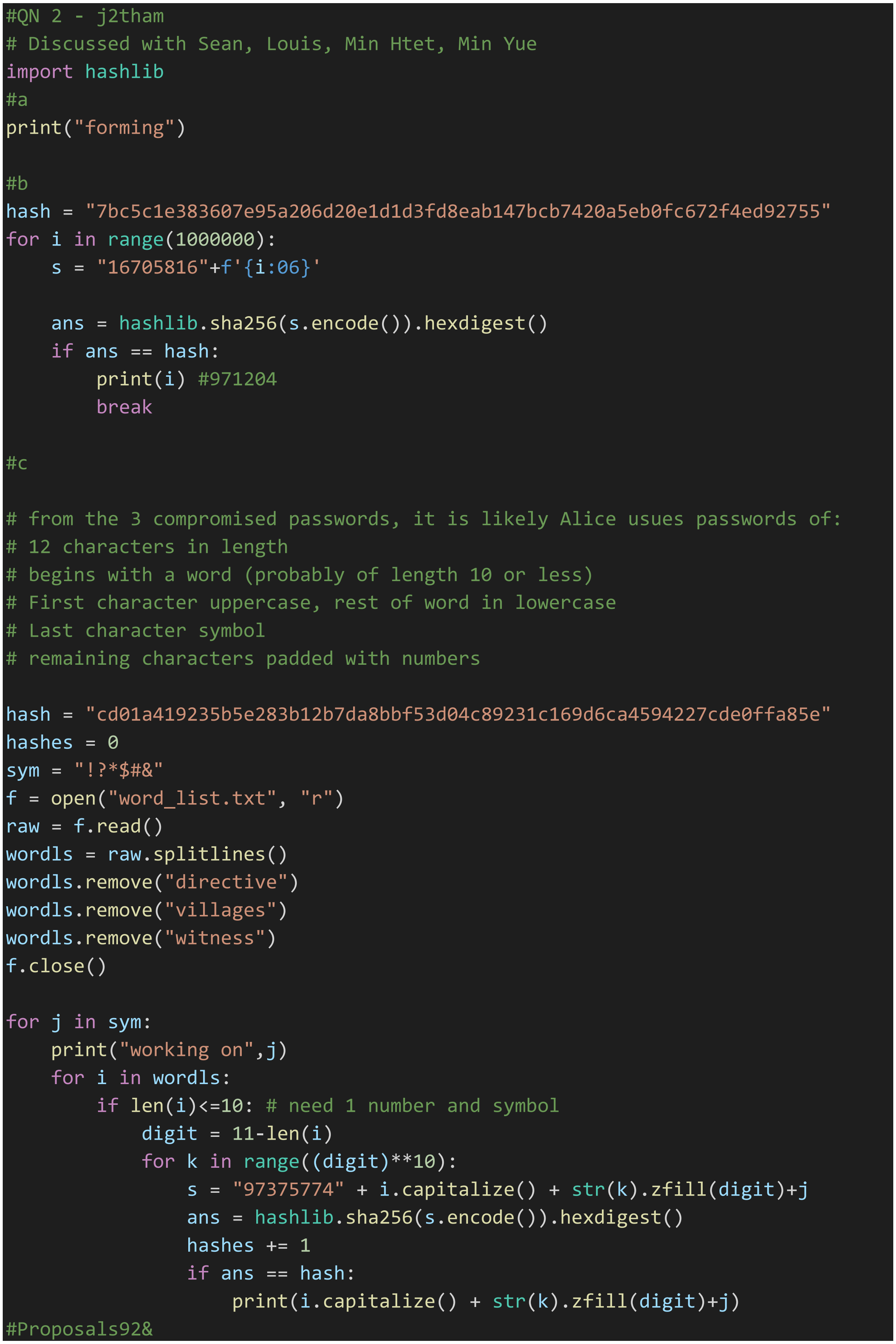
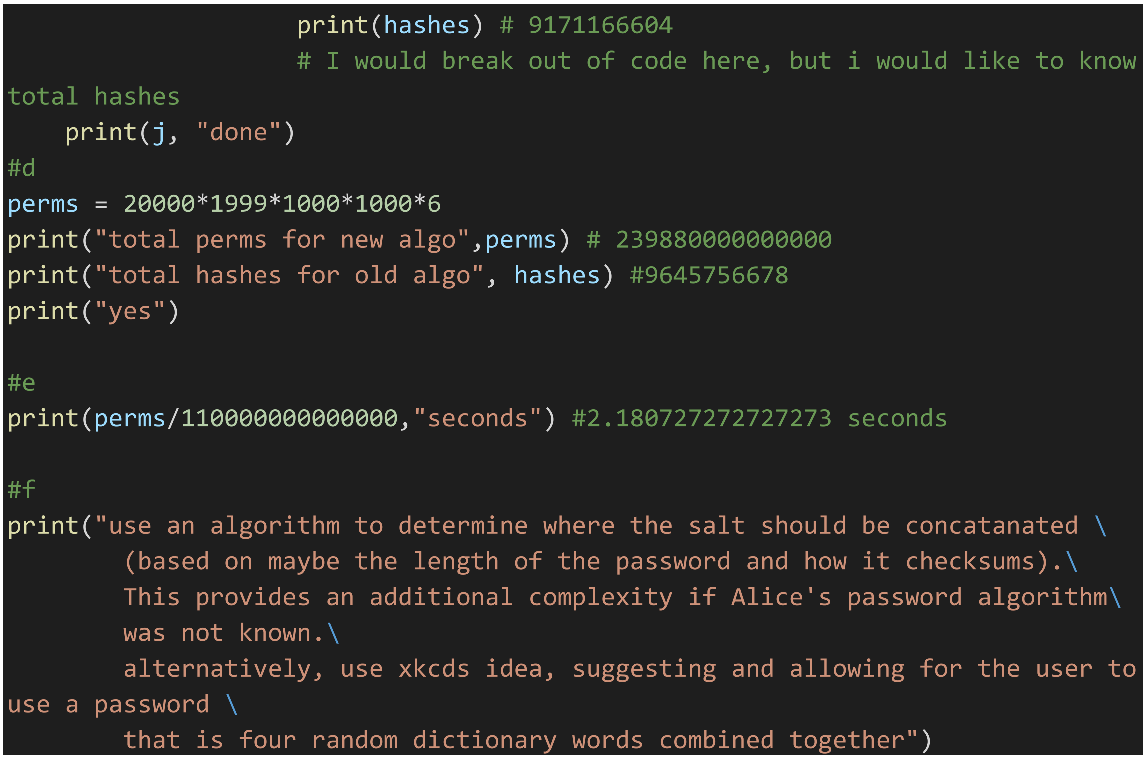


```python
#QN 2 - j2tham
# Discussed with Sean, Louis, Min Htet, Min Yue
import hashlib
#a
print("forming")

#b
hash = "7bc5c1e383607e95a206d20e1d1d3fd8eab147bcb7420a5eb0fc672f4ed92755"
for i in range(1000000):
    s = "16705816"+f'{i:06}'

    ans = hashlib.sha256(s.encode()).hexdigest()
    if ans == hash:
        print(i) #971204
        break

#c

# from the 3 compromised passwords, it is likely Alice usues passwords of:
# 12 characters in length
# begins with a word (probably of length 10 or less)
# First character uppercase, rest of word in lowercase
# Last character symbol
# remaining characters padded with numbers

hash = "cd01a419235b5e283b12b7da8bbf53d04c89231c169d6ca4594227cde0ffa85e"
hashes = 0
sym = "!?*$#&"
f = open("word_list.txt", "r")
raw = f.read()
wordls = raw.splitlines()
wordls.remove("directive")
wordls.remove("villages")
wordls.remove("witness")
f.close()

for j in sym:
    print("working on",j)
    for i in wordls:
        if len(i)<=10: # need 1 number and symbol
            digit = 11-len(i)
            for k in range((digit)**10):
                s = "97375774" + i.capitalize() + str(k).zfill(digit)+j
                ans = hashlib.sha256(s.encode()).hexdigest()
                hashes += 1
                if ans == hash:
                    print(i.capitalize() + str(k).zfill(digit)+j)
#Proposals92&
```



```
                    print(hashes) # 9171166604
                    # I would break out of code here, but i would like to know
total hashes
    print(j, "done")
#d
perms = 20000*1999*1000*1000*6
print("total perms for new algo",perms) # 239880000000000
print("total hashes for old algo", hashes) #9645756678
print("yes")

#e
print(perms/110000000000000,"seconds") #2.180727272727273 seconds

#f
print("use an algorithm to determine where the salt should be concatanated \
        (based on maybe the length of the password and how it checksums).\
        This provides an additional complexity if Alice's password algorithm\
        was not known.\
        alternatively, use xkcds idea, suggesting and allowing for the user to
use a password \
        that is four random dictionary words combined together")
```