

Progetto:

Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica. Dato il codice si richiede allo studente di:

1. Capire cosa fa il programma senza eseguirlo. X
2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
3. Individuare eventuali errori di sintassi / logici.
4. Proporre una soluzione per ognuno di essi.

Codice

```
import datetime

def assistente_virtuale(comando):

    if comando == "Qual è la data di oggi?":

        oggi = datetime.date.today()

        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")

    elif comando == "Che ore sono?":

        ora_attuale = datetime.datetime.now().time()

        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")

    elif comando == "Come ti chiami?":

        risposta = "Mi chiamo Assistente Virtuale"

    else:

        risposta = "Non ho capito la tua domanda."

    return risposta

while True

    comando_utente = input("Cosa vuoi sapere? ")

    if comando_utente.lower() == "esci":

        print("Arrivederci!")

        break

    else:

        print(assistente_virtuale(comando_utente))
```

1. Questo programma crea una sorta di conversazione tra l'utente e un assistente virtuale.

C'è l'**import datetime**, il quale è una libreria di python che fornisce classi per gestire date e orari.

Avendo un ciclo while che chiede un input all'utente, in base all'input si possono innescare le varie condizioni degli if-elif.

Esempio: avviando il codice all'inizio leggo la stringa "Cosa vuoi sapere?" e inserendo ad esempio "Qual è la data di oggi?" dovrei ottenere "La data di oggi è 11/07/25"

2. Una casistica non standard che il programma non gestisce è quella nella quale l'utente inserisca degli input non rispettando le lettere maiuscole entrando così in un loop che mostra sempre e solo la condizione dell'else. Un altro tipo di loop si può creare dal momento in cui non vengono mostrate le varie opzioni che l'utente può selezionare e anche come uscire dall'applicazione, entrando appunto in un loop che mostrerà sempre la condizione else.

3. Errori di sintassi li troviamo alla prima condizione if:

oggi = datetime.datetoday() → la soluzione è **datetime.date.today()**

Il secondo errore di sintassi nel ciclo while:

while True → **while True:**

4. La soluzione ai problemi potrebbe essere aggiungere il **.lower()** nelle condizioni dei vari if in modo che non ci siano problemi con l'input dell'utente, trasformando tutto quello che scrive in minuscolo e aggiungendo magari nell'input del while le varie opzioni che l'assistente virtuale supporta e l'opzione per uscire dall'applicazione. Per esempio implementando dei print prima della funzione in modo che appaiano a schermo.