class TestPasswordValidation(unittest.TestCase):

def test\_password\_too\_short(self):

result = validate\_password("abc123")

self.assertEqual(result, "Password must be at least 8 characters.")

def test\_password\_no\_number(self):

result = validate\_password("abcdefgh")

self.assertEqual(result, "Password must include at least one number.")

def test\_password\_short\_with\_number(self):

result = validate\_password("abc1234")

self.assertEqual(result, "Password must be at least 8 characters.")

def test\_valid\_password(self):

result = validate\_password("abc12345")

self.assertEqual(result, True)

def test\_long\_password\_with\_numbers(self):

result = validate\_password("mypassword1")

self.assertEqual(result, True)

def test\_all\_numbers\_valid(self):

result = validate\_password("12345678")

self.assertEqual(result, True)

def validate\_password(password):

if len(password) < 8:

return "Password must be at least 8 characters."

if not any(char.isdigit() for char in password):

return "Password must include at least one number."

return True

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

```

import re

```python

if len(password) < 8:

return False, "Password must be at least 8 characters long"

if not re.search(r"\d", password):

return False, "Password must contain at least one numeric digit"

if not re.search(r"[!@#$%^&\*(),.?\":{}|<>`]", password):

return False, "Password must contain at least one special character"

return True, "Password is valid"

def test\_passwords():

print(valid, msg)

print(valid, msg)

print(valid, msg)

print(valid, msg)

print(valid, msg)

print(valid, msg)

print(valid, msg)

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