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:

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

How might we be able to use one or more “%n”s 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.)

Here’s what the stack looks like for this program:
Q2: What might an attacker be able to accomplish even if they cannot execute code on the stack?