

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. [Sign in](#)

Python Coding Questions for Data Engineer Interview Part-I (Easy Level)

Mastering Python For Data Engineering Interview.



Kamireddy Mahendra · [Follow](#)

Published in Stackademic · 6 min read · Jan 18, 2024

89



“A learning curve is essential to growth.”

Keep learning:)

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



Photo by [Roman Synkevych](#) on [Unsplash](#)

As a Data Engineer, mastering the art of dissecting problem logic is indispensable for navigating the intricacies of real-world data infrastructure projects.

The ability to devise optimized solutions hinges on the exploration of diverse problem-solving strategies. By tackling problems through various avenues, individuals can hone their problem-solving acumen, fostering continuous skill development.

In this endeavor, I've embarked on solving a spectrum of problems, ranging from basic data structures to complex machine learning models. You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in with [Sign in](#).

These challenges are carefully curated from recent data engineering interviews, serving as litmus tests for candidates' problem-solving prowess and analytical aptitude in basic-level data engineering interviews.

The Essential PySpark Cheat Sheet for Data Engineers.

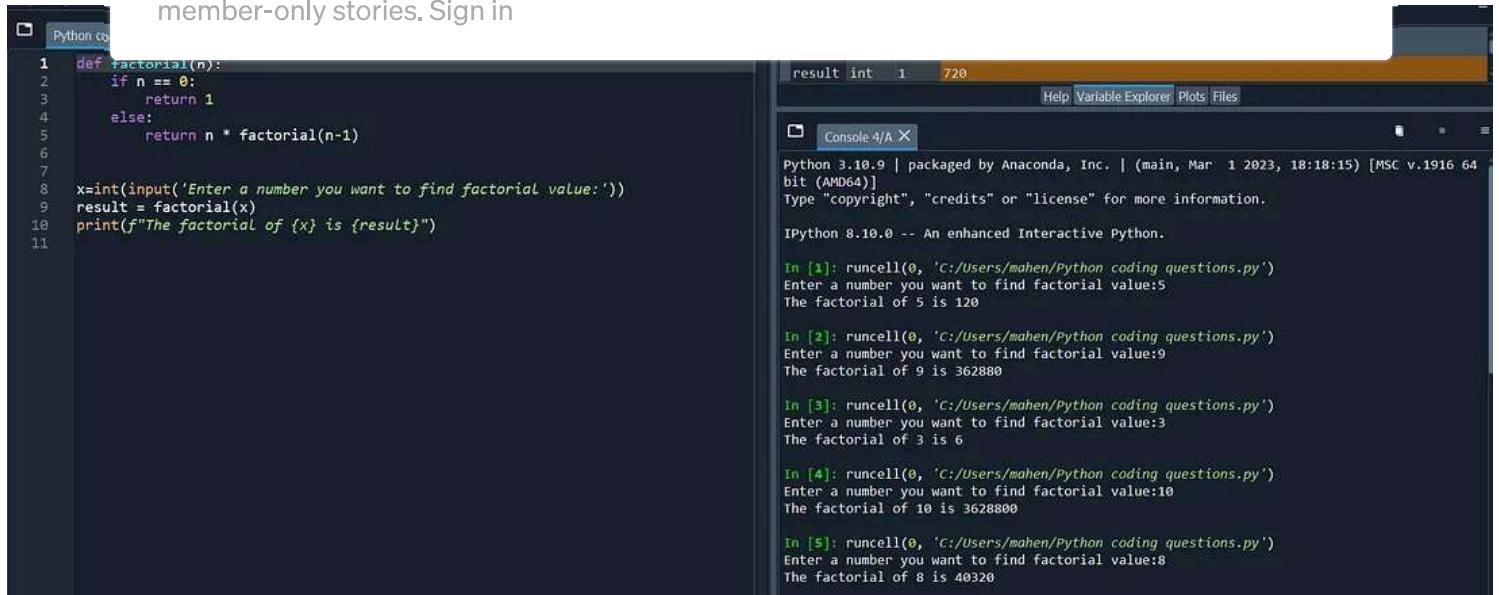
Let's get started.

Q1) Write a program to find the factorial of a number using recursion.

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

x=int(input('Enter a number you want to find factorial value:'))
result = factorial(x)
print(f"The factorial of {x} is {result}")
```

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



The screenshot shows a Jupyter Notebook interface. On the left, a code cell contains Python code for calculating the factorial of a number using recursion. On the right, a console cell shows the execution of the code and its output for various input values (5, 9, 3, 10, 8).

