REPORT S2L5

Osservazione critica di un programma

Vengono richieste le seguenti verifiche:
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

Procediamo ad analizzare:

Cosa fa il programma?
Il programma chiede all'utente cosa vuole sapere proponendo delle opzioni a cui risponderà un assistente virtuale:
Qual è la data di oggi;
Che ore sono;
Come ti chiami;
Esci (*Qualora l'utente decide di non proseguire*)
All'interno del programma abbiamo subito un ciclo while true

Procediamo con la spiegazione del programma:

Ora ragioniamo da un punto di vista dell'utente che avvia questo programma:
L'utente si troverà di fronte alla seguente domanda "Cosa vuoi sapere?". Qualsiasi utente potrebbe pensare a qualsiasi tipo di risposta che potrebbe anche non seguire il filo logico del programma. Noi che abbiamo scritto il codice ovviamente sappiamo il fine del programma, ma
eseguito così potrebbe essere travisato.

Per tanto io stamperei un menù con le seguenti proposte:

- "Cosa vuoi sapere?":
- 1 *Qual è la data di oggi*?
- 2 *Che ore sono*?
- 3 *Come ti chiami*?
- 4 *Esci dal programma*
-

In questo modo daresti all'utente una chiarezza importante, ma del resto anche al codice stesso che noi, presumibilmente, abbiamo scritto. Quando programmiamo dobbiamo sempre immedesimarci nell'utente che cerca interagire con il programma.

Inoltre per migliorare sempre l'esperienza dell'utente che andrà a interfacciarsi con il nostro eventuale programma aggiungere "/n" nella Quarta riga:
comando utente = input("Cosa vuoi sapere?: /n ")
Così da avere un risultato più o meno così:
Cosa vuoi sapere?..... (Digita risposta)

Errori e soluzioni:

- **Terza riga**: **While True (Il ciclo While esegue un blocco di codice dall'interno del ciclo fin quando la condizione non è valutata Vera)**
  Mancano i " **:** " che servono a inizializzare il ciclo.

- **Quarta riga**: dentro la stringa dettata dall'input della variabile *Comando Utente*
  C'è un errore di sintassi: ("Cosa vuoi sapere? ")
  Mancano i due punti: ("Cosa vuoi sapere?: ").
  Se non lo modifichiamo quando l'utente riceve la domanda gli apparirà
  Cosa vuoi sapere? senza i due punti.

- **Settima riga**: Nella settima riga troviamo un errore : **Break** presenta un errore di indentazione: **Break(***break point*) all'interno di un programma serve per impostare una pausa o un interruzione.
  Nel codice che ci hanno fornito il **Break** inserito fuori la condizione **IF**.
  Pertanto andremo spostare il nostro **Break** sotto **Print** così da farlo rientrare nel ciclo

- **Nona riga**: Non possiamo chiamare una **Funzione** senza prima averla dichiarata..
  (Ricordiamoci sempre che Python è un linguaggio di alto livello orientato agli oggetti che si esegue dall'alto verso il basso).
  Possiamo spostare **Def** alla riga **2**

- **Tredicesima riga**
  Import datetime (libreria)
  All'inizio del programma leggiamo "**Import datetime**", ciò significa che abbiamo importato la *libreria(modulo) datetime* per eseguire determinate casistiche. Prima di verificare se il nostro codice è corretto o meno, quello che potremmo fare è andare a controllare all'interno della libreria *datetime* per vedere quali classi ci sono all'interno. *Possiamo controllare queste librerie ad esempio nel sito ufficiale di python.*
  *Nella tredicesima riga leggiamo***: oggi == datetime.date today()**
  Facendo un controllo sulla libreria di python vediamo che **date today()** non è riportato. Dovremmo sostituirla con .**time** che è invece presente all'interno della libreria del codice riutilizzabile ***Datetime***

Nota:
Notate bene che questa esercitazione prevedeva l'analisi di questo programma senza utilizzare python.
L'esercizio voleva che noi cominciassimo ad abituare il nostro occhio come fossimo un hacker per individuare ipotetici malfunzionamenti del codice.