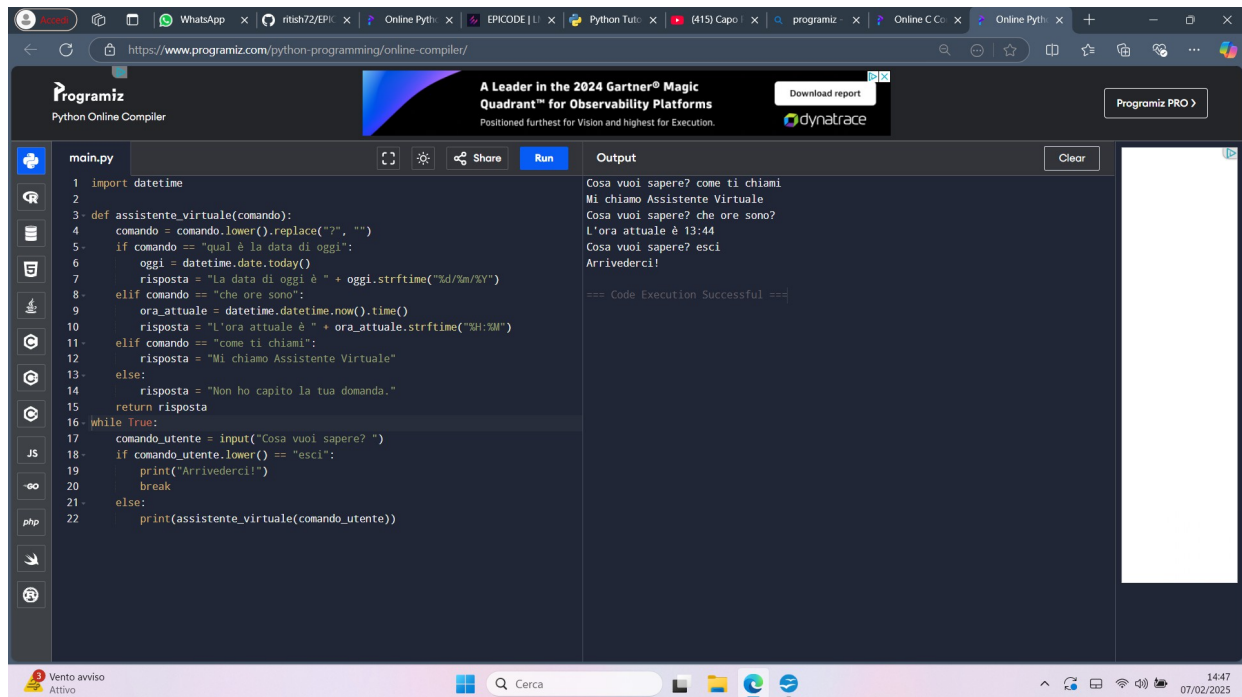


RELAZIONE

1. L'esercizio che ci è stato dato permette di chiedere al programma la data odierna, l'orario, con chi stiamo parlando e inoltre in caso scrivessimo un qualcosa che non c'entrasse con le domande, che ci vengono date dal programma, riceveremo un messaggio, dal programma stesso, in cui ci verrà detto che la macchina non avrà capito la nostra domanda e quindi di inserire una di quelle da lui date. Infine potremo anche uscire dal programma usando il comando "esci" e il programma ci saluterà. Come possiamo vedere il codice è formato dal comando "if", "elif", "else", i quali permettono di creare cicli all'interno di un codice per dimostrare se una condizione sia vera o falsa.
2. Tra le casistiche non standard del codice troviamo il caso in cui io, utente che va a digitare le domande, possa scrivere le parole date dal programma senza delle probabili virgole, punti esclamativi o delle mancate lettere minuscole e/o maiuscole, e ti risponda con il messaggio dell'ultimo else in cui ti dice che non ha capito, creando dei casi in cui non si hanno delle eccezioni nonostante delle semplici mancanze, per poter rimediare a questa cosa, possiamo usare la convenzione "camelCase" per rendere le lettere minuscole e le altre maiuscole o sennò il pascalCase per rendere maiuscole solo le prime lettere.
 - Un'altra casistica potrebbe essere l'orario visto che se chiedessimo al programma di dirci l'orario, esso possa variare a seconda del sistema che stiamo usando, ad esempio sul sito programiz, viene dato l'orario UTC mentre se andassimo sulla macchina virtuale, ci darebbe l'orario impostato sul sistema operativo;
3. Tra gli errori all'interno del codice abbiamo:
 - L'assenza dei ":" dopo il "while true" che, deve essere sempre messo perchè sta ad indicare l'inizio di un possibile blocco di comando, di un ciclo while e un ciclo for e inoltre, viene usata all'interno degli if e nelle definizioni di funzioni;
 - Un altro errore è il comando "datetoday" il quale non esiste, dando quindi un errore, il reale comando è "datetime.date.today"
- 4.



