

# **Progetto in python.**

In questo esercizio si andrà ad allenare la visione critica.

Ci vengono fatte 4 richieste:

- 1 capire cosa esegue il programma
  - 2 individuare nel codice sorgente le casistiche non standard che il programma non gestisce
  - 3 individuare eventuali errori di sintassi / logici
  - 4 proporre una soluzione per ognuno di essi.
- Prima di andare a rispondere alla prima richiesta, andiamo a vedere il programma che ci viene fornito.

# Programma python

```
1 |
2 import datetime
3
4 def assistente_virtuale(comando):
5
6     if comando=="qual e la data di oggi?":
7
8         oggi=datetime.datetime.today()
9
10        risposta="la data di oggi e"+oggi.strftime("%d/%m/%y")
11
12    elif comando=="che ore sono?":
13
14        ora_attuale=datetime.datetime.now().time()
15
16        risposta="l'ora attuale e"+ora_attuale.strftime("%H:%M")
17
18    elif comando=="come ti chiami?":
19
20        risposta="mi chiamo assistente virtuale"
21
22    else:
23
24        risposta="non ho capito la domanda."
25
26    return risposta
27
28    while true:
29
30        comando_utente=input("cosa vuoi sapere?")
31
32        if comando_utente.lower()=="esci":
33
34            print("arrivederci")
35
36            break
37
38        else:
39
40            print(assistente_virtuale(comando_utente))
41
```

Questo è il programma a noi fornito che dovremo analizzare. Possiamo già da subito capire lo scopo di questo programma, se non quello di rispondere alle domande poste che abbiamo programmato in precedenza.

Perciò qualora noi andassimo a chiedere al programma “che ore sono?”, ciò che ci andremo ad aspettare che lui risponda è ovviamente l’ora in quel preciso istante in cui lo andiamo a chiedere e questo in modo totalmente istantaneo.

Prendendo in considerazione sempre la stessa domanda “che ore sono?” bisogna far attenzione, in questo singolo caso, a non digitarlo con maiuscole perché altrimenti il programma non riuscirà a leggerlo rispondendovi “non ho capito la domanda”.

# **Problemi presenti nel programma**

Ora che abbiamo fatto una piccola presentazione del programma, andremo a vedere gli errori presenti che non ci permettono di avviarlo, in questo caso quelli che sono riuscito ad individuare sono:

la presenza di indentazione errate tra le righe del codice, che nel caso di python è fondamentale l'indentazione, questo vuol dire che il programma quando proveremo ad avviarlo, ci darà questa risposta:

```
(kali㉿kali)-[~]  
$ python prova.py  
File "/home/kali/prova.py", line 34  
    print("arrivederci")  
    ^  
IndentationError: expected an indented block after 'if' statement on line 32
```

un altro errore presente, si trova nella riga 8, dove possiamo vedere un codice errato cioè “datetime.datetoday()”.

In questo caso il codice oltre che errato viene a risultare inesistente, perciò lo andremo a cambiare con “datetime.datetime.today()”

prima:

```
8 oggi=datetime.datetoday()
```

dopo:

```
8 oggi=datetime.datetime.today()
```

continuando a vedere, possiamo notare un errore nella riga 28 con il codice “while true” che anche in questo caso viene scritto in modo errato, per correggerlo bisogna scrivere “true” con l’iniziale maiuscola “True” ed infine aggiungere i due punti alla fine, ottenendo così “while True:”.

Prima:

```
28 while true|
```

dopo:

```
28 while True:
```

correggendo i vari errori possiamo andare a vedere il risultato finale del programma.

# Correzione programma

```
1 import datetime
2
3 def assistente_virtuale(comando):
4     if comando == "qual e la data di oggi?":
5         oggi = datetime.datetime.today()
6         risposta = "la data di oggi è " + oggi.strftime("%d/%m/%Y")
7     elif comando == "che ore sono?":
8         ora_attuale = datetime.datetime.now().time()
9         risposta = "l'ora attuale è " + ora_attuale.strftime("%H:%M")
10    elif comando == "come ti chiami?":
11        risposta = "mi chiamo assistente virtuale"
12    else:
13        risposta = "non ho capito la domanda."
14
15    return risposta
16
17 while True:
18     comando_utente = input("cosa vuoi sapere? ")
19     if comando_utente.lower() == "esci":
20         print("arrivederci")
21         break
22     else:
23         print(assistente_virtuale(comando_utente))
24
```

Ora possiamo notare come sono diminuite le righe e anche come i codici corretti sembrerebbero essere attivi.

Non ci resta altro da fare se non quello di testare se funziona.

# *Test programma*

Per iniziare innanzitutto dobbiamo vedere se il terminale ci permette di avviarlo.

Per andarlo ad avviare dovremo semplicemente digitare nel terminale il codice “python nomedelfile.py” nel nostro caso “python progetto1.py”

```
(kali㉿kali)-[~]  
$ python progetto1.py  
cosa vuoi sapere? █
```

Possiamo notare che il nostro programma si è avviato, questo vuol dire che siamo riusciti a correggerlo, ora proviamo a porre la domanda “che ore sono?” e vediamo cosa succede.

```
cosa vuoi sapere? che ore sono?  
l'ora attuale è 07:32  
cosa vuoi sapere? █
```

Anche questa volta possiamo notare che il programma ci fa ricevere una risposta in maniera immediata, togliendo ogni dubbio riguardo la sua funzionalità.

## **Conclusione progetto**

Ora che abbiamo la massima certezza che il programma funziona correttamente, possiamo considerare tutti i problemi presenti risolti e di conseguenza anche il progetto terminato con successo.