Progetto 2S2

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 in allegato, si richiede allo studente di:

-Capire cosa fa il programma senza eseguirlo
-Individuare dal 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.

CODICE DA ANALIZZARE

```
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 programma è un assistente virtuale che ha lo scopo di rispondere ad alcuni comandi di base inseriti dall'utente.
Possiamo dedurlo da:
Il nome della funzione assistente_virtuale
dai blocchi if-elif-else che sono strutturati in modo da proporre all'utente delle risposte a delle specifiche domande come:
- Quale è la data di oggi?
- Che ore sono?
-Come ti chiami?
C'è anche un opzione per dei comandi non riconosciuti che dice:
-Mi dispiace non ho capito la tua domanda.
Un ulteriore conferma potrebbe essere l'inserimento della libreria datetime che ha lo scopo di fornire informazioni temporali.

```
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))
```

2) Individuare dal codice sorgente le casistiche non standard che il programma non gestisce (esempio, comportamenti potenziali che non sono stati contemplati).

Alcuni comportamenti non contemplati dal programma possono essere:
-Un diverso tipo di formato delle lettere, ad esempio inserendo una lettera maiuscola al posto di una minuscola o inserendo uno spazio in più.
-Comandi scritti in maniera simile ma non del tutto identica, ad esempio se al posto di "come ti chiami?" l'utente chiede "Quale è il tuo nome?" l'output sarà quello indicato dal blocco else ovvero"Non ho capito la tua domanda".
-Un input vuoto. Se l'utente non inserisce alcun tipo di input e si limita a cliccare invio, come output otterrà il "non ho capito la domanda.".

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))
```

Sono presenti alcuni errori che compromettono il corretto funzionamento del programma,ecco quali:
1)Nella riga 4 la funzione dovrebbe essere "datetime.date.today()".
2)Nella riga 8 dovrebbe essere scritta in questo modo "! datetime.datetime.now()".
3)Nella riga 11 la variabile risposta non definita correttamente.
4)Nella riga 15 mancano i 2 punti dopo il while.
5)Nell'ultima riga la funzione viene chiamata in modo errato.

4) Proporre una soluzione per ognuno di essi.
Io aggiornerei il codice in questo modo:

```
 GNU nano 8.1                                    Progetto2S22 *
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

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

in questo modo non dovremo più avere problemi durante l'esecuzione del programma.

```
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ nano Progetto2S22

┌──(kali㉿kali)-[~]
└─$ nano Progetto2S22

┌──(kali㉿kali)-[~]
└─$ python Progetto2S22.py
Cosa vuoi sapere? Qual è la data di oggi?
La data di oggi è 06/12/2024
Cosa vuoi sapere? Che ore sono?
L'ora attuale è 14:53
Cosa vuoi sapere? Come ti chiami?
Mi chiamo Assistente Virtuale
Cosa vuoi sapere? che ora è?
Non ho capito la tua domanda.
Cosa vuoi sapere? esci
Arrivederci!

┌──(kali㉿kali)-[~]
└─$
```

Come possiamo osservare dall'immagine a destra il programma si avvia senza problemi e svolge le funzioni implementate precedentemente nel codice.