# Exercises crypto two

## Exercise 1: Hashing basics

Thinking about these questions will help you when doing the next questions. Write some good notes for yourself on this topic.

**Part 1:**
What is hashing? When is it used?

**Part 2:**
What is salts? How does it work in general?

**Part 3:**
How do you make a strong hash in a modern system? What are your considerations?

## Exercise 2: Cracking hashed passwords with John the Ripper

**Download the jumbo-edition binaries at:**
https://www.openwall.com/john/

**Download the GUI tool at:**
https://openwall.info/wiki/john/johnny
*(In settings, point to the /run/john.exe file from the jumbo folder)*

Download the files *"passwords_md5.txt"* and *"passwords_salted_sha1.txt"* from the LearnIT page.

**Part 1:**
In options, set the hash format to "Raw-MD5". Crack atleast 30 passwords from the *"passwords_md5.txt"* file. Note roughly how long it takes to obtain the 30 first passwords. Look at the obtained cleartext passwords. Why do you think these passwords were the fastest to crack?

**Part 2:**
In the options, set the hash format to Salted-SHA1, and attack the *"passwords_salted_sha1.txt"* file. Note again roughly how long it takes to obtain the 30 first passwords.

**Bonus:** This method uses raw brute force, to crack the hashes. What other methods can you think of, that might speed up the process partially?

# Exercise 3: Cracking hashed passwords with Rainbow tables

**Part 1:**
Just to make sure everyone is on the same page:
https://en.wikipedia.org/wiki/Rainbow_table

Online rainbow table look-up service: https://crackstation.net/

Use the file *"hashes_md5.txt"* containing hashes just like in exercise 2. You should be able to crack all hashes.
Can you crack any of the hashed passwords that use salts from *"passwords_salted_sha1.txt"* using rainbow tables?

**Part 2:**
How does salts impact cracking of passwords with rainbow tables?

# Exercises crypto two

## Exercise 1: Hashing basics

Thinking about these questions will help you when doing the next questions. Write some good notes for yourself on this topic.

**Part 1:**
Hashing is a form of one-way encryption, which is used to check equality without exposing the content - typically used in password authentication.

**Part 2:**
A salt is a bit of extra content which is stored with the hash, and used while hashing. The point is to safeguard the content of the hash against previously computed hashes or existing tables/databases.
**Part 3:**
Use a good algorithm which has not been beaten, and apply salts.

## Exercise 2: Cracking hashed passwords with John the Ripper

When checking “show only cracked” in johnny, the output should contain some cleartext passwords. It may take upwards of 15 minutes to crack the first 30 passwords (from md5 pw file). Note that it might not be the same exact output for you.

The fatest passwords to crack are the one that are easiest to compute sequentially - like “aaaa” etc. In other terms, they are the less complex and shorter passwords.

The salted passwords should not take significantly longer to compute.

You could use a Rainbowtable or a wordlist, or some guessing, to crack the passwords faster.

## Exercise 3: Cracking hashed passwords with Rainbow tables

**Part 1:**
You should get 10 cleartext passwords from the first file, and nothing from the file using salts.

**Part 2:**
It makes rainbow table attacks infeasible, because there is a huge amount of possible salts for each single hash, and the salts are not shared. There are 2^n more options to compute, where n is the size of the salt alphabet.