S2L5

● 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.

●Questo programma rappresenta un semplice assistente virtuale che risponde a tre domande a ciclo infinito: la data corrente, l'ora corrente e il proprio nome. l'utente può continuare a interagire con l'assistente tramite video fino a quando non decidono di uscire, inserendo "esci".

```
1 import datetime
2 def assistente_virtuale(comando):
3     if comando= "Qual è la data di oggi?":
4         oggi = datetime.datetoday()  #1 Primo Errore
5         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
6     elif comando == "Che ore sono?":
7         ora_attuale = datetime.datetime.now().time() #2 Secondo Errore
8         risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
9     elif comando == "Come ti chiami?":
10        risposta = "Mi chiamo Assistente Virtuale"
11    else:
12        risposta = "Non ho capito la tua domanda." <<<<<
13    return risposta
14 while True   #3 Terzo errore
15    comando_utente = input("Cosa vuoi sapere? ")
16    if comando_utente.lower() == "esci":
17        print("Arrivederci!")
18        break
19    else:
20        print(assistente_virtuale(comando_utente))
21
22
23
24
25
26
```

Program explanation

Il programma come impostato sopra non funziona, ha due problemi principali. **datetime.datetoday()** non esiste, la forma corretta è **datetime.date.today()** altrimenti la funzione non può funzionare. **datetime** è una **libreria** standard usata per le date e gli orari, con determinate classi e moduli. Il secondo problema principale è la sintassi di while True (mancano i due punti **:**). Senza la corretta **sintassi ":"** python non capisce come associare la struttura di controllo come **while,if,for** o anche una funzione e lo script segnalerà un errore **(syntax error line 14)**. Una volta risolti questi problemi il programma funzionerà correttamente anche se ha delle carenze logiche. Se l'utente sbaglia a scrivere o inserisce un qualcosa di sbagliato non c'è nessuna gestione degli input non validi o errori imprevisti ed Inoltre Il menu non è user friendly e non presenta nessuna opzione video oltre a **"cosa vuoi chiedere?"** lasciando **l'User** senza sapere cosa o meno può fare. A finire **.strip** per evitare spazi all'inizio o alla fine da input sbagliati.

In-depth

Con **import datetime,** importiamo la libreria con funzioni per lavorare con date e orari.
La funzione **assistente_virtuale** accetta il parametro chiamato comando.
ha 3 condizioni con risultati diversi, Per la data per l'ora e per il nome.
Viene chiesto un input da parte dell'utente su video e se la risposte sono: **"Qual è la data di oggi?"** chiamerà **datetime.date.today() (Formattandola con %d/%m/%Y)** . Invece per l'orario (con input diverso da parte dell'utente) chiamerà **datetime.datetime.now()time() (formattandola con %H:%M)** e per il nome risponderà con una stringa "Mi chiamo Assistente Virtuale" I risultati vengono poi mostrati a video, uguale per l'errore generico se l'utente scriverà qualunque altro input ripetendosi da capo.

Bug List+Suggestions

#1 Nel primo errore abbiamo la forma datetime.datetoday() errata, bisogna corregerlo con datetime.date.today

#2 Nel secondo "errore" dobbiamo ottenere solo l'ora e datetime.datetime.now() fa al caso nostro, poi per formattarla uno usa %H%M

#3 Il terzo errore è una formattazione sbagliata, alla file di while true c'è bisogno di : per indicare l'inizio di un blocco di codice.

#4 Non è presente alcuna gestione delle eccezioni per gestire input non validi o errori imprevisti. O un menu a video user friendly.

#5 Mancanza di un'istruzione di uscita "chiara". Sebbene ci sia un comando per uscire, non è gestito in modo da fornire un feedback all'utente in caso di input non riconosciuto.

#6 .strip per evitare mis-input da parte dell'utente per gli spazi.

