}
```

7.23

This one took a little more messing with but mostly because I did not quite understand what the question was asking me to do at first. After some messing around I was able to get the tester to throw no errors. I just used the normal format that I had seen for base cases and then recursive calls.

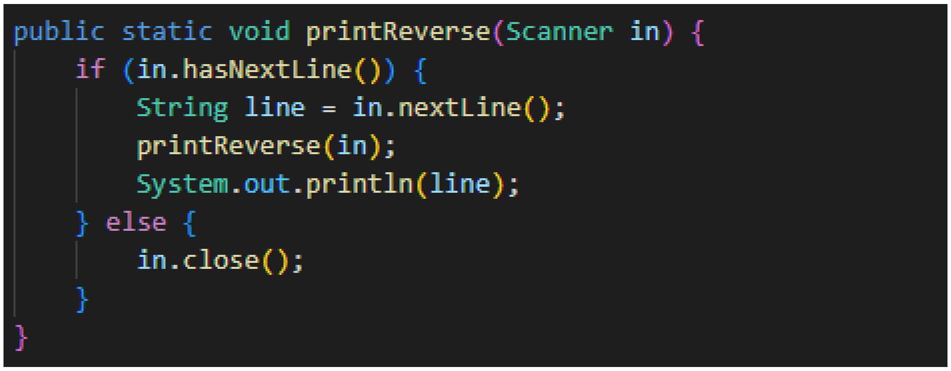


```java
public static int numberOfOnesInBinaryRepresentation(int num) {
    if (num==0) {
        return 0;
    } else if (num%2==1){
        return numberOfOnesInBinaryRepresentation(num/2)+1;
    } else {
        return numberOfOnesInBinaryRepresentation(num/2);
    }
}
```

Easton Jackson

7.36

This one was a lot more tricky because I have not messed with passing scanners to different methods before, but after messing around, I was able to find out that you could just close the scanner to end the recursion. After that it runs back through every instance and just prints the line that it found.



```java
public static void printReverse(Scanner in) {
    if (in.hasNextLine()) {
        String line = in.nextLine();
        printReverse(in);
        System.out.println(line);
    } else {
        in.close();
    }
}
```