Q1: Consider the following function:

```c
foo() {
    char buf[...];
    strncpy(buf, readUntrustedInput(), sizeof(buf));
    printf(buf); //vulnerable
}
```

Suppose `readUntrustedInput()` provides an attack string of the form:

... `attackString%n` ... `<shellcode>` ...

How might we be able to use `%n` to overwrite the saved EIP (aka RET) on the stack? (You don’t need to give the exact attack; just brainstorm about the general approach you might try.)

As a reminder, here’s what the stack looks like for this program:

<table>
<thead>
<tr>
<th>Saved FP</th>
<th>ret/IP</th>
<th>&amp;buf</th>
<th>buf</th>
<th>Saved FP</th>
<th>ret/IP</th>
<th>Caller’s frame</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Printf’s frame

Foo’s frame

Addr 0xFF...F

Q2: What might be a good value for a stack canary?