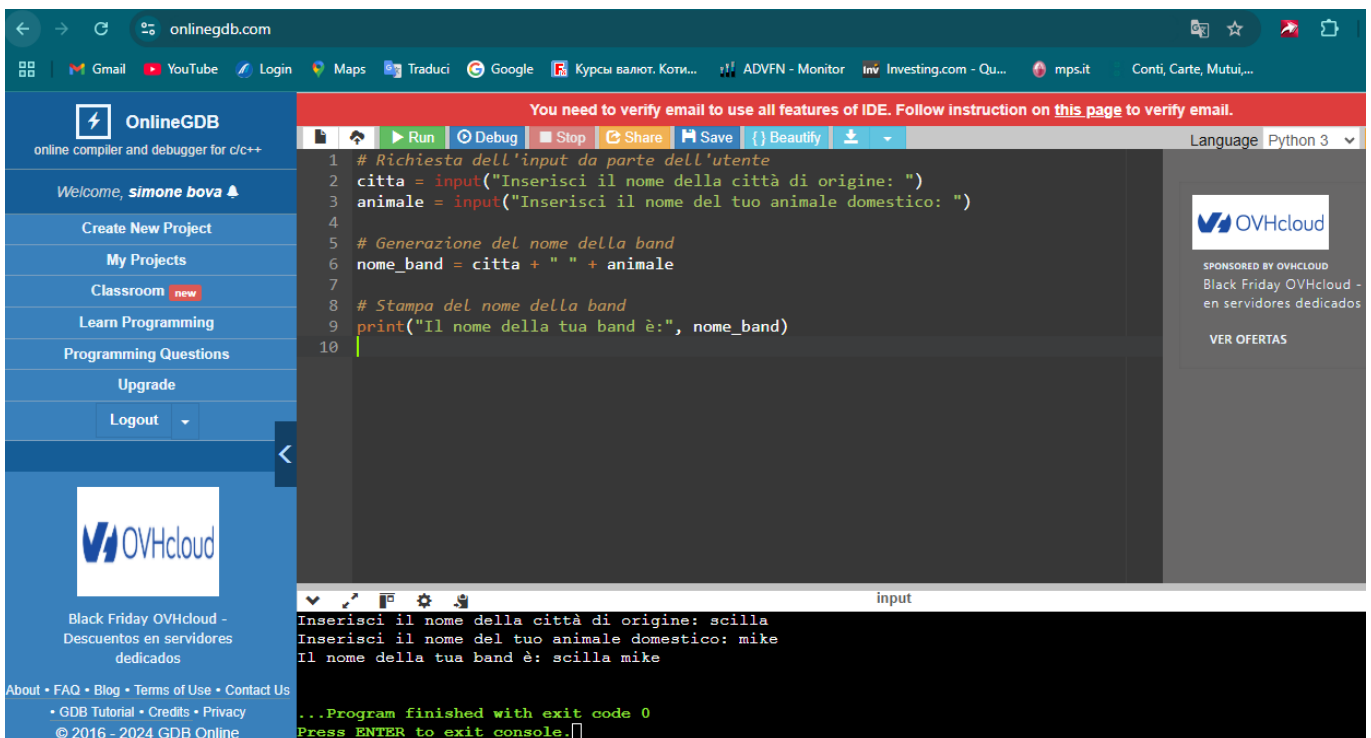


GENERAZIONE DI UN NOME PER LA MIA BAND

In questo esercizio, ho creato un programma che chiede all'utente di inserire due informazioni: il nome della mia città natale e il nome del proprio animale domestico. Dopo aver ricevuto questi dati, il programma li combina insieme per formare il nome di una band, unendo i dovuti input con uno spazio tra di loro. Alla fine, il programma stampa il nome della band generata. L'obiettivo era creare un nome divertente per una band partendo da due informazioni personali.



The screenshot shows the OnlineGDB web interface. The browser address bar displays 'onlinegdb.com'. The left sidebar contains navigation links: 'Welcome, simone bova', 'Create New Project', 'My Projects', 'Classroom' (marked as 'new'), 'Learn Programming', 'Programming Questions', 'Upgrade', and 'Logout'. The main editor area shows a Python script with the following code:

```
1 # Richiesta dell'input da parte dell'utente
2 citta = input("Inserisci il nome della città di origine: ")
3 animale = input("Inserisci il nome del tuo animale domestico: ")
4
5 # Generazione del nome della band
6 nome_band = citta + " " + animale
7
8 # Stampa del nome della band
9 print("Il nome della tua band è:", nome_band)
10
```

Below the code editor, the input/output console shows the execution results:

```
input
Inserisci il nome della città di origine: scilla
Inserisci il nome del tuo animale domestico: mike
Il nome della tua band è: scilla mike
...Program finished with exit code 0
Press ENTER to exit console.
```

