

Traccia Esercizio:

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.
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.

```
import datetime
```

```
def assistente_virtuale(comando):
```

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

```
        oggi = datetime.datetime.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))
```

Esecuzione:

1. Capire cosa fa il programma senza eseguirlo:

Dal codice si evince che il programma vuole simulare un assistente virtuale che ritorna al utente informazioni riguardanti data, ora e simili sfruttando il modulo DATETIME e richieste tramite input

2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).

Il programma comincia chiedendo come input al utente per cosa vuole sapere. Mi piacerebbe che al avvio il programma dia qualche informazione generale per aiutare l'utente a capire che programma ha avviato o il suo scopo, anche con una semplice descrizione o qualche frase che dia una contesto.

Il programma non dà al utente la possibilità di vedere chiaramente i comandi disponibili per effettuare una scelta, lasciando l'utente senza mezzi per usufruire del programma in ogni sua parte dove richiede input.

Nonostante il programma non dia i comandi disponibili al utente e prevista un opzione di uscita "esci" ma è accettata solo se scritta dal utente in minuscolo, portandolo di nuovo l'utente in confusione anche se prova casualmente qualche comando.

I metodi utilizzati riportano data e ora, ma senza prevedere posizione geografica o altri dettagli è impossibile avere l'ora corretta a causa di differenze di standard internazionali tra stati e altri fattori

Trovo che sarebbe utile utilizzare il metodo .lower per elaborare gli input in minuscolo per poi confrontarli con gli operatori logici nei vari casi IF ELIF ELSE in modo da dare al utente maggior flessibilità sugli input digitati per avere una miglior fruizione del programma.

3. Individuare eventuali errori di sintassi / logici.

```
oggi = datetime.datetoday() # Metodo errato, dalla documentazione python today = date.today(  
ora_attuale = datetime.datetime.now().time() # Metodo errato  
risposta = "Non ho capito la tua domanda." # Variabile non definita  
while True # Errore di sintassi, mancano i : dopo il true del while  
print(assistente_virtuale(comando_utente)) # Metodo errato
```