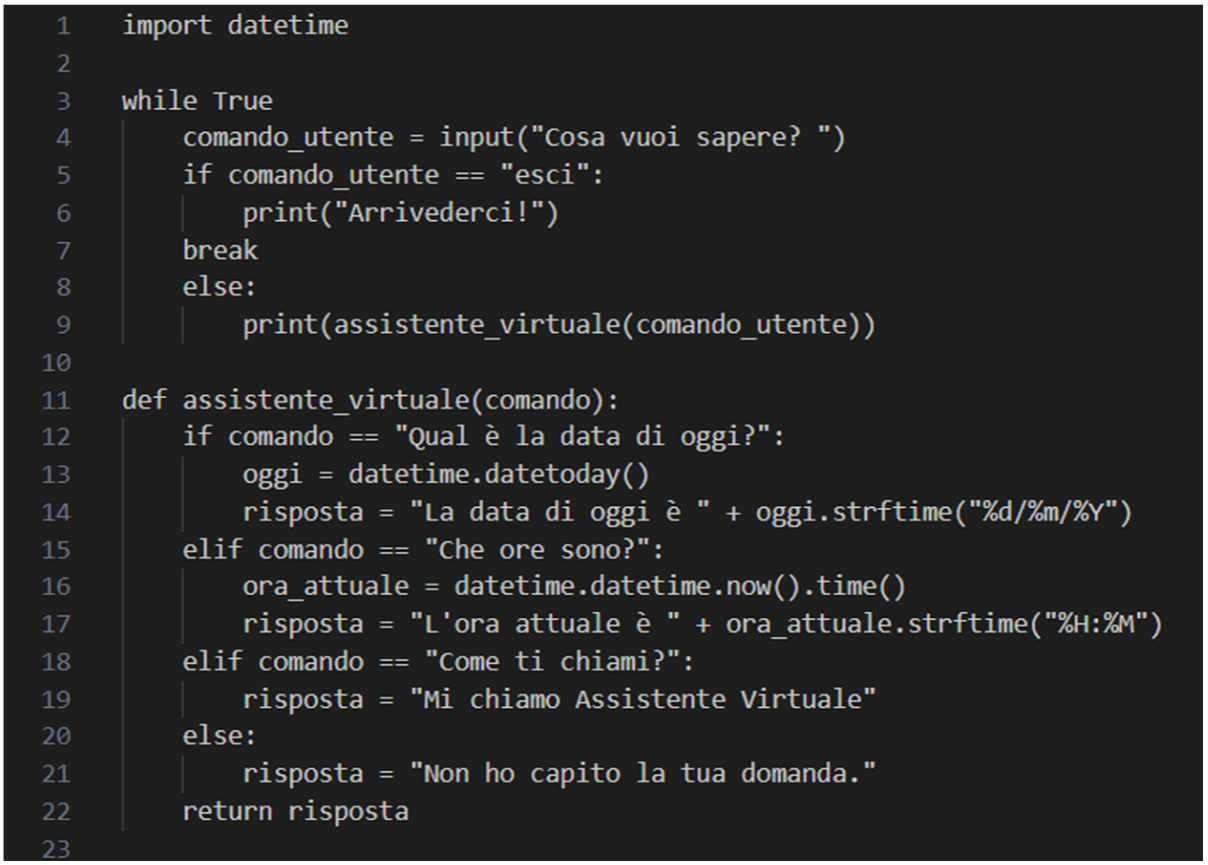
PROGETTO INDIVIDUALE U1-S2-L5

In questo esercizio bisogna fare un'analisi di un codice che ci viene mostrato, individuando gli errori che sono stati commessi.



```
import datetime

while True
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente == "esci":
        print("Arrivederci!")
    break
    else:
        print(assistente_virtuale(comando_utente))

def assistente_virtuale(comando):
    if comando == "Qual è la data di oggi?":
        oggi = datetime.datetoday()
        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

```

FASE 1: Comprensione della funzione del programma

Da una prima occhiata noto che il programma opera innanzitutto con la libreria **datetime**. Andando più nello specifico il programma parte con un **ciclo infinito** dove viene definito **while true**, cioè finché vero, che serve a ripetere ciò che viene inserito all'interno fino a quando non verrà dato un comando in input per interrompere il ciclo, quindi il programma in se.
subito dopo c'è un **if** che regola le possibili scelte dell'utente, nel caso si scelga "esci" viene stampata la stringa "Arrivederci!" ed il programma si chiude con un **break** che interrompe il while; nel caso invece si scelga qualsiasi altra cosa viene richiamata una funzione chiamata **assistente_virtuale** a cui viene passato come **parametro** la risposta data dall'utente.

