Question 1a: The value of a local variable named “i” has no direct relationship with that of a variable named “i” in its caller.

Answer: True

Reasoning:

The variable “i” in both methods only exist within the methods they were declared in. However, the value of the variable can be returned.

```
public class Assignment3 {

	public static void main(String[] args) {
		String i = "This variable named \"i\" is in the main method.";
		int x = 8;
		System.out.println(i);
		question1(x);
	}

	private static void question1(int x) {
		String i = "This variable, also named \"i\" is in the question1 method.";
		System.out.println(i);
		System.out.println("So, it is true that these two variables have no direct relationship.");
		System.out.println("");
		System.out.println("However, a variable defined in the calling method has a direct"
				+ "relationship with the parameters in this”
				+ “method.");
		System.out.println("This number was defined in the main method and given as a”
				+ parameter: " + x);
	}
}
```

Question 1b: The value of a parameter named x has no direct relationship with that of a variable named x in its caller.

Answer: True

Reasoning:

The values passed into the parameter of a method call are copies of the variable's value in the calling method. The two variables do not modify eachother. The only correlation is when a variable is passed into the arguments of a method call, which its copied value gets assigned to the variable declaration in the new method. See code below:

```
Public static void main(String[] args) {
        int x = 8;
        int y = 16;
        System.out.println(myMethod(x));
        System.out.println(myMethod(y));
}

public myMethod(int x) {
        int y = x * 2;
        return y;
}
```

//==========

Question 2:

Output:

snitch: x = 4004, y = 1001
quaffle: x = 2003, y = 1, z = 1001
bludger: x = 1001, y = 2001, z = 2003

Notes:

bludger(2001)
y = 2001
x = 2
z = 2003
x = 1001
quaffle(2003,2001)
z = 1001
y = 1
snitch(4004,2001)
y = 1001