import re

class BasePasswordManager:

old\_passwords = []

def get\_password(self):

if(len(self.old\_passwords) > 0):

self.current\_password = self.old\_passwords[-1]

return self.current\_password

def is\_correct(self,pwd):

self.password = pwd

if(self.password == self.get\_password()):

return True

else:

return False

class PasswordManager(BasePasswordManager):

def set\_password(self,new\_password):

self.new\_password = new\_password

self.level\_new = self.get\_level(self.new\_password)

if(len(self.new\_password) < 6):

print("Password too short! Try resetting again with a minimum 6 characters")

else:

if(self.is\_correct(self.new\_password)):

print("New Password can not be same as the old password")

elif((len(self.old\_passwords)>0)and(len(self.new\_password) > 6)):

self.current\_password = self.get\_password()

self.level\_current = self.get\_level(self.current\_password)

if((self.level\_new > self.level\_current) or (self.level\_new == 2 and self.level\_current ==2)):

self.old\_passwords.append(self.new\_password)

print("Password Changed!")

else:

print("Use a stronger password")

elif(len(self.old\_passwords)==0):

self.old\_passwords.append(self.new\_password)

print("Password Changed!")

def get\_level(self,pwd):

self.pwd = pwd

if(self.pwd.isalpha() or self.pwd.isdigit()):

return 0

elif(self.pwd.isalnum()):

return 1

elif(bool(re.findall('[^A-Za-z0-9]',self.pwd))) and (bool(re.findall('[!@#$%^&\*()\_<>?/\|}{~:]',self.pwd))):

return 2

else:

pass

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

while True:

choice = input("Press 1 to Change password \nPress 2 to exit.\n ")

if(choice == "1"):

password = input("Enter password: ")

obj = PasswordManager()

obj.set\_password(password)

elif(choice == "2"):

print("Thank you!!")

break

else:

print("Try Again!")

pass