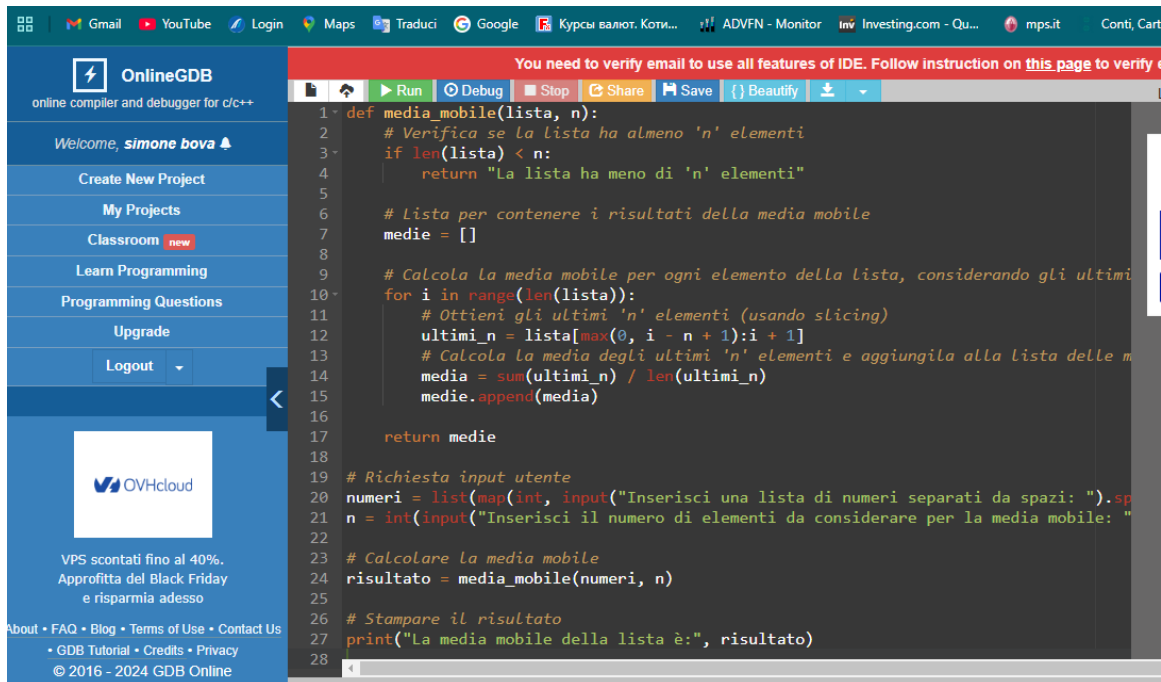
On the right side of the interface, there is a red banner stating 'You need to verify email to use all features of IDE. Follow instruction on this page to verify email.' Below this, there is an OVHcloud logo and a sponsored message: 'SPONSORED BY OVHcloud Black Friday OVHcloud - en servidores dedicados VER OFERTAS'.

esercizio bonus 1

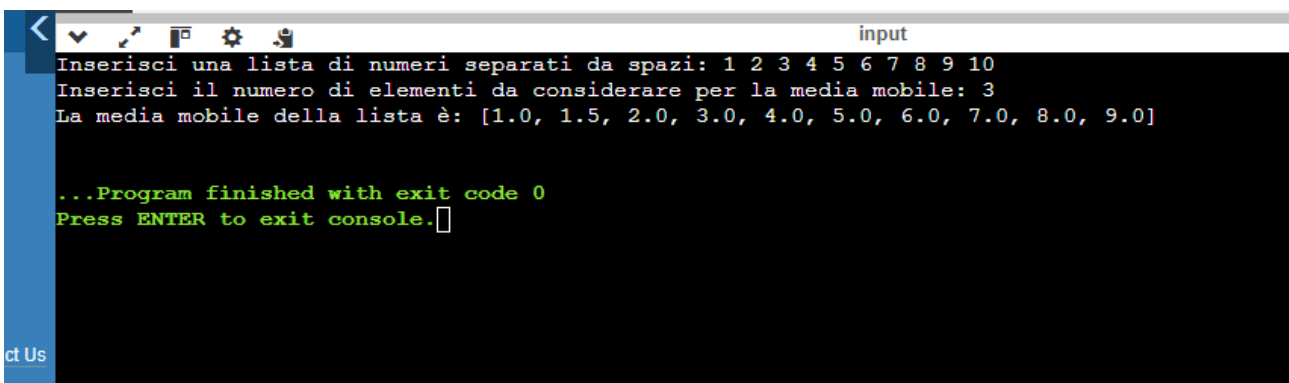
CALCOLARE LA MEDIA MOBILE

Ho creato un programma che calcola la media mobile di una lista di numeri. La funzione principale prende due argomenti: una lista di numeri e un numero n , che indica quanti numeri considerare per ogni calcolo della media. La funzione prima verifica se la lista ha abbastanza elementi per fare il calcolo; se è troppo corta, restituisce un messaggio di errore. Successivamente, per ogni elemento nella lista, calcola la media degli ultimi n numeri, aggiornando la finestra di calcolo man mano che si sposta lungo la lista. Infine, restituisce una lista di medie.

