



Traccia:

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.date.today()      --->      #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                                --->      #while True:

    comando_utente = input("Cosa vuoi sapere? ")

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

        print("Arrivederci!")

        break

    else:

        print(assistente_virtuale(comando_utente))
```

Errori di sintassi/logici trovati: 2

Il programma che stiamo analizzando utilizza il metodo “**datetime**” che è pensato per lavorare con date e orari.

La funzione che svolge questo programma è abbastanza semplice; In base alla domanda che noi poniamo ad esso, lui ci darà una risposta che è correlata alla domanda stessa, appoggiandosi al metodo **datetime** per ottenere le informazioni variabili che ottiene, appunto dalle **date** e dall’**orario**.

Nel nostro caso:

-Qual è la data di oggi? ---> La data di oggi è “giorno/mese/anno”

-Che ore sono? ---> L’ora attuale è “ore/minuti”

-Come ti chiami? --->Mi chiamo “Assistente Virtuale”

Tutto questo è preceduto da una domanda che è inserita all’interno di un ciclo while, la domanda assegnata nel ciclo while è “Cosa vuoi sapere?”. Questo sta a significare che il programma ci chiederà sempre questa domanda, prima di passare alle 3 domande all’interno del ciclo ifelse.

Il programma contiene anche un comando per uscire e una risposta nel caso si verificasse una richiesta dell’utente non contemplata.

Nonostante il codice ha una logica molto semplice sin da subito possiamo notare casistiche non standard che potenzialmente potrebbero creare problemi o in alcuni casi potrebbe non gestire correttamente le nostre richieste.

Il primo problema e credo quello più evidente, è che nel caso in cui l’utente inserisca una domanda diversa da quelle impostate, il programma reagisce solo con “Non ho capito la tua domanda”.

Stessa cosa vale per i formati di scrittura, ad esempio se l’utente fa una delle domande contemplate all’interno del programma, ma lo fa in minuscolo, il programma non riconoscerà la domanda. Un altro esempio potrebbe essere la lingua; per lo stesso principio di maiuscolo/minuscolo, anche nel caso dell’utilizzo, da parte dell’utente, di una lingua diversa dall’italiano, il programma vedrà la richiesta come un errore e risponderà con “Non ho capito la tua domanda”.

Un’altra cosa che il programma non gestisce è il fuso-orario, o il formato dell’ora (AM/PM).

Se l’utente si aspetta una risposta che includa anche i secondi, questo non sarà possibile perché i secondi non sono stati inseriti all’interno del formato.

Sappiamo anche che il programma dà la possibilità all’utente di fermare il ciclo tramite il comando “esci” ma questa funzione non viene specificata e mostrata all’utente in nessuna situazione possibile.

Vista la semplicità del programma in sé, propongo di lavorare sui punti precedentemente elencati e rendere il programma più chiaro, magari con una lista a schermo delle domande a cui può rispondere comprendendo anche la funzione “Esci” che altrimenti, resta nascosta agli occhi dell’utente.

Aggiungere dell’elasticità permettendo più formati di scrittura (MAISC or minsc) e più lingue, aggiungendo la libreria **langdetect** che rileva automaticamente la lingua di testo.

Stesso vale per il fuso-orario, aggiungendo la libreria **pytz**, si permette all’utente di specificare il fuso desiderato. Questa specifica funzione potrebbe essere implementata in modo da far scegliere la zona geografica prima di avviare i cicli, migliorando così l’esperienza dell’utente.

Conclusioni:

Il programma per quanto ideologicamente semplice, ha potenziale e può essere migliorato in svariati modi. In sintesi, l'implementazione di queste modifiche permetterebbe di migliorare l'interazione con l'utente, rendendo l'assistente virtuale più versatile, intuitivo e globalmente accessibile.