In questa funzione il **parametro** che viene passato viene utilizzato all'interno di un if/elif per i vari comandi specifici:

-Se il comando è: "**Qual è la data di oggi?**" viene utilizzato un metodo della libreria datetime che estrae la data odierna e viene assegnato ad una variabile "**oggi**".
Subito dopo la variabile viene formattata con il metodo strftime() nel formato classico "**girno/mese/anno**" e concatenata alla stringa "La data di oggi è: " il tutto assegnato alla variabile risposta.

-Se il comando è: "**Che ore sono?**" il processo è molto simile ma viene usato questa volta un metodo(**datetime.datetime.now()**) che estrae sia data che ora ma che poi con l'ausilio di un altro metodo(**.time()**) tiene conto soltanto dell'ora che poi viene assegnata ad una variabile.
Sempre con l'ausilio del comando **strftime()** viene formattato nel formato "**H:M**" ed assegnato alla variabile risposta.

-Se il comando è: "**Come ti chiami?**" Alla variabile risposta questa volta viene assegnata la stringa "Mi chiamo Assistente Virtuale".

-In **qualsiasi altro caso** che non sia presente nella lista alla variabile risposta viene assegnata la stringa "Non ho capito la tua domanda".

La variabile risposta alla fine della funzione viene restituita tramite il comando return risposta.

In sostanza il programma in se tenta di essere un **assistente virtuale** che permette all'utente di porre delle domande specifiche e di ottenere determinate risposte in base alle domande poste.

FASE 2: Individuazione errori sintattici, logici e di esperienza utente

Per questa fase sono partito dall'analisi, in primis, degli errori di sintassi riga per riga che avrebbero potuto causare errori nella compilazione. Sono riuscito ad individuare sveriate criticità:

-Riga 3: **assenza dei 2 punti dopo la condizione del while**; in questo modo python non riesce a capire quando far partire il ciclo.
-Riga 7: **break indentato male**; in python l'indentazione è importante perche non essendoci le parentesi è grazie a quella che il linguaggio capisce dov'è il corpo di un ciclo oppure di una condizione.
-Riga 9: **funzione assistente_virtuale non definita**: essendo python un linguaggio che legge dall'alto verso il basso le funzioni vanno definite all'inizio del codice prima di poterle richiamare. Un'alternativa sarebbe quella di definire una funzione main che gestisce la parte principale del programma, in quel caso si possono inserire le funzioni anche sotto al codice principale poiché il programma quando parte vede il main come prima cosa.
-Riga 13: **metodo datetoday() inesistente**: quando viene richiamata la libreria date time per ottenere la data odierna la sintassi corretta sarebbe:
oggi = datetime.datetime.today()

Passando poi agli errori della gestione dell'esperienza utente sono riuscito ad individuare:

-**Assenza di una definizione precisa dell'utilità del programma all'avvio**.

-**Assenza di una lista di comandi da inserire**: l'utente in questo modo all'avvio del programma non saprà quali sono i comandi precisi per ottenere le informazioni che gli servono.

-**Assenza di case insensitive**: Il programma fa differenza tra maiuscole e minuscole, quindi se l'utente non scrive i comandi esattamente nel formato previsto, potrebbero verificarsi errori o incomprensioni.
Questo si può risolvere con l'ausilio dei metodi **.lower() e .upper()**.

