1. What happens when the following is compiled/run?

class MyClass {

public static void main(String[] args) {

new MyClass();

}

MyClass() {

recurse("Hello");

}

String recurse(String s){

if(s==null) return null;  
int r = RandomNumbers.getRandomInt(); int n = s.length();  
if(r % 2 == 0)

return recurse(s.substring(0,n/2)); else {

return recurse(s.substring(n/2,n)); }

} }

1. Compiler error
2. Returns a null value
3. NullPointerException
4. **StackOverflowError**

Explain your answer

There is a stackoverflow error because the recursive function does not lead to the base case, therefore the recursion continues endlessly.

What happens when the following is compiled/run? You may assume that the method permute is implemented correctly elsewhere, and that it has the effect of randomly rearranging the characters of a String (for instance, on different runs of permute with input “events”, the return values could be, for example, “evtsen”, “eestnv” and “evenst”).

class MyClass {

public static void main(String[] args) {

new MyClass();

}

MyClass() {

recurse("Hello");

}

String recurse(String s){

if(s==null || s.equals("")) return "";  
int n = s.length();  
String t = permute(s); //rearrange characters of s return recurse(t);

\_\_\_\_\_ 2.

} }

1. Compiler error
2. Returns a null value
3. NullPointerException
4. **StackOverflowError**

Explain your answer

There is a stackoverflow error because the recursive function does not lead to the base case, therefore the recursion continues endlessly.