import string  
import random  
  
class BasePasswordManager():  
 def \_\_init\_\_(self, old\_passwords):  
 self.old\_passwords = old\_passwords  
 def get\_password(self):  
 return self.old\_passwords[-1]  
 def is\_correct(self, current\_password):  
 return current\_password == self.get\_password()  
  
class PasswordManager(BasePasswordManager):  
 def set\_password(length):  
 password\_characters = string.ascii\_letters + string.digits + string.punctuation  
 new\_password = ""  
 for i in range(length):  
 new\_password= new\_password + (random.choice(password\_characters))  
 print(new\_password)  
 return new\_password  
# this method takes the current password and returns the security level of the password  
 def get\_level(current\_password):  
 x = True  
 while x:  
 if current\_password.isalpha() or current\_password.isdigit():  
 print("current\_password is of level 0")  
 return 0  
 break  
 elif current\_password.isalnum():  
 print("current\_password is of level 1")  
 return 1  
 break  
 else:  
 print("current\_password is of level 2")  
 return 2  
 x = False  
 break  
#the idea here is to take the new generated password and runs it through this loop to find its security level  
 def get\_level2(new\_password):  
 x = True  
 while x:  
 if (len(new\_password)) < 6:  
 print('new\_passord not of accepted length')  
 print('generate another password of length > 6')  
 break  
 elif new\_password.isalpha() or new\_password.isdigit():  
 print('new\_password is of level 0')  
 return 0  
 break  
 elif new\_password.isalnum():  
 print('new\_password is of level 1')  
 return 1  
 break  
 else:  
 print('new\_password is of level 2')  
 return 2  
 x = False  
 break  
#bpm this is a list of past passwords with the current past word last on the list  
bpm = BasePasswordManager(['224466','zxywu','AbCdEquality', 'Jupyterishere','GameChanger123'])  
print(bpm.get\_password())  
print(bpm.is\_correct('GameChanger123'))  
current\_password\_level = PasswordManager.get\_level(input('Enter the Current\_password above'))  
new\_password = PasswordManager.set\_password(int(input('Enter the new password length')))  
new\_password\_level = PasswordManager.get\_level2(input('Enter the newpass generated above'))  
  
# this loop takes in the two security levels from get\_level and get\_level2 and compares which is of higher security  
current\_password = bpm.get\_password()  
  
while current\_password\_level >= new\_password\_level:  
 print('new\_password not accepted')  
 new\_password = (PasswordManager.set\_password(int(input('Enter the new password length'))))  
 new\_password\_level = PasswordManager.get\_level2(input('Enter the newpass generated above'))  
 if current\_password\_level == 2:  
 if len(new\_password) > len(current\_password):  
 print('Password is accepted')  
 print('new password is of greater strength')  
 print('You have successfully set a new password')  
 break  
  
else:  
 print('new password is of greater strength')  
 print('You have successfully set a new password')