

S2/L5

La consegna di oggi consiste nel rispondere ad alcune richieste per il seguente programma:

```
import datetime

def assistente_virtuale(comando):

    if comando == "Qual è la data di oggi?":

        oggi = datetime.date.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))
```

1. Capire cosa fa il programma senza eseguirlo

Leggendo il codice si intuisce che il programma sia un assistente virtuale che si occupa di fornire all'utente (dopo la specifica richiesta) la data di oggi e l'ora attuale.

Il programma ha anche la funzione di rispondere alla domanda "Come ti chiami?" oppure rispondere "Arrivederci" quando l'utente digita il comando "esci".

2. Individuare nel codice sorgente le casistiche non standard che il programma non gestisce

Il programma non riesce a gestire input diversi da quelli indicati nell'immagine seguente

```
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"
```

Quindi in caso di errori di battitura da parte dell'utente oppure di domande diverse rispetto quelle contemplate dal programma l'unico output che l'utente avrà è "Non ho capito la tua domanda".

3. Individuare eventuali errori di sintassi / logici.

Confrontando i comandi "datetime.datetime.today()" e "datetime.datetime.now().time()" ho notato che appunto c'è una discordanza tra i due quindi facendo una ricerca ho capito che il primo comando è sbagliato e dunque il comando giusto fosse:

"datetime.datetime.today()"

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"
```

Dopo aver avviato il programma mi è stato segnalato il seguente errore di sintassi:

```
(kali@kali)-[~/Desktop]
$ python progetto.py
File "/home/kali/Desktop/progetto.py", line 20
    while True
            ^
SyntaxError: expected ':'
```

Quindi rileggendo il codice ho visto la mancanza del segno di punteggiatura ":" dopo il comando "while True".

4. Proporre una soluzione per ognuno di essi.

Prima di tutto ho cominciato correggendo gli errori più banali e semplici da correggere (errori di sintassi).

Quindi ho corretto il comando “datetime.datetime.today()”.

```
if comando == "Qual è la data di oggi?":  
    oggi = datetime.datetime.today()  
    risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
```

In seguito ho corretto anche la sintassi per il ciclo while True.

```
while True:  
    comando_utente = input("Cosa vuoi sapere? ")  
    if comando_utente.lower() == "esci":
```

Dopo aver corretto questi errori rimaneva comunque il problema di cui ho parlato al punto n2, cioè il programma non riesce ad elaborare efficientemente eventuali errori di battitura oppure altre richieste da parte dell'utente.

Per risolvere questo problema in maniera facile senza dover importare altri moduli ho scelto di creare la funzione di un menù in cui ogni comando avrà un numero associato che l'utente dovrà inserire.

Il programma ha un ciclo che viene rotto solo quando l'utente digiterà il comando “Esci”.

Ho usato anche la struttura try except per correggere eventuali errori da parte dell'utente.

Nella prossima pagina segue il programma corretto da me con tutte le variazioni appena dichiarate.

```

GNU nano 8.1
import datetime

def assistente_virtuale(comando):
    if comando == 1:
        oggi = datetime.datetime.today()
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif comando == 2:
        ora_attuale = datetime.datetime.now().time()
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    elif comando == 3:
        risposta = "Mi chiamo Assistente Virtuale"
    else:
        risposta = "Errore. Inserisci un numero corretto"

    return risposta

def menu():
    print("Menu Assistente Virtuale")
    print("Inserisci il numero corrispondente alla tua richiesta")
    print("1: Qual è la data di oggi?")
    print("2: Che ore sono?")
    print("3: Come ti chiami?")
    print("4: Esci")
while True:
    menu()
    try:
        comando_utente = int(input("Inserisci un numero: "))
        if comando_utente == 4:
            print("Arrivederci!")
            break
        else:
            print(assistente_virtuale(comando_utente))
    except ValueError:
        print("Inserisci un numero valido")

```

Per terminare eseguo il programma e dimostro tutti i possibili output in base ai miei input.

La prima immagine mostra l'errore che si verifica inserendo un input errato (ad esempio il numero 5).

```

(kati@kati) ~[~/Desktop]
$ python progetto.py
Menu Assistente Virtuale
Inserisci il numero corrispondente alla tua richiesta
1: Qual è la data di oggi?
2: Che ore sono?
3: Come ti chiami?
4: Esci
Inserisci un numero: 5
Errore. Inserisci un numero corretto

```

La prossima immagine invece mostra gli output ricevuti dopo aver inserito i numeri: 1, 2 e 3.

```
Menu Assistente Virtuale
Inserisci il numero corrispondente alla tua richiesta
1: Qual è la data di oggi?
2: Che ore sono?
3: Come ti chiami?
4:Esci
Inserisci un numero: 1
La data di oggi è 06/12/2024
Menu Assistente Virtuale
Inserisci il numero corrispondente alla tua richiesta
1: Qual è la data di oggi?
2: Che ore sono?
3: Come ti chiami?
4:Esci
Inserisci un numero: 2
L'ora attuale è 06:50
Menu Assistente Virtuale
Inserisci il numero corrispondente alla tua richiesta
1: Qual è la data di oggi?
2: Che ore sono?
3: Come ti chiami?
4:Esci
Inserisci un numero: 3
Mi chiamo Assistente Virtuale
Menu Assistente Virtuale
Inserisci il numero corrispondente alla tua richiesta
1: Qual è la data di oggi?
2: Che ore sono?
3: Come ti chiami?
4:Esci
Inserisci un numero: █
```

Per concludere l'ultima immagine mostra il caso in cui viene digitato il numero 4 che rompe il ciclo e chiude il programma

```
Menu Assistente Virtuale
Inserisci il numero corrispondente alla tua richiesta
1: Qual è la data di oggi?
2: Che ore sono?
3: Come ti chiami?
4:Esci
Inserisci un numero: 4
Arrivederci!
```

```
(kali@kali)-[~/Desktop]
$ █
```