Improving

always

```python
import datetime

def assistente_virtuale(comando):
    comando = comando.strip()
    if comando == "Qual è la data di oggi?" or comando == "1":
        oggi = datetime.date.today()
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif comando == "Che ore sono?" or comando == "2":
        ora_attuale = datetime.datetime.now()
        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
    elif comando == "Come ti chiami?" or comando == "3":
        risposta = "Mi chiamo Assistente Virtuale"
    else:
        risposta = "Devi inserire una domanda valida.\nPuoi anche selezionare da 1 a 4."
    return risposta

def mostra_menu():
    print("Benvenuto nel tuo Assistente Virtuale!")
    print("Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.")
    print("Attenzione: Case Sensitive")
    print("1. Qual è la data di oggi?")
    print("2. Che ore sono?")
    print("3. Come ti chiami?")
    print("4. Esci o 4 per terminare il programma")

while True:
    mostra_menu()
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente.strip() == "4" or comando_utente.lower() == "esci":
        print("Arrivederci!")
        break
    else:
        try:
            risposta = assistente_virtuale(comando_utente)
            print(risposta)
        except Exception as e:
            print("Si è verificato un errore, riprova:", e)

```

Program Explanation and Improvements

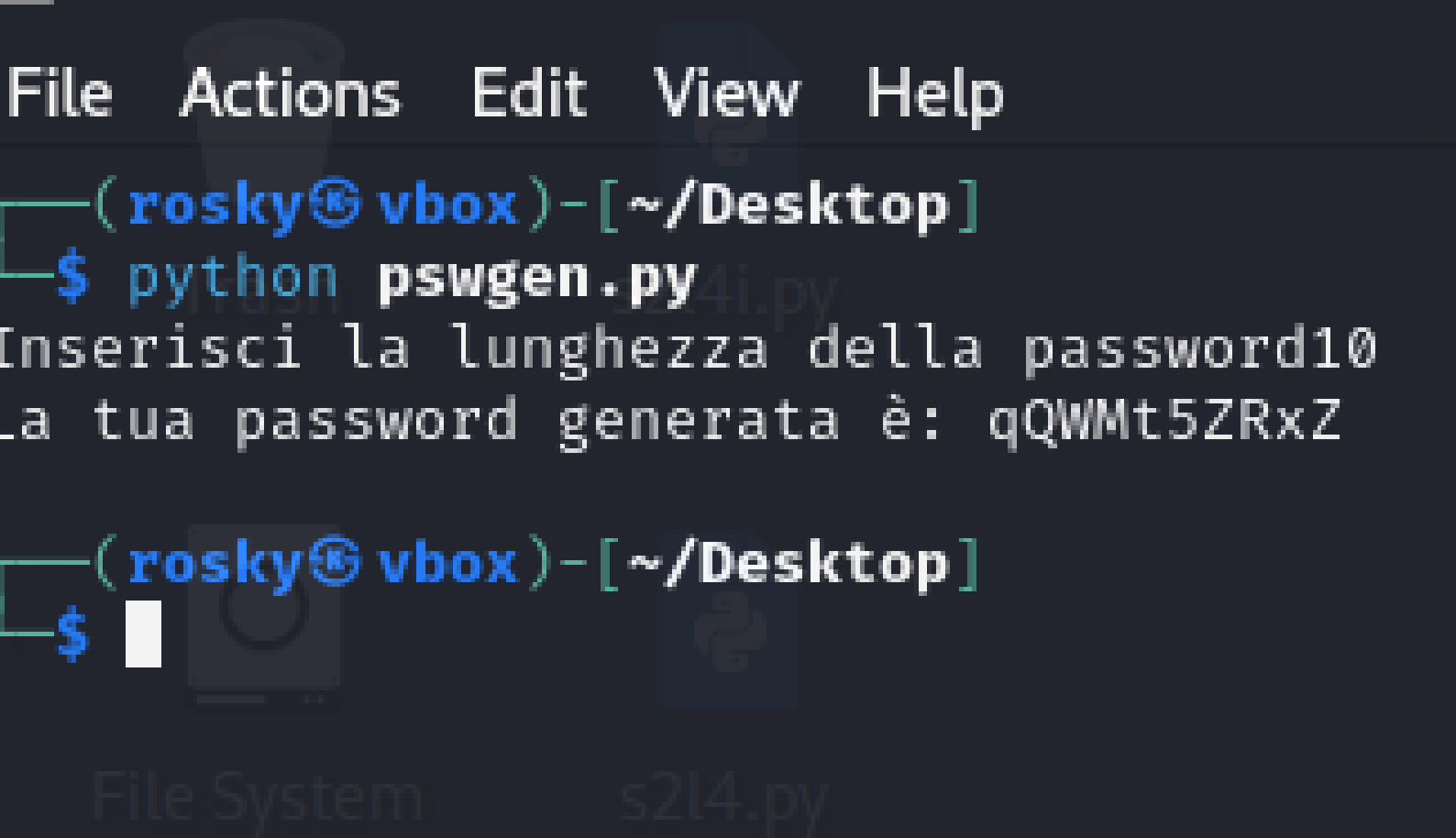
In questa versione aggiornata il programma accetta anche opzioni numeriche oltre alle domande stesse (con possibilità di aggiungere altre opzioni). Il menu è user friendly e spiega le opzioni disponibili su cosa e non si possa fare. I messaggiu di errore sono più chiari e è stato aggiunto un blocco try _except per gestire gli errori e fornire un feedback utile all'Utente. In poche parole è più interattivo e chiaro su cosa o non fare nel programma.

