

PROGETTO S2/L5

Dato il codice, si richiede di:

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 / logica.
 4. Proporre una soluzione per ognuno di essi.
- Il codice è:

```
1  import datetime
2
3  def assistente_virtuale(comando)
4
5      if comando == "Qual è la data di oggi?"
6
7          oggi = datetime.datetime.today()
8
9          risposta = "la data di oggi è " + oggi.strftime("%d/%m/%Y")
10
11     elif comando == "Che ore sono?":
12
13         ora_attuale = datetime.datetime.now().time()
14
15         risposta = "l'ora attuale è " + ora_attuale.strftime("%H:%M")
16
17     elif comando == "Come ti chiami?":
18
19         risposta = "Mi chiamo Assistente Virtuale"
20
21     else:
22
23         risposta = "Non ho capito la tua domanda."
24
25     return risposta
26
27 while True
28
29     comando_utente = input("Cosa vuoi sapere? ")
30
31     if comando_utente.lower() == "esci":
32
33         print("Arrivederci!")
34
35         break
36
37     else:
38
39         print(assistente_virtuale(comando_utente))
```

1. COSA FA IL PROGRAMMA?

- È un programma scritto in Python che ha lo scopo di fornire informazioni riguardo a data, orario e anche al nome.
- Risponde attraverso comandi di input da parte dell'utente.
- Accetta solo delle domande predefinite, quali: "Qual è la data di oggi?", "Che ore sono?", "Come ti chiami?" ed un input "esci" per terminare il programma.

Funzionamento

1. Ricezione dell'input:

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

2. Verifica dell'input ricevuto attraverso il costrutto **if-elif-else**:

```
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
```

```
if comando_utente.lower() == "esci":
    print("Arrivederci!")
    break
else:
    print(assistente_virtuale(comando_utente))
```

3. Per ogni comando di input restituisce una risposta (**output**) specifica, in un determinato formato. (Si avvale della libreria **datetime** per conoscere data e ora correnti):

```
import datetime
```



```
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"
```



```
if comando_utente.lower() == "esci":  
    print("Arrivederci!")
```

4. Come **output** si ottiene la risposta della funzione stampata a schermo:

```
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."
```

```
if comando_utente.lower() == "esci":
    print("Arrivederci!")
```

2. CASISTICHE NON GESTITE DAL PROGRAMMA

- Guardando il codice si notano diverse situazioni non contemplate dal programma:

1. Lettere maiuscole, minuscole e spazi:

- Il programma è molto rigido ed accetta solo delle risposte digitate esattamente come si aspetta, perciò non accetterà nessun tipo di variazione né nell'assegnazione delle maiuscole né nell'assegnazione delle minuscole. Faccio un esempio:

→ - “Qual è la data di oggi?”, verrà accettato perché riconosciuto dal programma.

- ➔ - “qual è la data di oggi?”, non verrà riconosciuto perché presenta un carattere presentato in maniera diversa (q)
- ➔ - “QUAL È LA DATA DI OGGI?”, anche in questo caso non verrà riconosciuto perché ci sono diversi caratteri mutati.
- Anche nel caso in cui dovessi digitare più spazi di quelli previsti il programma non sarebbe in grado di riconoscere il comando, in quanto anche quegli spazi vengono letti come carattere aggiuntivo, e quindi risulterebbe una stringa più lunga di quella che si aspetta. Esempio:
- ➔ - “Qual è la data di oggi_?”, quello spazio tra oggi e ? verrà calcolato come carattere aggiuntivo e quindi il programma non sarà in grado di riconoscere la stringa.
- In sostanza l’input deve corrispondere esattamente a ciò che il programma si aspetta altrimenti la richiesta verrà scartata!

*Nel punto finale vedremo una dimostrazione

2. Mancanza di una risoluzione

- Quando, come abbiamo visto prima, il comando non viene riconosciuto, appare il messaggio “Non ho capito la tua domanda.”, senza offrire nessun tipo di motivazione che possa aiutare l’utente nella risoluzione dell’errore.

*Nel punto finale vedremo una dimostrazione

3. Mancanza di orario e data variabili (fusi)

- L'orario e la data vengono forniti in maniera standard, senza specificare a quale fuso orario si riferiscono, di conseguenza il programma non tiene presente e traslascia la potenziale posizione dell'utente. Esempio:

➔ - Se il codice viene eseguito allo stesso momento da una persona che si trova in Italia e da una che si trova negli stati uniti, entrambi gli utenti riceveranno la stessa data e la stessa ora come output, cosa che non è assolutamente possibile, in quanto una variabile fondamentale come il fuso influisce sull'orario e talvolta sulla data.

Roma, 06/12/2024, 2 a.m. = New York, 05/12/2024, 8 p.m.

