

COMPITO

Cosa fa il programma

```
1 import datetime
2
3 while True:
4     comando_utente = input("Cosa vuoi sapere? ")
5     if comando_utente == "esci":
6         print("Arrivederci!")
7         break
8     else:
9         print(assistente_virtuale(comando_utente))
10
11 def assistente_virtuale(comando):
12     if comando == "Qual è la data di oggi?":
13         oggi = datetime.datetime.now()
14         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
15     elif comando == "Che ore sono?":
16         ora_attuale = datetime.datetime.now().time()
17         risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
18     elif comando == "Come ti chiami?":
19         risposta = "Mi chiamo Assistente Virtuale"
20     else:
21         risposta = "Non ho capito la tua domanda."
22
23 return risposta
```

Il programma implementa un assistente virtuale che:

- Entra in un loop infinito chiedendo comandi all'utente
- Permette di uscire digitando "esci"
- Risponde a tre comandi specifici:
 - Qual è la data di oggi? restituisce la data
 - Che ore sono? restituisce l'ora attuale
 - Come ti chiami? risponde "Mi chiamo Assistente Virtuale"
- Per qualsiasi altro input restituisce "Non ho capito la tua domanda."

Errori:

Case sensitivity:

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

Il confronto è case-sensitive. Varianti come:

"qual è la data di oggi?" (minuscolo)

"QUAL È LA DATA DI OGGI?" (maiuscolo)

"Qual è la data di oggi?" (apostrofo diverso)

Non vengono riconosciute.

Spazi in più:

Input inseriti dall'utente con spazi prima e dopo come "Qual è la data di oggi? " non vengono gestiti.

Variazioni Linguistiche:

Sinonimi o varianti non sono gestiti come

"Che giorno è oggi?"

"Dimmi la data"

"Qual è l'ora?"

"Che ora è?"

Accenti e Caratteri Speciali

Il carattere "è" può essere codificato diversamente su sistemi diversi, causando problemi di matching.

Manca la gestione del formato input errato

Se inserisco un numero o inserisco un input vuoto “ ”

risponde sempre "non ho capito", ma potrebbe gestirlo meglio facendo inviare reinserire all'utente la scelta

Errori di Sintassi/Logici

```
if comando_utente == "esci":
```

```
    print("Arrivederci")
```

Break

Break in riga 7 dovrebbe essere allineato con il `print("Arrivederci!")` sopra.

La funzione `assistente_virtuale` usa `return` fuori dal blocco globale, il `return` è indentato correttamente, ma è meglio riorganizzare la logica per essere più chiari.

```
else: in riga 8 print(assistente_virtuale(comando_utente))
```

Questo `else` si riferisce all'`if comando_utente == "esci"`, ma sembra illogico perché viene eseguito per qualsiasi comando diverso da "esci".

La funzione `assistente_virtuale` usa "elif" per tutti i comandi, ma l'ultimo `else` viene eseguito per qualsiasi input non riconosciuto, rendendo ridondanti alcuni controlli.

messaggio con spazi

```
"L'ora attuale è " + ora_attuale.strftime("%H:%M")
```

Nella stampa del `print` risulterà un doppio spazio. Nessun bug grave, ma sarebbe meglio da evitare.

Miglioramenti pensati

- `strip()` elimina spazi extra
- `lower()` evita problemi con maiuscole/minuscole
- Indentazione corretta

- Input gestito in forma normalizzata
- Input vuoto va gestito esplicitamente con messaggio appropriato
- Varianti comandi che supporta molteplici formulazioni della stessa domanda
- “Non ho capito...” potrebbe proporre suggerimenti
- Se un utente inserisce input malformato (es. 1000 caratteri), non esiste protezione