

Assignment 2 (Lab 4)

1.

```
amr@AmrPC:~$ echo -n "testuser:password123" > testpass.txt
amr@AmrPC:~$ openssl passwd -1 "password123" > hash.txt
amr@AmrPC:~$ cat -v hash.txt
$1$ch0iIWCN$olu8jJeGWuvJdVv0mSHct0
amr@AmrPC:~$

amr@AmrPC:~$ echo "testuser:$(cat hash.txt)" > ex1.txt
amr@AmrPC:~$ cat -v ex1.txt
testuser:$1$ch0iIWCN$olu8jJeGWuvJdVv0mSHct0
amr@AmrPC:~$ |

amr@AmrPC:~/john/run$ ./john --format=md5crypt-long --wordlist=10k-most-common.txt ex1.txt
./john --show ex1.txt
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
password123 (testuser)
1g 0:00:00:00 DONE (2025-11-07 20:41) 2.222g/s 22224p/s 22224c/s 22224C/s booper..password123
Use the "--show" option to display all of the cracked passwords reliably
Session completed
testuser:password123

1 password hash cracked, 0 left
```

2.

```
amr@AmrPC:~/john/run$ echo "user2:$(openssl passwd -1 'P@ssw0rd2023!')" > ../ex2.txt
amr@AmrPC:~/john/run$ cat ../ex2.txt
user2:$1$nqwp3Ykg$WxjuzFPx00xE0MaiDeeAf0
amr@AmrPC:~/john/run$ ./john --session=ex2inc --incremental=Alpha ../ex2.txt

amr@AmrPC:~/john/run$ printf '%s\n' 'P@ssw0rd2023!' 'Password123!' 'P@ssword1' 'Admin2023!'
> small_wordlist.txt
./john --format=md5crypt-long --wordlist=small_wordlist.txt --rules ex2.txt
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt-long, crypt(3) $1$ (and variants) [MD5 32/64])
Will run 8 OpenMP threads
Press 'q' or Ctrl-C to abort, 'h' for help, almost any other key for status
Enabling duplicate candidate password suppressor using 256 MiB
P@ssw0rd2023! (user2)
1g 0:00:00:00 DONE (2025-11-07 21:41) 3.030g/s 193.9p/s 193.9c/s 193.9C/s P@ssw0rd2023!..P@
ssword1a
Use the "--show" option to display all of the cracked passwords reliably
Session completed
amr@AmrPC:~/john/run$ ./john --show ex2.txt
user2:P@ssw0rd2023!

1 password hash cracked, 0 left
```

3.

```
amr@AmrPC:~/john/run$ cat << 'EOF' > custom.rules
[List.Rules:Custom]
> c
> $2023
> $!
> EOF
amr@AmrPC:~/john/run$ cat custom.rules
[List.Rules:Custom]
c
$2023
$!
amr@AmrPC:~/john/run$ cat custom.rules
[List.Rules:Custom]
c
$2023
$!
amr@AmrPC:~/john/run$ printf '%s\n' 'password' 'admin' 'letmein' > wordlist.txt
amr@AmrPC:~/john/run$ ls -la wordlist.txt
-rw-r--r-- 1 amr amr 23 Nov  7 21:53 wordlist.txt
amr@AmrPC:~/john/run$ echo "---- wordlist.txt ----"
---- wordlist.txt ----
amr@AmrPC:~/john/run$ echo "---- wordlist.txt ----"
---- wordlist.txt ----
amr@AmrPC:~/john/run$ cat wordlist.txt
password
admin
letmein
```

```
amr@AmrPC:~/john/run$ echo "-----"
echo "user3:$(openssl passwd -1 'Password2023!')" > ex3.txt
-----
amr@AmrPC:~/john/run$ ls -la ex3.txt
-rw-r--r-- 1 amr amr 41 Nov  7 21:55 ex3.txt
amr@AmrPC:~/john/run$ echo "---- ex3.txt ----"
---- ex3.txt ----
amr@AmrPC:~/john/run$ cat ex3.txt
user3:$1$.xHoDxMf$joSKihw.82JBmQ74wh7L3/
amr@AmrPC:~/john/run$ echo "-----"
-----
```

```
1 password hash cracked, 0 left
Backup saved to custom.rules.bak - contents:
[List.Rules:Custom]
# Capitalize first letter
c
# Append year digits (one append command per digit)
$2
$0
$2
$3
# Add special char
$!
```