

Test 06/12/24

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:

- Capire cosa fa il programma senza eseguirlo. Esercizio Progetto
- Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
- Individuare eventuali errori di sintassi / logici.
- 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()

        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))
```

Questo programma è basato su una funzione che dice di chiamarsi "Assistente Virtuale"(riga 10). Lo scopo del programma si basa appunto sul dare un" input" per avere informazioni da parte dell' utente, e ricevere un' output dall' assistente virtuale come risposta. Notiamo **3 "input"** con output. Se l' utente chiede **qual' è la data di oggi?**, l' assistente risponderà **"la data di oggi è.."**, se chiede **"Che ore sono?"** come risposta si riceverà **"l' ora attuale è.."**, se si chiede **"come ti chiami?"** dice **"Mi chiamo assistente virtuale"**, invece se si digita **"esci"**, l'assistente virtuale dirà **"Arrivederci"**.

Sotto rivediamo il programma migliorato con i commenti che spiegano ciascuna riga:

Importa il modulo datetime, che serve per lavorare con date e orari

```
import datetime
```

Definisce una funzione chiamata assistente_virtuale che prende un input

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

 # Controlla se il comando è "Qual è la data di oggi?"

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

 # Prende la data di oggi

```
        oggi = datetime.datetime.today() # Corretto: usa datetime.datetime.today per ottenere la data
```

 # Crea una risposta con la data di oggi nel formato giorno/mese/anno

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

 # Controlla se il comando è "Che ore sono?"

```
    elif comando == "Che ore sono?":
```

 # Prende l'ora attuale

```
        ora_attuale = datetime.datetime.now().time() # Ottiene solo l'ora
```

 # Crea una risposta con l'ora attuale nel formato ore:minuti

```
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M") # Usa ":" per separare ore e minuti
```

 # Controlla se il comando è "Come ti chiami?"

```
    elif comando == "Come ti chiami?":
```

 # Crea una risposta con il nome dell'assistente virtuale

```

    risposta = "Mi chiamo Assistente Virtuale"

    # Se il comando non è nessuno di quelli sopra

else:

    # Crea una risposta per dire che non ha capito il comando

    risposta = "Non ho capito la tua domanda."

    # Restituisce la risposta creata

return risposta

# Continua a chiedere all'utente un comando

while True:

    # Chiede all'utente di scrivere un comando

    comando_utente = input("Cosa vuoi sapere? ")

    # Controlla se l'utente ha scritto "esci"

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

        # Scrive un saluto per uscire

        print("Arrivederci!")

        # Esce e termina il programma

        break # Aggiunto per uscire

    # Se il comando non è "esci", chiama la funzione assistente_virtuale con il comando
    dell'utente

else:

    risposta = "assistente_virtuale" (comando_utente) # Aggiunto per chiamare la funzione

    # Stampa la risposta che ha dato la funzione

    print(risposta) # Aggiunto per la risposta

```

Adesso vediamo e le casistiche non standard che il programma non gestisce e errori di sintassi e logica:

. Ci sono errori per quanto riguarda l' **orario e la data**, se si scrivesse un comando in un modo diverso (per esempio, "Qual è la data di oggi?" con lettere **maiuscole** o **con punteggiatura**), il

programma non lo riconoscerebbe. Ciò accade perchè in programmazione c'è una netta differenza tra i due e si usano comandi diversi per distinguerle per esempio **lower()** e **upper()**. Inoltre non è specificato in base a quale parte del mondo (paese) si sta chiedendo la data e l'ora, ciò potrebbe confondere il programma.

. Inoltre se l'utente invia un **input vuoto** o composto **solo da spazi**, il programma non sa come gestire questa situazione.

. Se ci sono problemi con la **libreria "datetime"**, come un errore durante l'importazione, il programma si fermerà senza gestire l'errore.

. La riga `oggi = datetime.datetoday()` dovrebbe essere `oggi = datetime.datetime.today()`. `datetoday()` non è un metodo valido.

. **while True** non chiude correttamente il codice. Dovrebbe essere scritto così:
"assistente_virtuale".