

Progetto

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.
- Individuare dal 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.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))
```

- 1) Il programma è un assistente virtuale programmato per restituire data odierna, ora attuale ed il suo nome (Assistente Virtuale). Nella prima parte del codice è definita la funzione principale del programma chiamata `assistente_virtuale`, che con una serie di condizioni ritorna la risposta richiesta, oppure nel caso in cui l'input non sia valido, ritorna "Non ho capito la tua domanda".
Il ciclo `while` invece lancia il programma, finché l'utente non scriverà "esci" (non case-sensitive, visto che l'input viene convertito in minuscolo) il ciclo continuerà a richiamare la funzione `assistente_virtuale` e attendere gli input del utente.
- 2) Le casistiche non standard che il programma non gestisce sono:
 - Se l'input del utente non è formattato come da richiesta, il programma non lo riconoscerà;
 - L'input deve essere preciso e non viene elencato.
- 3) Gli errori di sintassi sono:
 - Dopo `while True` il compilatore si aspetterà " : " ;
 - Nella condizione della data odierna, la formattazione corretta è `datetime.date.today()` .

- 4) Gli errori di sintassi sono stati corretti al punto 3, per quanto riguarda la gestione dei casi non considerati, io procederei come segue:
- Normalizzerei l'input iniziale con `.lower()` all'inizio del metodo `assistente_virtuale` e cambierei tutte le stringhe di paragone con delle stringhe minuscole, per evitare ogni tipo di differenza tra ciò che l'utente andrà a inserire e ciò che il programma si aspetta.
 - Se l'utente entra nel `else` del metodo `assistente_virtuale`, aggiungerei una lista dei comandi validi alla risposta.

Il codice con le modifiche sarebbe così:

```
import datetime
def assistente_virtuale(comando):
    comando=comando.lower()
    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 = f"Non ho capito la tua domanda. Questo è l'elenco delle domande valide:\nQual è la data di oggi?\nChe ore sono?\nCome ti chiami?\n"
    return risposta
while True:
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente.lower() == "esci":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))
```