```

1 def factorial(n):
2     if n == 0:
3         return 1
4     else:
5         return n * factorial(n-1)
6
7
8 x=int(input('Enter a number you want to find factorial value:'))
9 result = factorial(x)
10 print(f"The factorial of {x} is {result}")
11

```

```

result int 1 720
Help Variable Explorer Plots Files
Console 4/A X
Python 3.10.9 | packaged by Anaconda, Inc. | (main, Mar 1 2023, 18:18:15) [MSC v.1916 64
bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.10.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'c:/Users/mahen/Python coding questions.py')
Enter a number you want to find factorial value:5
The factorial of 5 is 120

In [2]: runcell(0, 'c:/Users/mahen/Python coding questions.py')
Enter a number you want to find factorial value:9
The factorial of 9 is 362880

In [3]: runcell(0, 'c:/Users/mahen/Python coding questions.py')
Enter a number you want to find factorial value:3
The factorial of 3 is 6

In [4]: runcell(0, 'c:/Users/mahen/Python coding questions.py')
Enter a number you want to find factorial value:10
The factorial of 10 is 3628800

In [5]: runcell(0, 'c:/Users/mahen/Python coding questions.py')
Enter a number you want to find factorial value:8
The factorial of 8 is 40320

```

Factorial of a number using recursion, image designed by author [Kamireddy Mahendra](#)

Q2) Write a program to find the square of the largest element in an array.

```

def find_squared_largest(arr):
    largest = arr[0]
    for i in arr:
        if i > largest:
            largest = i
    return largest**2

```

Open in app ↗

[Sign up](#)

[Sign in](#)

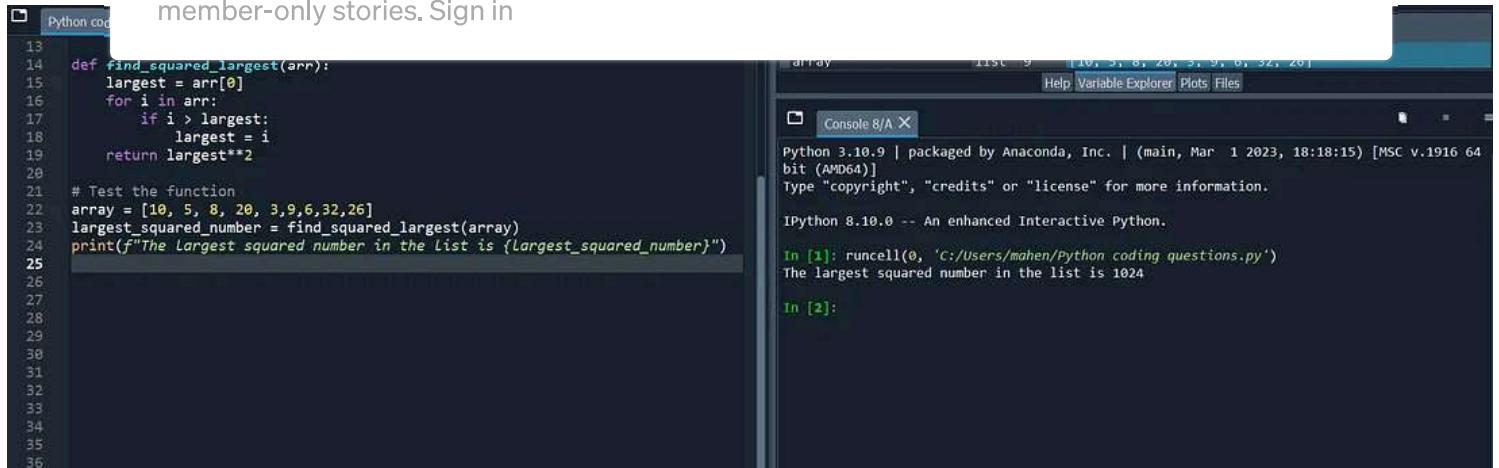
 Medium

 Search

 Write



You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



