

PROGETTO S2 L5

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.

PROGRAMMA:

```
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. COSA FA IL PROGRAMMA?

Questo programma è un assistente virtuale. Utilizza la libreria **datetime** per rispondere a domande su data e ora, dove l'utente inserisce un comando (una domanda) ed il programma controlla se il comando corrisponde ad una frase specifica : ad ogni richiesta valida, l'applicazione fornisce una risposta pertinente.

- **“ Qual è la data di oggi?” >>> risponde con la data**
- **“ Che ore sono?” >>> risponde con l’ora**
- **“Come ti chiami?” >>> risponde con il nome**
- **Se invece non capisce >>> dirà “ Non ho capito la tua domanda.”**
- **Continua a chiedere input finchè l’utente non scrive “esci”**

Le risposte alle tre domande principali di cui sopra sono gestite all’interno di una struttura condizionale con **“if”, “elif”, “else”**.

L’interazione è gestita tramite un ciclo continuo (**“while”**) che sollecita costantemente l’utente con la domanda **“Cosa vuoi sapere?”**.

È presente anche un meccanismo per terminare l’esecuzione e una risposta generica per le richieste non riconosciute:

Risposta generica **>>> risposta = “Non ho capito la tua domanda.”**

Meccanismo per terminare l’esecuzione **>>> if comando_utente.lower() == “esci”:**

print(“Arrivederci!”)

break

2. CASISTICHE NON STANDARD CHE IL PROGRAMMA NON GESTISCE

- Il programma riconosce solo le domande scritte **esattamente** come previsto. Se scrivi la domanda con lettere maiuscole per esempio non le capisce.

- Non dice all'utente come uscire dal programma.
- Si presenta poco "friendly"
- Se scrivi qualcosa che non è una domanda prevista, ti risponde sempre nello stesso modo generico " Non ho capito la tua domanda"
- Ho notato che il programma non tiene conto dei fusi orari, infatti non chiede di scegliere un paese o città per poi indicare l'ora in base al fuso orario.

3. ERRORI NEL PROGRAMMA

PRIMO ERRORE : C'è un errore nel metodo per prendere la data : viene usato `datetime.datetoday()` che non esiste.

Il metodo corretto è `>>> datetime.date.today()`

SECONDO ERRORE : Manca il simbolo ":" dopo `while True` per completare la sintassi correttamente

4. SOLUZIONE PROPOSTA

Come soluzione per apportare delle migliorie al programma ho ragionato in questi termini :

1. Darei un'impostazione meno rigida per quanto riguarda l'utilizzo delle maiuscole e degli spazi per ridurre le possibilità di errore

ESEMPIO: `nome_variabile.lower().strip()`

2. Implementare la possibilità di scegliere la città per poter ottenere l'orario corretto basato sul fuso orario della città scelta.

ESEMPIO: importare libreria "pytz"

3. Implementerei un comando print per far sapere all'utente che può uscire dal programma digitando "esci ora"

ESEMPIO : `print("Quando hai finito, puoi digitare 'esci ora' per terminare")`

4. Implementerei dei comandi "print" iniziali per dare all'utente più informazioni sulle possibilità di scelta che ha.

**ESEMPIO : `print(" Ciao! Sono il tuo Assistente Virtuale, mi chiamo Tassisto")`
`print(" Queste sono le domande che puoi farmi : ")`
`print("Qual'è la data di oggi?")`
`print("Che ore sono?")`
`print("Come ti chiami?")`**