

DATO QUESTO CODICE SCRITTO IN PYTHON STABILIRE COSA FA E CORREGGERE EVENTUALI ERRORI (SINTASSI, LOGICI, CASISTICHE NON CONTEMPLATE

```
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))
```

Dopo aver letto questo programma ho notato che si tratta di un assistente virtuale a cui si possono fare 3 domande: Data, Orario e Nome dell'assistente virtuale stesso. A livello di scrittura/sintassi/logica ho visto diversi errori che ho corretto. (Codice rifatto inserito in fondo al report). Il primo grosso errore è che così scritto non ci può essere interazione tra l'utente e l'assistente virtuale. In questo caso la variabile COMANDO non è assegnata a nessun tipo di input, quindi così non ci può essere avvio della funzione. Poi a livello personale sarebbe preferibile prima di assegnare la variabile comando fare dei print (stampe) con presentazione dell'assistente virtuale e gli input che saranno disponibili. Qui sotto le prime righe che ho modificato: (ho anche tolto la variabile dalla funzione principale nelle parentesi tonde (), mi sembra più corretto in questo modo)

```
import datetime
def assistente_virtuale():
    print("Ciao sono il tuo assistente virtuale, puoi farmi diverse domande:")
    print("1 Data di oggi")
    print("2 Orario")
    print("3 Il mio nome")
    comando_virtuale = input("Scegli la domanda 1, 2 o 3: ")
```

Più sotto come errore (aiutandomi con libreria Python) ho trovato che nella riga 4 "oggi = datetime.datetime.today()" manca il punto tra date e today, quindi la scrittura corretta sarà "oggi = datetime.date.today()". Poi a seguito dei vari if ed elif mancano i print leggibili dall'utente alle varie risposte, anche in questo caso non ci sarebbe interazione tra assistente virtuale e user (vedi sotto un esempio di riga scritta in modo corretto):

```
if comando_virtuale == "1":
    oggi = datetime.date.today()
    risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    print(risposta)
elif comando_virtuale == "2":
    ora_attuale = datetime.datetime.now().time()
    risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    print(risposta)
```

Ho anche tolto il "return risposta" (alla riga 13) considerandola non utile avendo in seguito un ciclo while che può far ripartire tutta la funzione da capo finché non viene dato il comando "esci".

Nella parte finale del programma con il while ho riscontrato come errori che dopo while True mancano i 2 punti necessari all'avvio del ciclo, ho modificato la domanda fatta all'utente con i 2 punti e lo spazio alla fine per avere un'interfaccia più chiara per chi usufruisce della funzione e ho riscritto parte del programma come segue:

```
while True:
    comando_utente = input("Vuoi farmi un'altra domanda? (Si o No): ")
    if comando_utente == "No":
        print("Arrivederci!")
        break
    elif comando_utente == "Si":
        input(assistente_virtuale())
    else:
        print("Comando non riconosciuto")
        break
```

Rispetto al programma fornito in origine ho aggiunto un elif che mi permette rispondendo "Si" di far ripartire la funzione da capo, e ho aggiunto in caso di una risposta diversa da "Si" e "No" il fatto che il ciclo si interrompa dando come output la scritta "Comando non riconosciuto". Dopo assistente_virtuale ho utilizzato le parentesi tonde vuote togliendo all'interno comando_utente (riga 20) per il richiamo alla funzione principale def assistente_virtuale() altrimenti avrei richiamato solo la domanda contenuta nel while. In fondo ho anche aggiunto la dicitura assistente_virtuale() in modo da far partire correttamente la funzione (altrimenti non partirebbe). (vedi sotto)

```
        else:
            print("Comando non riconosciuto")
            break

assistente_virtuale()
```

Come considerazione finale aggiungo (anche se non ho fatto questa modifica nel programma finale) che in alcuni punti del programma dove c'è il + tra diverse stringhe (sequenza di caratteri) per avere delle

espressioni più compatte si sarebbe potuta utilizzare la lettera f con le parentesi graffe { } (la f serve a formattare diverse stringhe).

Qui sotto il programma che ho riscritto partendo come base da quello errato della traccia. Ho modificato anche le variabili per rendere il programma più leggibile a chi si approccia a questo linguaggio di programmazione:

```
import datetime
def assistente_virtuale():
    print("Ciao sono il tuo assistente virtuale, puoi farmi diverse domande:")
    print("1 Data di oggi")
    print("2 Orario")
    print("3 Il mio nome")
    comando_virtuale = input("Scegli la domanda 1, 2 o 3: ")
    if comando_virtuale == "1":
        oggi = datetime.date.today()
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
        print(risposta)
    elif comando_virtuale == "2":
        ora_attuale = datetime.datetime.now().time()
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
        print(risposta)
    elif comando_virtuale == "3":
        risposta = "Mi chiamo Assistente Virtuale"
        print(risposta)
    else:
        risposta = "Non ho capito la tua domanda."
        print(risposta)

    while True:
        comando_utente = input("Vuoi farmi un'altra domanda? (Si o No): ")
        if comando_utente == "No":
            print("Arrivederci!")
            break
        elif comando_utente == "Si":
            input(assistente_virtuale())
        else:
            print("Comando non riconosciuto")
            break

assistente_virtuale()
```