PROGETTO S2L5:

- Per agire come un Hacker bisogna capire come pensare fuori dagli schemi. L'esercizio di oggi ha lo scopo di allenare l'osservazione critica.

  Dato il codice si richiede allo studente 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 / logici.
  4. Proporre una soluzione per ognuno di essi.

- **Esecuzione**:
  - **Codice con errori:**

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

  - **Cosa fa il programma?**

- Il programma è un semplice Assistente virtuale che risponde a delle specifiche domande che digita l'utente. Funziona in questo modo:
  - Importazione del modulo **datetime** che permette di ampliare le funzionalità **python** con funzioni per la gestione della **data** e dell'**ora**
  - Il programma è in un ciclo infinito <**while true:** > che, tramite imput dell'utente, scrive una domanda con il prompt <"cosa vuoi sapere?">
  - Se l'utente scrive < esci > (sia in maiuscolo che minuscolo grazie a **.lower())** il programma ti saluta con <"arrivederci"> e termina il ciclo con < **break** >
  - Tramite < **if**,**elif** ed **else** > la funzione < **def assistente_virtuale(comando):** > controlla il testo della domanda e se è una delle previste, risponde così:
    -
  - < **if** comando == "Qual è la data di oggi?": > se la condizione è vera < True >, il blocco di codice indentato sotto < if > viene eseguito. Se la condizione è < False > il blocco di codice viene ignorato. Ti risponde con data/mese/anno;
  - < **elif** comando == "Che ore sono?": > L'istruzione < elif > viene utilizzata per verificare ulteriori condizioni, ma solo se la condizione < if > precedente è risultata falsa. Ti risponde con l'ora attuale con formato < %H:%M >;
  - < **elif** comando == "Come ti chiami?": > L'istruzione < elif > viene utilizzata per verificare ulteriori condizioni, ma solo se la condizione < elif > precedente è risultata falsa. Ti risponde con < "Mi chiamo Assistente Virtuale" >;
  - < **else:** > L'istruzione else fornisce un blocco di codice da eseguire se nessuna delle condizioni precedenti (if e elif) è risultata vera. Risponde con < "Non ho capito la tua domanda." >;
  - Infine, il programma stampa la risposta ricevuta e poi torna a chiederti un'altra domanda;

- **<u>Errori di Sintassi e Logici:</u>**
  - <u>Errore 1</u>: "datetime.datetoday()" .datetoday è un errore perché non esiste nessun metodo con questo nome nel modulo datetime.
    - Soluzione:
      1. datetime.datetime.today(): che ti restituisce data+ore
      2. datetime.date.today(): che restituisce solamente la data
  - <u>Errore 2:</u> "dt.datetime.time().now()".time() è un errore perchè dt.datetime ti restituisce già le ore.
    - Soluzione: è sostituirlo con "dt.datetime.now()"

- Errore 2: “while True” è un errore perché ogni istruzione di controllo (es. If/elif/else e appunto while) deve terminare con i “:” che indicano che sta per iniziare un blocco di codice indentato.
  - Soluzione:
    3. Correggere aggiungendo i “:” a while True.
- Errore 3: “if comando, elif comando, else comando” per far sì che le domande non siano troppo precise a livello di maiuscole e minuscole basta aggiunger .lower.
  - Soluzione:
    1. Indentato sotto “def assistente_virtuale(comando): “aggiungerei una variabile “comando = comando.lower()”
- Errore 4: il programma risponde solo a domande fisse (es.”che ore sono?”), per renderlo più tollerante e flessibile basta sostituire la stringa con parole chiave (es.”che ore”)
  -soluzione:
    1. “if "data" in comando or "giorno" in comando:”
    2. “elif"ora" in comando or "che ore" in comando:”
    3. “elif "come ti chiami" in comando or "il tuo nome" in comando or “nome” in comando:”
- Errore 5: “Import datetime” per una questione di semplicità potresti importare il modulo riassumendolo in “dt”
  - Soluzione:
    1. Sostituire “import datetime” con “import datetime as dt “

- **Codice corretto:**

```
Import datetime as dt

def assistente_virtuale(comando):

    comando = comando.lower()

    if "data" in comando or "giorno" in comando:
        oggi = dt.date.today()
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif "ora" in comando or "che ore" in comando:
        ora_attuale = dt.datetime.now()
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    elif "come ti chiami" in comando or "il tuo nome" in comando or "nome" in comando:
```

```
            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))
```