

esercizio UNIT1 s2_l5

Dato il codice

```
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
```

1. Capire cosa fa il programma senza eseguirlo.
2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
3. Individuare eventuali errori di sintassi / logici.
4. Proporre una soluzione per ognuno di essi.

1: il programma teoricamente dovrebbe impostare una “chat” in cui un assistente virtuale chiede “cosa vuoi sapere?” ad un utente e in base alla risposta può chiudere la chat tramite la richiesta del comando “esci”, rispondere col proprio nome, o rispondere a info su data e ora dato che è stata importata la libreria datetime di python. Nel caso in cui una domanda non fosse chiara riporterebbe “non ho capito la domanda”. Essendo tutto in un ciclo il programma non dovrebbe interrompersi

2: a livello di casistiche, un errore è stato quello di aver impostato un assistente virtuale che risponde solo a specifiche domande che devono esser poste facendo attenzione al case sensitive, al punto di domanda o con un verbo differente. Quindi senza una maiuscola, uno spazio di troppo o una diversa impostazione della frase, rientrerebbero nel ciclo while tramite la risposta “ non ho capito la tua domanda”. Una soluzione sarebbe quindi proporre nel testo stampato in console le specifiche domande che si possono fare per ricevere risposta e convertire il testo dinamicamente tutto in minuscolo o eliminando gli spazi vuoti ad esempio. Sarebbe ancora più comodo se non dovesse scrivere domande specifiche ma magari collegare le domande alle quali sa rispondere con un numero da comunicare all’assistente, così da evitare disguidi. tipo “premi 1 se vuoi sapere l’ora, 2 se vuoi sapere la data” etc altro dubbio a livello logico, è che non si può definire un fuso orario differente.

3-4: partendo dall'alto i problemi col codice sia a livello di sintassi che logici sono:

riga 3, mancano i due punti dopo il `while True` e basterebbe quindi scrivere `"while True :"`

riga 7, il `break` si trova indentato nella posizione errata perchè dovrebbe essere compreso nell'`if`, invece così si trova sullo stesso livello di `if` ed `else`. Quindi bisogna premere tab una volta per farlo avanzare sulla riga.

riga 9, viene chiamata in causa una funzione che però viene definita solo nella riga 11 quindi bisognerebbe scrivere prima `import datetime`, poi la funzione e poi il ciclo `while`.

riga 13, vedendo la documentazione ufficiale di `datetime` da python.org possiamo vedere come importando `datetime` ci siano ad esempio le classi `datetime`, `date`, `time` ma non `datetoday`. Quindi andrebbe risolto scrivendo `datetime.date.today()`

riga 16, dopo avere richiamato `datetime` e il parametro `now()`, viene aggiunto un `-time()` che non fa parte del modulo `datetime` ed è illogico per come è scritto.