

```

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.today()
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     return risposta
23

```

Cosa fa il programma:

Il programma è un assistente virtuale testuale, funziona in un ciclo infinito (while True) in cui chiede all'utente: Cosa vuoi sapere?

L'utente può chiedere 3 cose:

- La data di oggi
- L'ora attuale
- Come si chiama l'assistente

Se l'utente scrive "esci", il programma stampa arrivederci! ed esce dal ciclo.

Per tutte le altre domande il programma passa il testo alla funzione assistente_virtuale() che analizza la domanda e restituisce una risposta.

Tabella con tutti gli errori presenti nel codice:

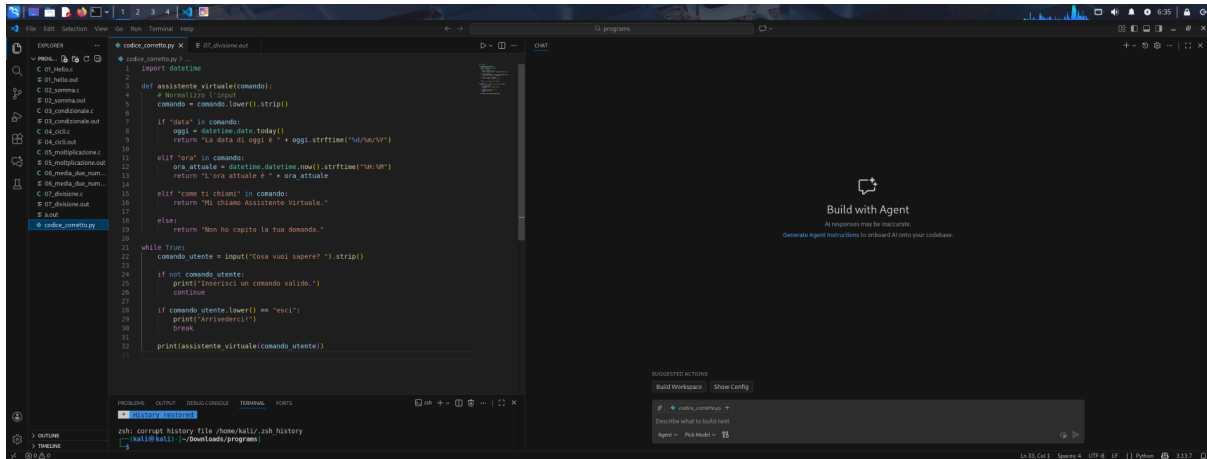
N° Errori	Riga	Tipo errore	Descrizione	Causa	Soluzione
1	2	Sintassi	Manca : dopo while True	Omissione sintattica	while True:
2	7	Logica/indentazione	break è fuori dal blocco if	Indentazione errata	Rientrare il break dentro if

3	13	Sintassi	<code>datetime.datetimeoday()</code>	Metodo sbagliato	<code>datetime.date.today()</code>
4	2-20	Logica	Programma sensibile a maiuscole/minuscole	Confronto testuale molto rigido	Convertire input con <code>.lower()</code>
5	11-20	Logica	Domande funzionano solo se identiche al testo	Match stringa molto precisa	Usare condizioni più generiche ("data" in comando)
6	1-22	Stile/Struttura	Funzione definita dopo la sua chiamata	Struttura disordinata	Spostare la funzione sopra il ciclo
7	21	Sintassi/Indentazione	<code>return</code> risposta ha indentazione incoerente	Rientro sbagliato	Indentare correttamente <code>return</code>
8	15	Stile/coerenza	uso misto di <code>datetime.datetime</code> e <code>datetime.date</code>	Incoerenza nell'uso del modulo	Usare un approccio uniforme
9	3-4	Logica	Nessuna gestione input vuota	Manca validazione	Aggiungere controllo su input
10	2-22	Non-standard	Non gestisce comandi diversi da 3 predefiniti	Scarsa copertura dei casi	Aggiungere gestione generica errori

Il codice non gestisce:

- Input vuoto (solo pressione INVIO)
- Input con errori ortografici ("che ore sono", senza punto interrogativo)
- Input con variazioni ("Che ore sono adesso?")
- Input con spazi extra
- Comandi simili ma non identici ("ora", "tempo", "oggi data", ecc)
- Numeri o simboli
- Input misti ("dimmi la data e l' ora")
- Interruzioni improvvise (CTRL+C)

Codice corretto:



```
1 import datetime
2
3 def assistente_virtuale(comando):
4     # Normalizza il input
5     comando = comando.lower().strip()
6
7     if "data" in comando:
8         oggi = datetime.date.today()
9         return "La data di oggi è " + oggi.strftime("%d/%m/%Y")
10
11     elif "ora" in comando:
12         ora_attuale = datetime.datetime.now().strftime("%H:%M")
13         return "L'ora attuale è " + ora_attuale
14
15     elif "come ti chiami" in comando:
16         return "Mi chiamo Assistente Virtuale."
17
18     else:
19         return "Non ho capito la tua domanda."
20
21 while True:
22     comando_utente = input("Cosa vuoi sapere? ").strip()
23
24     if not comando_utente:
25         print("Inserisci un comando valido.")
26         continue
27
28     if comando_utente.lower() == "esci":
29         print("Arrivederci!")
30         break
31
32     print(assistente_virtuale(comando_utente))
```

Spiegazione finale

Il programma ha lo scopo di simulare un assistente virtuale testuale. L'utente può fare domande sulla:

- Data di oggi
- Ora corrente
- Chiedere all'assistente come si chiama

Il tutto avviene all'interno di un ciclo infinito che termina solo quando l'utente digita "esci". Analizzando il codice senza eseguirlo, emergono diversi problemi di sintassi, logica e struttura. Mancano due punti dopo il costrutto "while True", il comando break è posizionato in modo errato causando l'interruzione immediata del programma, e alcune funzioni del modulo "datetime" sono utilizzate in modo scorretto. Inoltre, il confronto delle stringhe è troppo rigido: il programma funziona solo se l'utente digita le domande in forma esattamente identica a quella prevista. Sono presenti anche errori di indentazione, scarsa gestione degli input non previsti e assenza di commenti, tutti aspetti che compromettono leggibilità e robustezza.

Per ciascuno di questi problemi è stata proposta una soluzione:

- correzione sintattica
- miglioramento della gestione dell'input
- uso coerente del modulo "datetime"
- ristrutturazione del codice per renderlo più chiaro e funzionale.