```

13 def find_squared_largest(arr):
14     largest = arr[0]
15     for i in arr:
16         if i > largest:
17             largest = i
18     return largest**2
19
20
21 # Test the function
22 array = [10, 5, 8, 20, 3, 9, 6, 32, 26]
23 largest_squared_number = find_squared_largest(array)
24 print(f"The Largest squared number in the List is {largest_squared_number}")
25
26
27
28
29
30
31
32
33
34
35
36

```

Squared of the largest number in a list, an image designed by author [Kamireddy Mahendra](#)

Q3) Write a program to find the number of times a number exists in a list. return in descending order of number of occurrences a number exists.

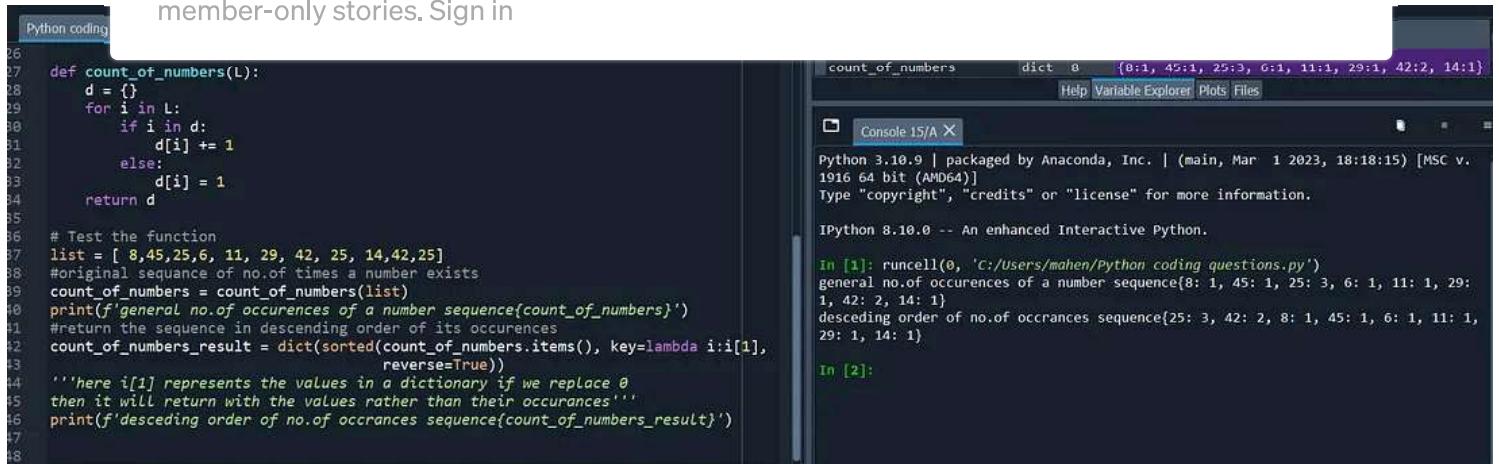
```

def count_of_numbers(L):
    d = {}
    for i in L:
        if i in d:
            d[i] += 1
        else:
            d[i] = 1
    return d

list = [ 8,45,25,6, 11, 29, 42, 25, 14,42,25]
#original sequence of no.of times a number exists
count_of_numbers = count_of_numbers(list)
print(f'general no.of occurences of a number sequence{count_of_numbers}')
#return the sequence in descending order of its occurrences
count_of_numbers_result = dict(sorted(count_of_numbers.items(), key=lambda i:i[1]
                                         reverse=True))
''' here i[1] represents the values in a dictionary if we replace 0
then it will return with the values rather than their occurrences'''
print(descending order of no.of occurrences sequence{count_of_numbers_result})

```

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



