

## Esercizio di Progetto

Codice:

```
import datetime
```

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

```
    if comando == "Qual è la data di oggi?": → (*) Miglioramento codice "Qual è la data di oggi?" .strip().lower():
```

```
        oggi = datetime.date.today() → → Errore nel codice - correzione > datetime.date.today()
```

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

```
    elif comando == "Che ore sono?": → (*) Miglioramento codice "Che ore sono?" .strip().lower():
```

```
        ora_attuale = datetime.datetime.now().time() → Qui si potrebbe anche evitare di ripetere il .time()
```

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

```
    elif comando == "Come ti chiami?": → (*) Miglioramento codice "Come ti chiami?" .strip().lower():
```

```
        risposta = "Mi chiamo Assistente Virtuale"
```

```
    else:
```

```
        risposta = "Non ho capito la tua domanda."
```

```
    return risposta
```

```
while True → → Errore di sintassi mancano i ( : ) correzione = while True: → miglioriamo questa parte (**)
```

```
comando_utente = input("Cosa vuoi sapere? ")
```

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

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

```
        break
```

```
    else:
```

```
        print(assistente_virtuale(comando_utente))
```

## 1 - Capire cosa fa il programma senza eseguirlo:

Ovviamente da come è scritto nel comando, si parla di un programma di Assistente vocale che tramite le domande impostate, comunica il giorno e l'ora all'utente.

## 2 – Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).

La prima cosa che mi viene da pensare in questo caso è il fatto che python sia key-sensitive.

Sell'utente scrivesse "che Ore sOno?" per esempio, il programma non riconoscerebbe la domanda, e porterebbe all'else "Non ho capito la tua domanda". Lato user experience non è il massimo, visto che il significato della domanda è lo stesso. In questo caso uno dei miglioramenti che possiamo fare è aggiungere un `.strip().lower()` alla fine delle domande, come evidenziato nell'asterisco (\*).

## 3 – Individuare eventuali errori di sintassi/logici.

Tutti gli errori che sono riuscita a trovare, li ho segnalati con gli asterischi – e in rosso quelli importanti.

Il primo errore chiave è quello `datetime.date.today()` che essendo errato il comando non parte a priori.

Altro errore importante è nel " while True " in cui mancano i due punti, questo è più un errore di sintassi.

Tutte le altre cose le ho aggiunte affianco al codice.

## 4 – Proporre una soluzione per ognuno di essi.

Tutte le quick solution sono nella parte destra del codice.

(\*\*) Miglioramento parte While True qui:

```
comando_utente = ""

while comando_utente.lower() != "esci"

    comando_utente = input("Cosa vuoi sapere? ")

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

        print("Arrivederci!")

    else:

        print(assistente_virtuale(comando_utente.strip())) (***)
```

(\*\*) Abbiamo rimosso il while True, perché in genere non è sicuro in quanto può generare loop infiniti soprattutto se non è specificato il break all'interno del while correttamente.

(\*\*\*) Abbiamo, inoltre, aggiunto lo `.strip()` per controllare gli spazi (se l'utente dovesse aggiungere uno spazio in più nella domanda, pulisce la stringa e lo formatta ad uno spazio, per evitare che vada nell'else "Non ho capito la domanda").

Sicuramente ci sarebbero più cose da aggiungere, la prima in assoluto, è l'introduzione al programma.

Cioè l'utente arriva e gli compare subito "Cosa vuoi sapere?" Senza però spiegargli che cosa sia il programma e cosa faccia. Magari aggiungere un comando Help – con elencate le funzioni che il programma può eseguire.

Nel nostro caso "Che ore sono?" – "Come ti chiami?" – "Qual è la data di oggi?"

Aggiungerei un:

```
print("***Benvenuto nella tua Assistente Virtuale\n***)

comando_utente= input("Scrivi Help per visualizzare i comandi disponibili")
```

Anais Fabriani

11/04/2025 - 13:53 – Settimo Torinese (TO)