The screenshot shows the Programiz Python Online Compiler interface. The left pane contains a Python script named `main.py` with the following code:

```
1 import datetime
2
3 def assistente_virtuale(comando):
4     comando = comando.lower().replace("?", "")
5     if comando == "qual è la data di oggi":
6         oggi = datetime.date.today()
7         risposta = "La data di oggi è " + oggi.strftime("%d/%m/%Y")
8     elif comando == "che ore sono":
9         ora_attuale = datetime.datetime.now().time()
10        risposta = "L'ora attuale è " + ora_attuale.strftime("%H:%M")
11    elif comando == "come ti chiami":
12        risposta = "Mi chiamo Assistente Virtuale"
13    else:
14        risposta = "Non ho capito la tua domanda."
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))
```

The right pane shows the output of the program:

```
Cosa vuoi sapere? come ti chiami
Mi chiamo Assistente Virtuale
Cosa vuoi sapere? che ore sono?
L'ora attuale è 13:44
Cosa vuoi sapere? esci
Arrivederci!

=== Code Execution Successful ===
```

SPIEGAZIONE DEL PROGRAMMA E DELLE ISTRUZIONI UTILIZZATE

Come abbiamo detto nel punto 1 questo programma visualizza il giorno, l'orario e il nome dell'assistente. Vediamo la funzione `def` che serve a definire una funzione o dei parametri. Successivamente vediamo il comando `.lower` il quale ci permette di rendere le lettere di una parola minuscole, senza però creare alcun problema se la parola venisse scritta tutta in maiuscola, accanto vediamo invece il comando `.replace` che appunto ci fa rimuovere un determinato carattere, in questo caso il `"?"` con uno spazio vuoto, anche in questo caso se volessimo inserire il punto interrogativo comunque il programma non darebbe alcun problema. Da qui inizia il ciclo di `if`, `elif` ed `else`, che ci mettono davanti delle condizioni da scegliere, e attraverso le domande che andremo a fare, avremo delle risposte diverse. Gli `elif` sono altre condizioni che vengono fatte all'interno dell'`if` se abbiamo più di due condizioni da controllare, infine abbiamo l'`else` che chiude questo ciclo e serve quando una delle opzioni date dall'utente risulta essere `"falsa"`, se vogliamo rifarci ad un linguaggio booleano, oppure sbagliata, nel nostro caso appunto, avremo la risposta da parte del programma che non ha capito la domanda, riponendoci quindi quest'ultima. Alla fine del ciclo troviamo `"return"`, questo comando ci farà visualizzare la risposta alla nostra domanda. Poi troviamo il comando `"while true"`, anche questo è un ciclo, che in questo caso fino a quando la condizione sarà vera rimarrà sempre all'interno del ciclo, in questo caso questo ciclo può essere interrotto solo manualmente dall'utente, che nel nostro caso sarà la domanda che ci verrà fatta ogni volta dal programma ovvero `"che cosa`

vuoi sapere?”, poi ritroviamo di nuovo il comando if-else che in questo caso ci fa scegliere se uscire dal programma e quindi farci dire da quest'ultimo “arrivederci”, per poi infine chiudere il ciclo del while true tramite il comando break.