FASE 3: Correzione del codice

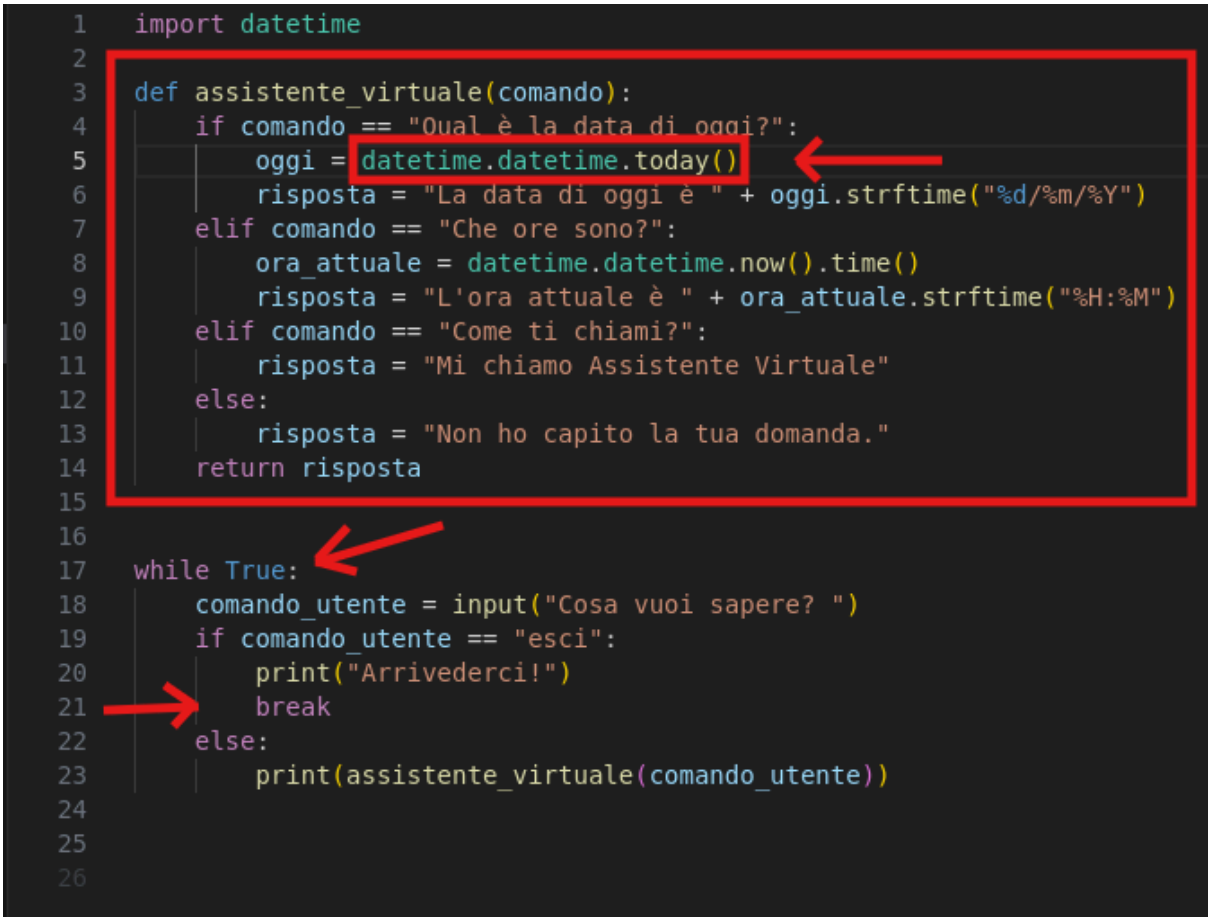
Per la correzione del codice mi sono occupato in primis degli errori sintattici:

-Ho inserito i due punti dopo il while.

-Ho indentato nel modo corretto il break all'interno dell'if.

-Ho spostato la funzione assistente_virtuale() sopra a tutto il codice principale di modo che venisse definita prima di essere richiamata.

-Per ultimo mi sono occupato di correggere la sintassi del metodo datetime nella funzione con la sintassi corretta(datetime.datetime.today()).

