import hashlib

import time

start\_time = time.time()

allowedCharacters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&\'()\*+,-./:;<=>?@[\\]^\_`{|}~'

totalCharacters = len(allowedCharacters)

def CompareHash(generatedPassword, providedHash):

if hashlib.md5(str(generatedPassword).strip()).hexdigest() == hash\_md5:

end\_time = time.time()

print("Crecked password is: " + generatedPassword)

print("Total time comsumed: %.4f sec" % (end\_time - start\_time))

exit()

def CharacterToIndex(char):

return allowedCharacters.index(char)

def IndexToCharacter(index):

if totalCharacters <= index:

raise ValueError("Index out of range.")

else:

return allowedCharacters[index]

def next(string):

if len(string) <= 0:

string.append(IndexToCharacter(0))

else:

string[0] = IndexToCharacter((CharacterToIndex(string[0]) + 1) % totalCharacters)

if CharacterToIndex(string[0]) is 0:

return list(string[0]) + next(string[1:])

return string

def GeneratePasswords(hash\_md5):

global start\_time

start\_time = time.time()

sequence = list()

while len(sequence) < 7:

generatedPassword = ""

sequence = next(sequence)

for character in sequence:

generatedPassword = generatedPassword + character

CompareHash(generatedPassword, hash\_md5)

if \_\_name\_\_ == "\_\_main\_\_":

hash\_md5 = str(input("Enter MD5 hash of password you want to crack: "))

GeneratePasswords(hash\_md5)