

ESERCIZIO

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.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))
```

INDICE DI SVOLGIMENTO:

1. FUNZIONE DEL PROGRAMMA
2. CASISTICHE NON STANDARD NON GESTITE DAL PROGRAMMA
3. ERRORI EVENTUALI DI SINTASSI/LOGICI
4. SOLUZIONE ERRORI
5. CONCLUSIONE

1) FUNZIONE DEL PROGRAMMA

Il programma in questione è un Assistente Virtuale programmato per rispondere alle domande poste dall'utente. In questo determinato programma l'utente può esprimere domande solo riguardanti la data, l'ora e a quale nome risponde il programma (Assistenza Virtuale), per tutte le altre domande il programma risponderà con : " Non ho capito la tua domanda."

2) CASISTICHE NON STANDARD NON GESTITE DAL PROGRAMMA

- Il codice risulta "rigido" poichè risponde a poche domande (è presente una sola domanda per la data, una sola domanda per l'ora ed una sola domanda per la richiesta del nome) e non gestisce le eccezioni come ad esempio le lettere maiuscole nelle domande poste dall'utente. Per risolvere la sua rigidità bisogna inserire una lista nel programma che abbia più funzioni (bisogna inserire più domande) e inserire il comando `.lower`. Es: `if comando.lower () in ["qual è la data di oggi?", "che giorno è oggi?", "qual è la data di questo giorno?", "che data è oggi?"]:`
- Il programma non gestisce le eccezioni, se il programma non riconosce un dato specifico non lo legge. Possiamo gestire questo caso inserendo `try`: subito sotto la riga del `def` e aggiungendo :
`except Exception as e:`
`risposta = f"Si è verificato un errore: {e}"`
permettendo così a chi utilizza il programma di capire che c'è un errore.

3) ERRORI EVENTUALI DI SINTASSI/LOGICI

Nell' analisi del programma sono stati individuati due tipi di errori di sintassi quali :

1. `oggi = datetime.datetoday()`
2. `while True`
3. `ora_attuale = datetime.datetime.now().time()`

4) CORREZIONE ERRORI

Correzione errori individuati :

1. `oggi = datetime.date.today()` inserendo un punto tra date e today.
2. `while True` : aggiungendo i due punti alla fine.
3. `ora_attuale = datetime.date.time.now().time()` aggiungendo un punto tra date e time.

5) CONCLUSIONE

Dopo aver apportato le migliorie sovraccitate il programma risulta funzionare correttamente e permette all' utente di avere maggiori possibilità di porre domande e ricevere comunque la risposta corretta inerente a data, ora e nome del programma.

In calce il programma.

```

import datetime

def assistente_virtuale(comando):
    try:

        if comando.lower () in ["qual è la data di oggi?", "che giorno è oggi?", "qual è la data di
questo giorno?", "che data è oggi?"]:
            oggi = datetime.date.today()
            risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")

        elif comando.lower () in ["che ore sono?", "che ora è?", "mi dici qual'è l'ora?", "sai che ore
sono?"]:
            ora_attuale = datetime.date.time.now().time()
            risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")

        elif comando.lower () in ["come ti chiami?", "chi sei?", "qual è il tuo nome?"]:
            risposta = "Mi chiamo Assistente Virtuale"

        else:
            risposta = "Non ho capito la tua domanda."

    except Exception as e:
        risposta = f"Si è verificato un errore: {e}"

    return risposta
while True:
    comando_utente = input("Cosa vuoi sapere? ").strip()
    if not comando_utente:
        print("Per favore, inserisci un comando valido.")
        continue
    if comando_utente.lower() == "esci":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))

```

main.py



Share

Run

```
1
2 import datetime
3
4 def assistente_virtuale(comando):
5     try:
6
7         if comando.lower () in ["qual è la data di oggi?", "che
            giorno è oggi?", "qual è la data di questo giorno?",
            "che data è oggi?"]:
8             oggi = datetime.date.today()
9             risposta = "La data di oggi è " + oggi.strftime("%d/%m
                /%Y")
10
11
12         elif comando.lower () in ["che ore sono?", "che ora è?", "mi
            dici qual'è l'ora?", "sai che ore sono?"]:
13             ora_attuale = datetime.date.time.now().time()
14             risposta = "L'ora attuale è " + ora_attuale.strftime("%H
                :%M")
15
16
17         elif comando.lower () in ["come ti chiami?", "chi sei?",
            "qual è il tuo nome?"]:
18             risposta = "Mi chiamo Assistente Virtuale"
19
20
21     else:
22         risposta = "Non ho capito la tua domanda."
23
24 except Exception as e:
25     risposta = f"Si è verificato un errore: {e}"
26
27 return risposta
28 while True:
29     comando_utente = input("Cosa vuoi sapere? ").strip()
30     if not comando_utente:
31         print("Per favore, inserisci un comando valido.")
32         continue
33     if comando_utente.lower() == "esci":
34         print("Arrivederci!")
35         break
36     else:
37         print(assistente_virtuale(comando_utente))
38
39
```