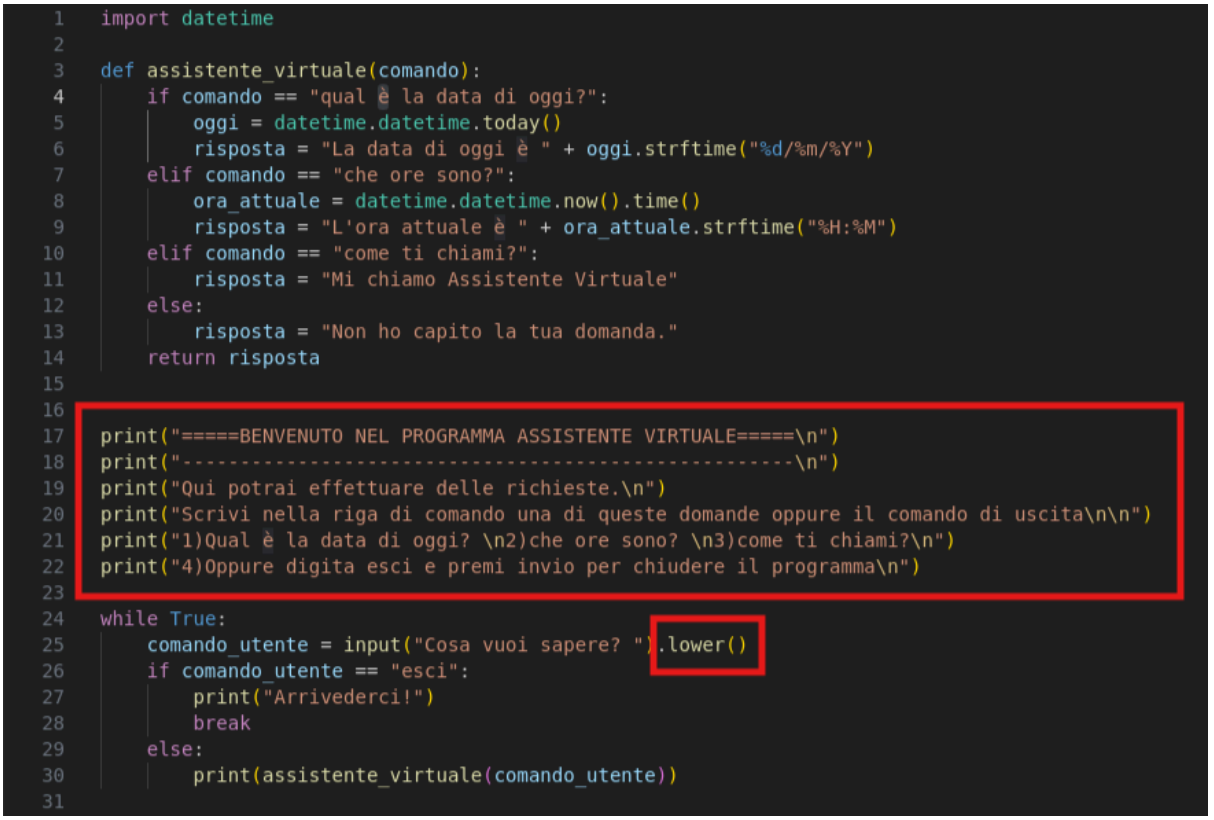


Infine sono passato alla correzione della gestione dell'esperienza utente:

-Innanzi tutto sono andato a far stampare un messaggio al programma che dice all'utente l'utilità del programma stesso.

-Poi sono andato a creare anche una stampa contenente un menù dei comandi possibili ed il comando per uscire dal programma.

-E per ultimo mi sono occupato di inserire il case insensitive con .lower() dopo l'input nel programma di modo che se l'utente dovesse inserire dei caratteri maiuscoli/minuscoli il programma li vedrà sempre corretti per la richiesta fatta.



```python
import datetime

def assistente_virtuale(comando):
    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


print("=====BENVENUTO NEL PROGRAMMA ASSISTENTE VIRTUALE=====\n")
print("-----------------------------------------------------\n")
print("Qui potrai effettuare delle richieste.\n")
print("Scrivi nella riga di comando una di queste domande oppure il comando di uscita\n\n")
print("1)Qual è la data di oggi? \n2)che ore sono? \n3)come ti chiami?\n")
print("4)Oppure digita esci e premi invio per chiudere il programma\n")

while True:
    comando_utente = input("Cosa vuoi sapere? ").lower()
    if comando_utente == "esci":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))

```