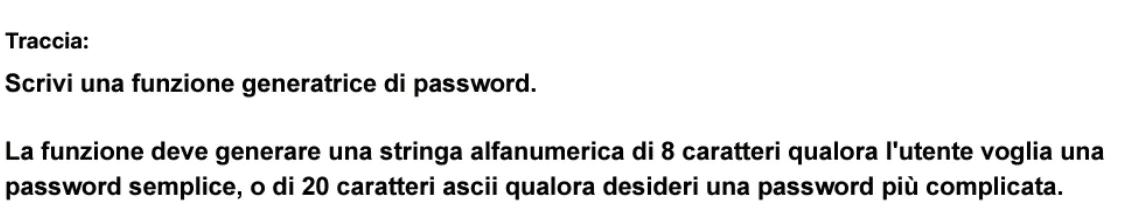
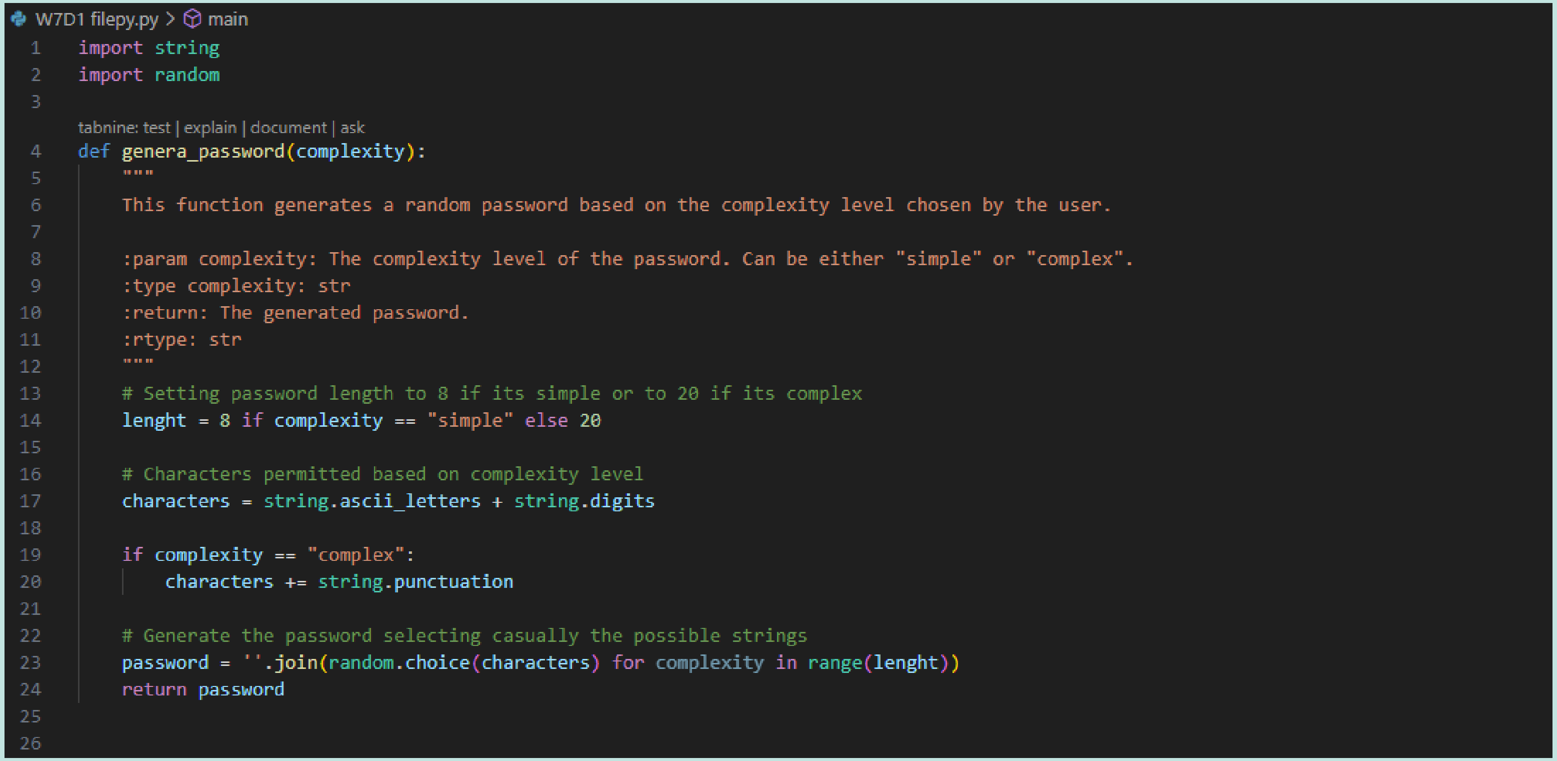
W7D3 - ESERCIZIO



Traccia:

