

Progetto S2L5

Traccia:

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:

- Capire cosa fa il programma senza eseguirlo.
- Individuare nel codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).
- Individuare eventuali errori di sintassi / logici.
- Proporre una soluzione per ognuno di essi.

Punto 1. Analisi del codice

Analizzando il codice possiamo notare come si tratti di un programma dove un utente può porre determinate domande ad un assistente virtuale.

```
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))
```

```
import datetime
```

```
def assistente_virtuale(comando):
```

In queste prime righe di codice importiamo il modulo datetime che ci permette di implementare le funzioni per la gestione della data e dell'ora.

Definizione di una funzione con 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
```

Successivamente andiamo ad utilizzare il costrutto if-elif-else.

Tale costrutto ci permette di eseguire un gruppo di istruzioni diverse a seconda del verificarsi di una condizione o più condizioni.

Nel primo if ad esempio se il comando è “Qual è la data di oggi?” utilizza la variabile oggi che acquisisce tramite datetime la data e ci restituisce una risposta richiamandolo attraverso oggi.strftime. Il caso dell’else invece in caso di comando non riconosciuto da parte dell’assistente virtuale ci ritorna una risposta negativa.

```
while True
```

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

```
    if comando_utente.lower() == "esci":
```

```
        print("Arrivederci!")
```

```
        break
```

```
    else:
```

```
        print(assistente_virtuale(comando_utente))
```

Infine utilizziamo il ciclo while. Questo ciclo viene utilizzato per eseguire un blocco di codice affinché una condizione specifica è vera. Con la variabile `comando_utente = input("Cosa vuoi sapere?")` chiediamo all'utente di inserire un comando che verrà poi salvato nella variabile.

Se l'utente digita "esci" il ciclo viene interrotto grazie al `break`, altrimenti verrà chiamata la funzione `assistente_virtuale` definita precedentemente.

2.Casistiche

Come casistiche non contemplate ho individuato il case insensitive nel momento di una richiesta all'assistente virtuale. Un utente di fretta potrebbe non rendersi conto di una maiuscola a inizio frase, un eventuale spazio di troppo o di una mancanza di ? Interpretandola già come domanda.

Un'altra casistica potrebbe essere la non digitazione di un carattere da parte dell'utente, in quanto in questo caso riceverebbe come risposta "non ho capito la domanda" piuttosto che "Errore di digitazione. Inserire un comando"

3.Errorri sintassi/logici

Per quanto riguarda gli errori a livello logico ho individuato:

1) Una mancanza di un messaggio di benvenuto che spiega all'utente che sta avendo a che fare con un assistente virtuale e quali potrebbero essere delle domande da porgli.

2) Una gestione rigida dei comandi in quanto se l'utente non scrive esattamente la domanda, facendo attenzione alla punteggiatura, agli spazi, alle maiuscole ecc, il programma continua a stampargli come risposta "Non ho capito la tua domanda" e gli richiede in loop "Cosa vuoi sapere" senza informare l'utente di quali siano le scelte

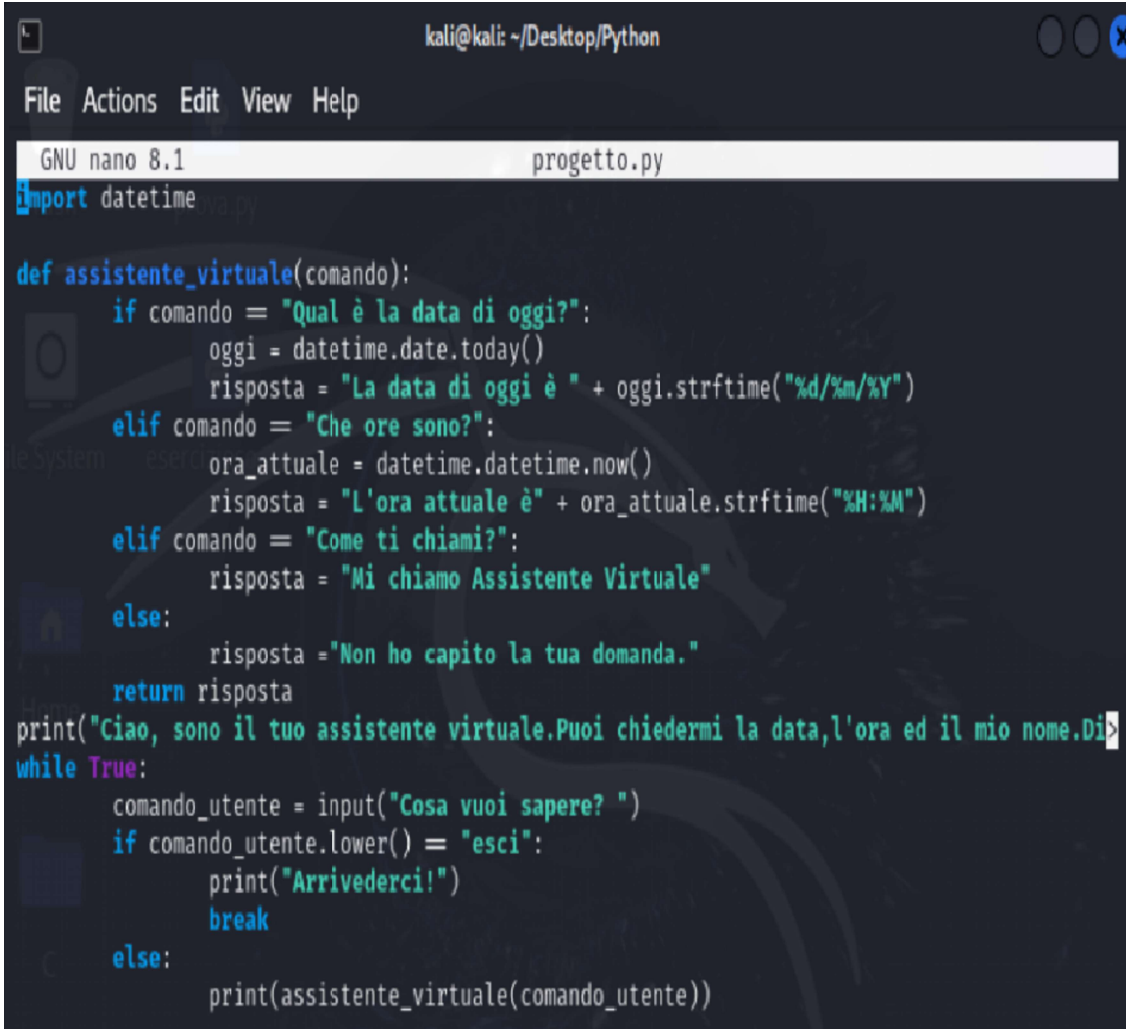
Per quanto riguarda gli errori a livello di sintassi ho individuato:

1)datetime.datetoday()

2)datetime.datetime.now().time()

3)Manca il : al while

4. Soluzione errori di sintassi



