

# ESERCIZIO DI DEBUGGING

**Traccia:** 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.

La prima azione da andare ad intraprendere è quella di lettura ed interpretazione del codice per andare a definire la funzione del programma al fine di comprenderne il corretto funzionamento.

Il codice analizzato ha la funzione di andare a generare un assistente virtuale in grado di fornire dati quali data e ora attuali sotto sua esplicita richiesta.

La seconda azione sarà quella di individuare i comportamenti non corretti che implicano il cattivo funzionamento del programma arrestandone l'esecuzione.

Qui di seguito il codice con gli errori rilevati segnalati da un commento <<#>>:

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

Ora andrò ad illustrare l'errore ed il motivo per cui il programma non può funzionare con tale bug:

- **#bug 1:** questo bug è presente sulla riga 14, si tratta dell'assenza dei due punti <<:>> alla fine della riga. Si tratta di un errore di sintassi che non permetterà di far funzionare il ciclo while che serve per avviare l'input da parte dell'utente sin dall'inizio del codice. Questo errore non consente al programma di aprirsi;
- **#bug 2:** il secondo bug è presente sulla riga 4, si tratta di un errore nell'inserire una delle funzioni della libreria datetime (libreria che consente l'inserimento di dati quali data, ora, anno, ecc..), anzichè la scritta datetime.datetime.today(), bisognerebbe trovare la funzione datetime.datetime.today(), modificandola e scrivendola correttamente. Con questo errore il programma si blocca subito dopo il comando "Qual è la data di oggi?";
- **#bug 3:** il terzo bug invece è un errore che non inficia sul funzionamento del programma, ovvero la presenza di .time() sulla riga 7, funzione della libreria datetime che si rivela del tutto inutile al funzionamento del codice in quanto è già più che sufficiente datetime.datetime.now(), pertanto sarebbe meglio eliminarla allo scopo di alleggerire il codice.

Di seguito una immagine del codice corretto e funzionante:

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

**Ricapitolando:** come prima cosa ho aggiunto il doppio punto alla linea 14, dopo `while True`, ho corretto la funzione della libreria `datetime` con la funzione corretta `datetime.datetime.today()` al posto di `datetime.datetoday()` sulla riga 4, ed, infine, eliminato `.today()` dalla riga 7. Così il codice risulta corretto e funzionante e più pulito.

## Miglioramenti

Utilizzando il programma ho riscontrato alcuni problemi a livello di fruibilità, il primo tra tutti è sicuramente l'impossibilità di scrivere il comando da parte dell'utente con caratteri maiuscoli e minuscoli diversi da come sono riportati all'interno del codice, l'utente non può conoscere in modo esatto questa particolarità del programma. A tale scopo ho aggiunto la funzione `.lower()` ad ogni comando (prima presente solo per il comando "esci"), ed eliminato le lettere maiuscole dai vari comandi possibili, così che il comando possa essere utilizzato scrivendo sia con caratteri maiuscoli che minuscoli indistintamente, risultato: il comando verrà ugualmente eseguito, purché sia scritto con i giusti caratteri.

Il secondo problema che mi sono posta riguarda l'impossibilità da parte dell'utente di sapere quali comandi sarà possibile eseguire, quindi ho scelto di renderlo più comprensibile.

Innanzitutto ho aggiunto delle righe di testo a capo del programma in maniera tale da avere una panoramica completa delle possibili scelte, dopodiché ho aggiunto delle condizioni al ciclo `if/else` tali da permettere all'utente una scelta più rapida ed intuitiva tramite numeri, quindi potrà fare le sue richieste al programma digitando sia l'intera domanda oppure digitando solamente il numero corrispondente.

Di seguito il codice redatto con le modifiche cui sopra:

```
File Edit Search View Document Help
import datetime
print("Benvenuto dal tuo Assistente virtuale!")
print("Adesso ti illustrerò cosa so fare.")
print("Ecco qui cosa puoi chiedermi:")
print("1.Qual è la data di oggi?")
print("2.Che ore sono?")
print("3.Come ti chiami?")
print("\n")
def assistente_virtuale(comando):
    if comando.lower() == "qual è la data di oggi?":
        oggi = datetime.datetime.today()
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif comando.lower() == "che ore sono?":
        ora_attuale = datetime.datetime.now()
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    elif comando.lower() == "come ti chiami?":
        risposta = "Mi chiamo Assistente Virtuale"
    elif comando.lower() == "1":
        oggi = datetime.datetime.today()
        risposta = "La data di oggi è:" + oggi.strftime("%d/%m/%Y")
    elif comando.lower() == "2":
        ora = datetime.datetime.now()
        risposta = "Adesso sono le: " + ora.strftime("%H:%M")
    elif comando.lower() == "3":
        risposta = "Il mio nome è Assistente Virtuale, piacere di conoscerti!"
    else:
        risposta = "Non ho capito la tua domanda."
    return risposta
while True:
    comando_utente = input("Cosa vuoi sapere? Puoi inserire anche solo il numero corrispondente alla domanda.")
    if comando_utente.lower() == "esci":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))
```