As part of your daily routine, you are browsing through the news and status updates of your friends on the social network FaceSpace.

(a) While looking for a particular friend, you notice that the text you entered in the search string is displayed in the result page. Next to you sits a suspicious looking student with a black hat who asks you to try queries such as

```html<br>&lt;script&gt;alert(42);&lt;/script&gt;```

in the search field. What is this student trying to test?

(b) The student also asks you to post the code snippet to the wall of one of your friends. How is this test different from part (a)?

(c) The student is delighted to see that your browser spawns a JavaScript pop-up in both cases. What are the security implications of this observation? Write down an example of a malicious URL that would exploit the vulnerability in part (a).

(d) Why does an attacker even need to bother with XSS? Wouldn’t it be much easier to just create a malicious page with a script that steals all cookies of all pages from the user’s browser?
Question 2  SQL Injection

(a) Explain the bug in this PHP code. How would you exploit it? Write what you
would need to do to delete all of the tables in the database.

    $query = "SELECT name FROM users WHERE uid = $UID";
    // Then execute the query.

    (Here, $UID represents a URL parameter named UID supplied in the HTTP request.
The actual representation of such a value in PHP is a bit messier than we've shown
here. We leave out the syntactic details so we can focus on the functionality.)

(b) How does blacklisting work as a defense? What are some difficulties with blacklist-
ing?

(c) What is the best way to fix this bug?

Question 3  Cross Site Request Forgery (CSRF)  (10 min)

In a CSRF attack, a malicious user is able to take action on behalf of the victim. Consider
the following example. Mallory posts the following in a comment on a chat forum:

    <img src="http://patsy-bank.com/transfer?amt=1000&to=mallory"/>

Of course, Patsy-Bank won't let just anyone request a transaction on behalf of any given
account name. Users first need to authenticate with a password. However, once a user
has authenticated, Patsy-Bank associates their session ID with an authenticated session
state.

(a) Explain what could happen when Victim Vern visits the chat forum and views
Mallory’s comment.

(b) What are possible defenses against this attack?

Question 4  Session Fixation  (15 min)

Some web application frameworks allow cookies to be set by the URL. For example, visiting the URL

    http://foobar.edu/page.html?sessionid=42.

will result in the server setting the sessionid cookie to the value “42”.

(a) Can you spot an attack on this scheme?

(b) Suppose the problem you spotted has been fixed as follows. foobar.edu now es-
tablishes new sessions with session IDs based on a hash of the tuple (username,
time of connection). Is this secure? If not, what would be a better approach?