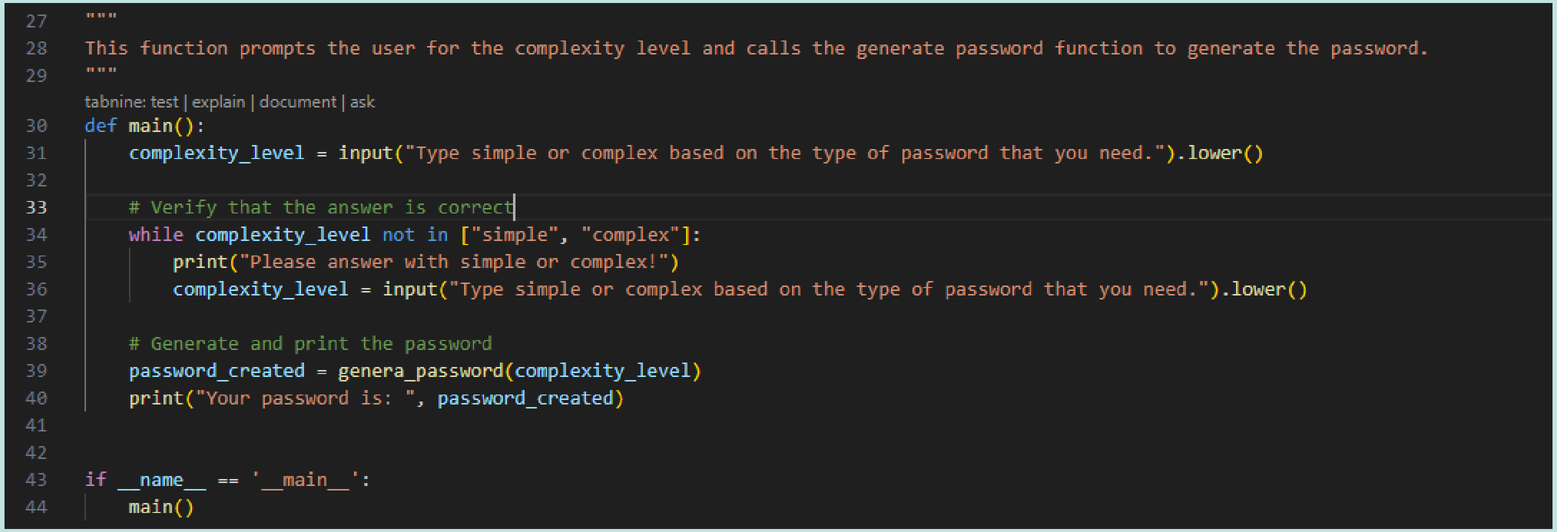
Scrivi una funzione generatrice di password.

La funzione deve generare una stringa alfanumerica di 8 caratteri qualora l'utente voglia una password semplice, o di 20 caratteri ascii qualora desideri una password più complicata.



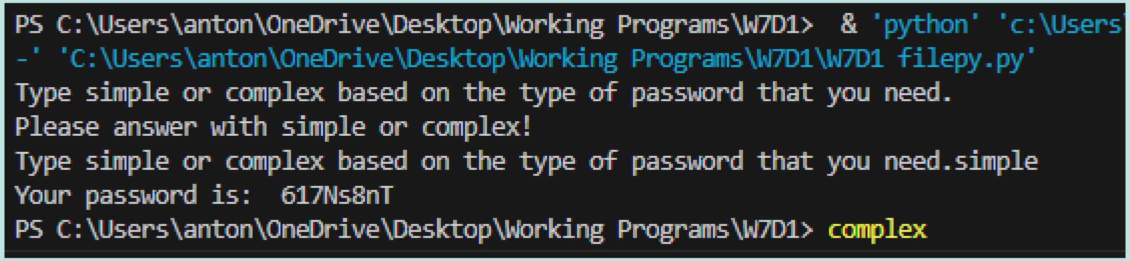
```python
import string
import random

def genera_password(complexity):
    """
    This function generates a random password based on the complexity level chosen by the user.

    :param complexity: The complexity level of the password. Can be either "simple" or "complex".
    :type complexity: str
    :return: The generated password.
    :rtype: str
    """
    # Setting password length to 8 if its simple or to 20 if its complex
    lenght = 8 if complexity == "simple" else 20

    # Characters permitted based on complexity level
    characters = string.ascii_letters + string.digits

    if complexity == "complex":
        characters += string.punctuation

    # Generate the password selecting casually the possible strings
    password = ''.join(random.choice(characters) for complexity in range(lenght))
    return password


```

- Iniziamo con la scrittura del codice spiegando anche cosa fa ogni linea del codice e cercando di spiegare ogni parametro



```python
"""
This function prompts the user for the complexity level and calls the generate password function to generate the password.
"""
def main():
    complexity_level = input("Type simple or complex based on the type of password that you need.").lower()

    # Verify that the answer is correct
    while complexity_level not in ["simple", "complex"]:
        print("Please answer with simple or complex!")
        complexity_level = input("Type simple or complex based on the type of password that you need.").lower()

    # Generate and print the password
    password_created = genera_password(complexity_level)
    print("Your password is: ", password_created)


if __name__ == '__main__':
    main()
```

- Una volta finito di scrivere il code , procediamo con la funzione di Run e Debug del code e otteniamo il seguente risultato:



```
PS C:\Users\anton\OneDrive\Desktop\Working Programs\W7D1>  & 'python' 'c:\Users
-' 'C:\Users\anton\OneDrive\Desktop\Working Programs\W7D1\W7D1 filepy.py'
Type simple or complex based on the type of password that you need.
Please answer with simple or complex!
Type simple or complex based on the type of password that you need.simple
Your password is:  617Ns8nT
PS C:\Users\anton\OneDrive\Desktop\Working Programs\W7D1> complex
```

- Notiamo che avendo dato una risposta diversa da complex o simple il code ci chiede di nuovo di scrivere simple o complex dal momento che la risposta era invalida.