

Progetto S2/L5

Il progetto attuale ha lo scopo di mettere alla prova la nostra capacità analitica e di pensare come un Hacker. Dato un codice già creato, bisogna rispettivamente:

- Capire cosa fa il programma senza eseguirlo
- Individuare nel codice sorgente eventuali anomalie e comportamenti non contemplati
- Individuare errori logici e di sintassi
- Proporre una soluzione per ognuno di essi

Ecco dunque come si presenta il codice, analizziamolo passo passo:

```
import datetime
```

```
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
```

```
while True
```

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

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

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

```
        break
```

```
    else:
```

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

Spiegazione codice

Per prima cosa viene importato un modulo di default chiamato **"datetime"**.

Viene definita una funzione chiamata **"assistente virtuale"** che prende in considerazione i valori scritti dall'utente .

In base alla scelta dell'utente, si aprono varie casistiche determinate dal costrutto if-elif-else.

Nel primo caso viene creato un oggetto **"oggi"** che restituisce il metodo `datetime.today()` e successivamente l'oggetto **"risposta"** restituirà una stringa in cui ci scriverà la data di oggi specificando giorno, mese e anno.

Lo stesso procedimento si applica alla seconda casistica, in cui viene chiesto però che ore sono

Spiegazione codice

Nel secondo caso viene creato un oggetto `"ora_attuale"` che richiama dalla libreria i metodi `time()` e `now()` la cui funzione è quella di restituire in output l'orario corrente. In risposta verrà stampato a schermo l'orario, specificando ora e minuti.

Nel terzo caso, invece, se noi chiediamo al programma "come ti chiami?" lui risponderà "Sono il tuo assistente virtuale". A qualunque altra cosa, risponderà "Non ho capito la tua domanda." e chiederà un nuovo input all'utente.

Alla fine del codice il costrutto `while` stabilisce che, mentre tutte le casistiche risultano vere, il programma ripeterà il ciclo chiedendoci ogni volta la domanda: "Cosa vuoi sapere?" e seguirà poi una nostra risposta. Uscirà dal ciclo solamente nel caso in cui noi scriveremo la parola "esci". A quel punto il programma risponderà : "Arrivederci" e si chiuderà il ciclo.

Anomalie

- Per quanto riguarda l'individuazione di comportamenti inattesi da parte del programma si potrebbe dire che l'output atteso considera sempre, ad ogni domanda, la lettera maiuscola all'inizio di ogni domanda. Ad esempio "**C**he ore sono?". Però, se io provassi a riformulare la stessa domanda scrivendo in minuscolo "che ore sono?" il programma mi risponderebbe dicendo "Non ho capito la tua domanda". Lo stesso discorso si può allargare anche ad altri esempi (sbagliando un accento, non inserendo uno spazio o il punto di domanda finale). Questo risulterebbe alquanto ironico, nonchè illogico.

Errori di sintassi

- Per quanto riguarda errori di sintassi si può dire che dopo "while True" ci vogliono i ":", altrimenti il programma non funzionerebbe.