Il programma chiede all'utente di inserire una lista di numeri e il valore di n , e poi mostra i risultati delle medie mobili. Se la lista ha meno di n numeri, viene restituito un messaggio che avvisa dell'impossibilità di calcolare la media mobile. Ad esempio, con la lista $[1, 3, 5, 7, 9]$ e $n = 3$, il programma calcola la media dei numeri in finestre di 3 elementi: per il primo numero è solo 1, per il secondo è la media di 1 e 3, per il terzo la media di 1, 3, 5, e così via. Il risultato sarà una lista di medie, come $[1.0, 2.0, 3.0, 5.0, 7.0]$. Questo programma aiuta a calcolare la media mobile, utile per analizzare dati e ridurre il rumore nelle informazioni.



```
1 def media_mobile(lista, n):
2     # Verifica se la lista ha almeno 'n' elementi
3     if len(lista) < n:
4         return "La lista ha meno di 'n' elementi"
5
6     # Lista per contenere i risultati della media mobile
7     medie = []
8
9     # Calcola la media mobile per ogni elemento della lista, considerando gli ultimi
10    for i in range(len(lista)):
11        # Ottieni gli ultimi 'n' elementi (usando slicing)
12        ultimi_n = lista[max(0, i - n + 1):i + 1]
13        # Calcola la media degli ultimi 'n' elementi e aggiungila alla lista delle m
14        media = sum(ultimi_n) / len(ultimi_n)
15        medie.append(media)
16
17    return medie
18
19 # Richiesta input utente
20 numeri = list(map(int, input("Inserisci una lista di numeri separati da spazi: ").sp
21 n = int(input("Inserisci il numero di elementi da considerare per la media mobile: ")
22
23 # Calcolare la media mobile
24 risultato = media_mobile(numeri, n)
25
26 # Stampare il risultato
27 print("La media mobile della lista è:", risultato)
28
```



```
input
Inserisci una lista di numeri separati da spazi: 1 2 3 4 5 6 7 8 9 10
Inserisci il numero di elementi da considerare per la media mobile: 3
La media mobile della lista è: [1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0]

...Program finished with exit code 0
Press ENTER to exit console.
```

esercizio bonus 2

CONTA PAROLE

Ho creato questo programma che analizza una stringa di testo e restituisce un dizionario con il conteggio delle occorrenze di ciascuna parola. Per fare in modo che il conteggio non dipenda dalle maiuscole o minuscole, ho usato il metodo `lower()` per convertire tutto il testo in minuscolo. Ho anche rimosso la punteggiatura utilizzando il modulo `re` per mantenere solo le parole e gli spazi. Poi, ho diviso il testo in parole e ho usato un dizionario per contare quante volte ogni parola appare. Alla fine, il programma restituisce un dizionario con le parole come chiavi e il numero di occorrenze come valori.

```
You need to verify email to use all features of IDE. Follow instruction on this page to verify e

1 import re
2 def conta_parole(testo):
3     # Converti il testo in minuscolo per rendere il conteggio case-insensitive
4     testo = testo.lower()
5
6     # Rimuovi la punteggiatura usando il modulo re (manteniamo solo lettere e numeri)
7     testo = re.sub(r'[^\w\s]', '', testo)
8
9     # Dividi il testo in parole
10    parole = testo.split()
11
12    # Crea un dizionario per memorizzare il conteggio delle parole
13    conteggio = {}
14
15    # Conta le occorrenze di ciascuna parola
16    for parola in parole:
17        if parola in conteggio:
18            conteggio[parola] += 1
19        else:
20            conteggio[parola] = 1
21    return conteggio
22
23 # Richiesta dell'input utente
24 testo = input("Inserisci un testo: ")
25
26 # Chiamata alla funzione e stampa del risultato
27 risultato = conta_parole(testo)
28 print("Conteggio delle parole:", risultato)
```

```
20 conteggio[parola] = 1
input
Inserisci un testo: Ciao sono Simone e vivo a Scilla
Conteggio delle parole: {'ciao': 1, 'sono': 1, 'simone': 1, 'e': 1, 'vivo': 1, 'a': 1, 'scilla': 1}

...Program finished with exit code 0
Press ENTER to exit console.
```