4. Mancanza di indicazioni per uscire dal programma

- Se l'utente non conosce il comando esci per terminare il programma, che deve essere digitato anche in questo caso esattamente come il programma si aspetta, non c'è modo di uscire dal ciclo infinito di **while**; non viene in nessun modo suggerito il comando da inserire né tantomeno una risoluzione in caso di input errato. Esempio:

➔ - Poniamo il caso che una volta ottenute le informazioni richieste io voglia uscire dal programma, per farlo dovrò necessariamente forzare la chiusura (Ctrl + C) perché non conosco l'input che permette di eseguire l'ultimo ciclo if che porta al **break**.

*Nel punto finale vedremo una dimostrazione

3. ERRORI SINTATTICI / LOGICI

Errori di sintassi

- Riga 27: Mancanza dei due punti dopo true - `while true:`

```
while True
```



- Riga 7: Mancanza di un punto tra date e today - `datetime.date_today()`

```
oggi = datetime.datetoday()
```



- Riga 29: Mancanza dei due punti dopo la il punto interrogativo della domanda "Cosa vuoi sapere? - `Cosa vuoi sapere?;`

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



Errori di logica

- Riga 29: La prima informazione che appare all'utente è la domanda "Cosa vuoi sapere?".

Ma non propone nessuna soluzione quindi potenzialmente potrei scrivere qualsiasi cosa senza nemmeno sapere cosa è capace di fare il programma!

Sarebbe il caso di indicare all'utente la lista dei comandi che può eseguire.

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

- Riga 31: Il comando `esci` non viene mai citato all'utente che di conseguenza non sa come terminare il programma, si potrebbe aggiungere magari all'inizio del programma insieme alla lista di comandi possibili.
- Riga 31: Il ciclo **while** si usa generalmente all'inizio del programma in modo che comprenda tutta la logica, anche in questo modo funziona ma pecca di comprensibilità.

4. PROPOSTA DI UNA SOLUZIONE

- Ora per ognuno di questi problemi mettiamo in pratica una soluzione, per farlo riscriviamo il codice correggendo gli errori di sintassi e riorganizzando dove è necessario:

```
import datetime

def assistente_virtuale():
    print("Ciao! Sono il tuo assistente virtuale. Puoi chiedermi:")
    print("Che ore sono?")
    print("Qual è la data di oggi?")
    print("Come ti chiami?")
    print("Oppure scrivi 'esci' per terminare il programma.")
    while True:
        comando_utente = input("Cosa vuoi sapere?: ")
        if comando_utente == "esci":
            print("Arrivederci!")
            break
        elif comando_utente == "Che ore sono?":
            ora = datetime.datetime.now().strftime("%H:%M:%S")
            print("Sono le:", ora)
        elif comando_utente == "Qual è la data di oggi?":
            oggi = datetime.date.today().strftime("%d/%m/%Y")
            print("La data di oggi è:", oggi)
        elif comando_utente == "Come ti chiami?":
            print("Mi chiamo Assistente Virtuale")
        else:
            print("Non ho capito la tua domanda. Puoi chiedermi 'Che ore sono?', 'Qual è la data di oggi?', o 'Come ti chiami?'.")
    assistente_virtuale()
```


- Prima di tutto, con le funzioni **print**, ho fatto in modo di rendere comprensibile all'utente la lista dei possibili comandi.
- Ho reso da subito comprensibile la possibilità di terminare il programma attraverso il comando **esci**.
- Ho spostato il ciclo **while** in alto in modo che anche visivamente possa comprendere tutta la logica del programma.
- Nel caso in cui non venisse inserita la domanda in modo corretto, oltre al messaggio Non ho capito la domanda, ho fatto in modo di ricordare all'utente quali sono le possibilità.
- Ovviamente come si vede ho anche corretto gli errori di sintassi che abbiamo individuato prima.

4. ESECUZIONE DEL PROGRAMMA

- Eseguo il programma per verificare che funzioni correttamente.

```
(kali@kali)-[~]
$ python project.py
Ciao! Sono il tuo assistente virtuale. Puoi chiedermi:
Che ore sono?
Qual è la data di oggi?
Come ti chiami?
Oppure scrivi 'esci' per terminare il programma.
Cosa vuoi sapere?: fgh
Non ho capito la tua domanda. Puoi chiedermi 'Che ore sono?', 'Qual è la data di oggi?', o 'Come ti chiami?'.
Cosa vuoi sapere?: Che ore sono?
Sono le: 09:54:41
Cosa vuoi sapere?: Qual è la data di oggi?
La data di oggi è: 06/12/2024
Cosa vuoi sapere?: Come ti chiami?
Mi chiamo Assistente Virtuale
Cosa vuoi sapere?: esci
Arrivederci!
```

- Tutti i comandi vengono eseguiti correttamente!