Progetto S2/L5

Consegna

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.

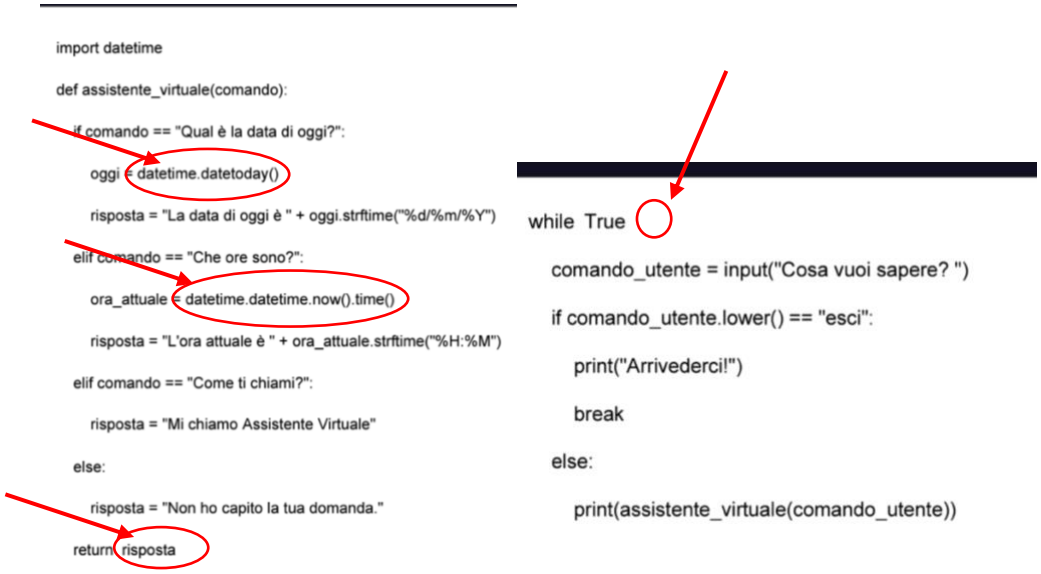
```
import datetime

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))
```

1. **Capire cosa fa il programma senza eseguirlo.**
   Il codice è un assistente virtuale che risponde a 3 comandi. La data di oggi, che ore sono e come ti chiami. Se viene inserito un comando, all'utente, diverso, allora l'assistente virtuale risponde con ''non ho capito la tua domanda.'' Il codice usa un ciclo while per richiedere in continuazione all'utente un comando. Se l'utente scrive esci, allora il ciclo finisce e il codice stampa ''arrivederci''.

2. **Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).**
   I comandi sono stati scritti sia con lettere maiuscole che maiuscole, ma il programma non riconosce questa differenza. Nel caso in cui l'utente preme invio senza scrivere nulla, il caso non viene gestito in nessuno modo (nemmeno con la stampa di arrivederci). Il programma inoltre non riconosce comandi con eventuali spazi in più.

3. Individuare eventuali errori di sintassi/logici.



```
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))
```

Gli errori individuarti sono 4 in totale.

- datetime.datetoday().
- datetime.datetime.now().time().
- variabile ''risposta'' che non è mai stata definita prima di if else
- l'assenza dei : (due punti) dopo il comando while e

4. Proporre una soluzione per ognuno di essi.

- datetime.date.today() – aggiungendo quindi un punto tra le parole date e today.
- datetime.datetime.now() – in questo caso è stato tolto il .()time()
- La variabile risposta dovrebbe essere inizializzata cosi

def assistente_virtuale(comando):
risposta = "Non ho capito la tua domanda.

-Aggiunta dei : dopo while true :