

**Question 0.** No answer given

**Question 1.**

*/\*\* = n, as a String, but with its digits reversed.*

*Precondition: n >= 0. \*/*

```
public static String rev(int n) {
    if (n < 10)
        return "" + n;
    return (n%10) + rev(n/10);
}
```

**Question 2.**

**A.** Add this declaration and comment:

*// Numbers 0..nextTag-1 have been used as tags,*

*// and number nextTag is the next one to use.*

```
private static int nextTag= 0;
```

Then add these statements to the body of the second constructor:

```
tag= nextTag;
nextTag= nextTag + 1;
```

**B.** `this(1, 0);`

```
C. public boolean equals(Object obj) {
    if (!(obj instanceof Bee))
        return false;
    Bee b= (Bee) obj;
    return month == b.month &&
        year == b.year;
}
```

**Question 3.**

*/\*\* = number of female ancestors of bee b \*/*

```
public static int femAnc(Bee b) {
    if (!(b instanceof MaleBee) &&
        !(b instanceof FemaleBee)) {
        return 0;
    }
    // b is not a queen
    if (b instanceof MaleBee) {
        return 1 +
            femAnc(((MaleBee) b).getMother());
    }
    // b is a female but not the queen bee
    return 1 +
        femAnc(((FemaleBee) b).getMother()) +
        femAnc(((FemaleBee) b).getFather());
}
```

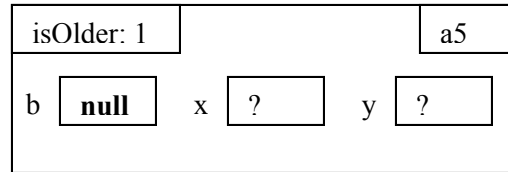
**Question 4. (a)** 1. Draw the frame for the call.

2. Assign the argument values to the parameters.

3. Execute the function body.

4. Erase the frame and return the value of the expression on the return statement.

**(b)**



**Question 5. (a)** When the frame for the call is drawn, before the argument values are assigned to the parameters.

**(b)** An argument is an expression that appears within the parentheses of a method call.

**(c)** The apparent class is `Bee`; the real class is `MaleBee`.

**(d)** `b.getTag()` is legal.

`b.getMother()` is illegal because function `getMother` is not available in apparent class `Bee`. Since the real class is `MaleBee`, use this expression instead:

```
((MaleBee)b).getMother()
```

`b.getFather()` is illegal because function `getMother` is not available in apparent class `Bee`. It cannot be made legal because the real class is `MaleBee`, and that class does not have function `getFather` in it.