```
kali@kali: ~/Desktop/Python
File Actions Edit View Help
GNU nano 8.1 progetto.py
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()
        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("Ciao, sono il tuo assistente virtuale. Puoi chiedermi la data, l'ora ed il mio nome. Di")
while True:
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente.lower() == "esci":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))
```

Questo il codice senza gli errori di sintassi.

1) Manca un (.) tra date e today --> datetime.date.today().

2) .time() è di troppo in quanto .now ci prende l'ora esatta di quel momento --> datetime.datetime.now().

3) fine della riga del while --> while True:

4. Soluzione errori logici

```
kali@kali: ~/Desktop/Python
File Actions Edit View Help
GNU nano 8.1 progetto.py
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()
        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("Ciao, sono il tuo assistente virtuale. Puoi chiedermi la data, l'ora ed il mio nome. Digli")
while True:
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente.lower() == "esci":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))
```

Per quanto riguarda gli errori logici ho inserito un messaggio di benvenuto per l'utente che spiega quali sono i comandi dell'assistente virtuale e come fare per uscire dal programma.

Esecuzione del programma

```
(kali@kali)-[~/Desktop/Python]
```

```
$ python progetto.py
```

Ciao, sono il tuo assistente virtuale. Puoi chiedermi la data, l'ora ed il mio nome. Digita esci per uscire

Cosa vuoi sapere? Che ore sono?

L'ora attuale è: 06:32

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? Che tempo fa?

Non ho capito la tua domanda.

Cosa vuoi sapere? esci

Arrivederci!