Password Generator (EXTRA)

```python
import random
import string

# psw len
lunghezza = int(input("Inserisci la lunghezza della password"))

# Cosa posso o meno inserire nella password
caratteri = string.ascii_letters + string.digits  # Tutto

# Genera la password casuale
password = ''.join(random.choice(caratteri) for _ in range(lunghezza)) #specificato dall'utente

# Mostra la password
print("La tua password generata è:", password)

```

Programma (basic) per una creazione di una password con l'utilizzo di due librerie.

Con lunghezza = int(input l'utente inserirà la lunghezza della password, mentre
con string.ascii_letters e digits la password uscirà con qualuncue Carattere normale e non o numeri. for_in range per specificare la lunghezza da usare.
alla fine verrà mostrata la password generata in maniera casuale, all'utente.


File Actions Edit View Help
(rosky㉿vbox)-[~/Desktop]
$ python pswgen.py
Inserisci la lunghezza della password10
La tua password generata è: qQWMt5ZRxZ
(rosky㉿vbox)-[~/Desktop]
$
File System
s2l4.py

```
┌──(rosky㉿vbox)-[~/Desktop]
└─$ python S2L5IMP.py
Benvenuto nel tuo Assistente Virtuale!
Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.
Attenzione: Case Sensitive
1. Qual è la data di oggi?
2. Che ore sono?
3. Come ti chiami?
4. Esci o 4 per terminare il programma
Cosa vuoi sapere? 1
La data di oggi è 11/10/2024
Benvenuto nel tuo Assistente Virtuale!
Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.
Attenzione: Case Sensitive
1. Qual è la data di oggi?
2. Che ore sono?
3. Come ti chiami?
4. Esci o 4 per terminare il programma
Cosa vuoi sapere? 2
L'ora attuale è 09:19
Benvenuto nel tuo Assistente Virtuale!
Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.
Attenzione: Case Sensitive
1. Qual è la data di oggi?
2. Che ore sono?
3. Come ti chiami?
4. Esci o 4 per terminare il programma
Cosa vuoi sapere? 3
Mi chiamo Assistente Virtuale
Benvenuto nel tuo Assistente Virtuale!
Puoi selezionare scrivendo 1, 2, 3 o 4 o anche la domanda stessa.
Attenzione: Case Sensitive
1. Qual è la data di oggi?
2. Che ore sono?
3. Come ti chiami?
4. Esci o 4 per terminare il programma
Cosa vuoi sapere? 4
Arrivederci!
```

```python
import datetime
def assistente_virtuale(comando):
    if comando= "Qual è la data di oggi?":
        oggi = datetime.date.today()  #1 Primo Errore
        risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
    elif comando == "Che ore sono?":
        ora_attuale = datetime.datetime.now().time() #2 Secondo Errore
        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:   #3 Terzo errore
    comando_utente = input("Cosa vuoi sapere? ")
    if comando_utente.lower() == "esci":
        print("Arrivederci!")
        break
    else:
        print(assistente_virtuale(comando_utente))
```

Funzionante