```

26
27 def count_of_numbers(L):
28     d = {}
29     for i in L:
30         if i in d:
31             d[i] += 1
32         else:
33             d[i] = 1
34     return d
35
36 # Test the function
37 list = [8, 45, 25, 6, 11, 29, 42, 25, 14, 42, 25]
38 #original sequence of no.of times a number exists
39 count_of_numbers = count_of_numbers(list)
40 print(f'general no.of occurrences of a number sequence{count_of_numbers}')
41 #return the sequence in descending order of its occurrences
42 count_of_numbers_result = dict(sorted(count_of_numbers.items(), key=lambda i:i[1],
43                                     reverse=True))
44 '''here i[1] represents the values in a dictionary if we replace 0
45 then it will return with the values rather than their occurrences'''
46 print(f'descending order of no.of occurrences sequence{count_of_numbers_result}')
47
48

```

count_of_numbers dict 8 {8:1, 45:1, 25:3, 6:1, 11:1, 29:1, 42:2, 14:1}

Console 15/A

Python 3.10.9 | packaged by Anaconda, Inc. | (main, Mar 1 2023, 18:18:15) [MSC v. 1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.10.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
general no.of occurrences of a number sequence{8: 1, 45: 1, 25: 3, 6: 1, 11: 1, 29: 1, 42: 2, 14: 1}
descending order of no.of occurrences sequence{25: 3, 42: 2, 8: 1, 45: 1, 6: 1, 11: 1, 29: 1, 14: 1}

In [2]:

No.of occurrences of a number in a list, image designed by author [Kamireddy Mahendra](#)

Q4) write a program to find a number that is prime or not prime.

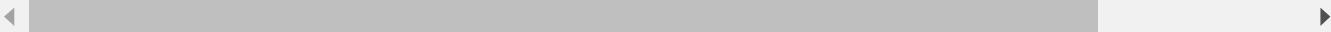
Yes, Interviewers often seem to ask simple questions in data engineering roles not to assess the straightforwardness of the task but to gauge the candidate's problem-solving approach.

They aim to discern whether the candidate tackles challenges with optimization strategies or resorts to brute force methods. The focus lies in understanding how candidates optimize solutions, demonstrating their ability to efficiently address problems.

For instance, when determining whether a number is prime or not, iterating through the entire range up to the given number exhibits a time complexity of $O(n)$, representing a brute-force approach. Conversely, employing an optimized method with a time complexity of $O(\sqrt{n})$ allows for quicker determination of primality. This showcases the candidate's capability to utilize optimization techniques for enhanced efficiency in problem-solving scenarios.

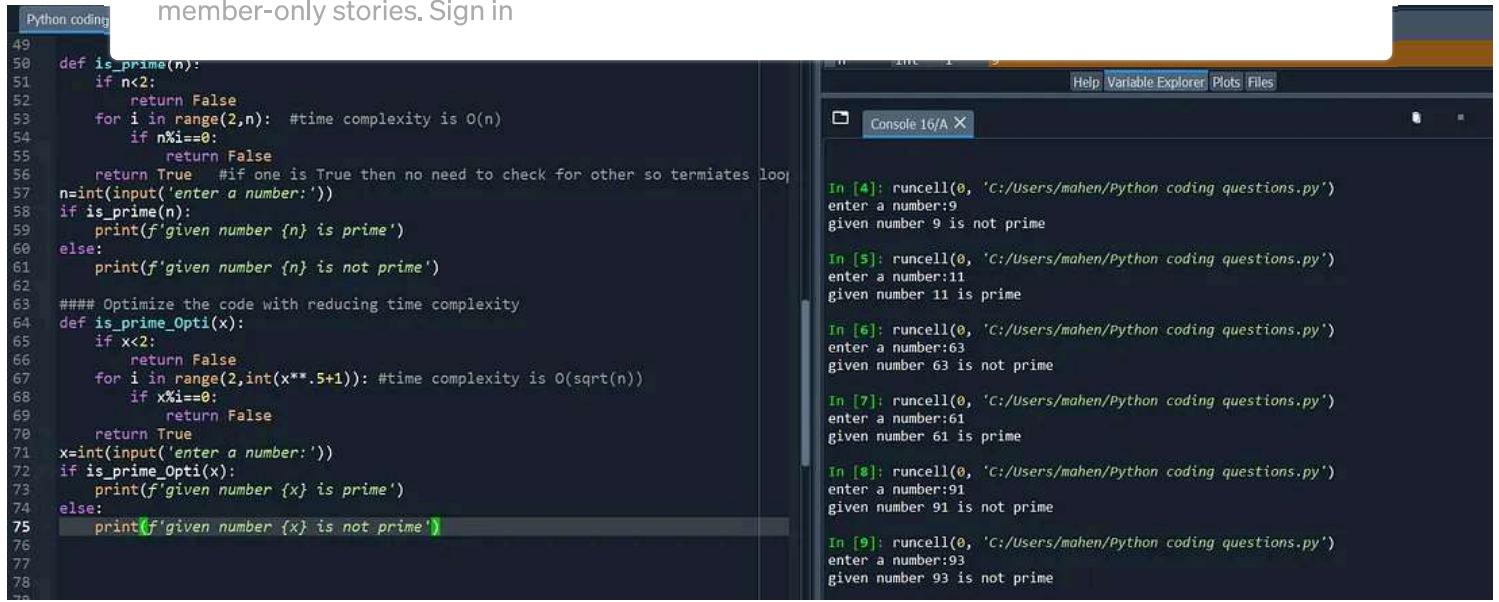
You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. [Sign in](#)

```
        return False
    for i in range(2,n): #time complexity is O(n)
        if n%i==0:
            return False
    return True #if one is True then no need to check for the other so terminate
n=int(input('enter a number:'))
if is_prime(n):
    print(f'given number {n} is prime')
else:
    print(f'given number {n} is not prime')
```



```
#### Optimize the code by reducing time complexity
def is_prime_Opti(x):
    if x<2:
        return False
    for i in range(2,int(x**.5+1)): #time complexity is O(sqrt(n))
        if x%i==0:
            return False
    return True
x=int(input('enter a number:'))
if is_prime_Opti(x):
    print(f'given number {x} is prime')
else:
    print(f'given number {x} is not prime')
```

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



The image shows a Jupyter Notebook interface. On the left, there is a code editor with Python code. On the right, there is a console window showing the execution of the code and its output. The code in the editor is as follows:

```

49
50 def is_prime(n):
51     if n<2:
52         return False
53     for i in range(2,n): #time complexity is O(n)
54         if n%i==0:
55             return False
56     return True #if one is True then no need to check for other so terminates loop
57 n=int(input('enter a number:'))
58 if is_prime(n):
59     print(f'given number {n} is prime')
60 else:
61     print(f'given number {n} is not prime')
62
63 ##### Optimize the code with reducing time complexity
64 def is_prime_Opti(x):
65     if x<2:
66         return False
67     for i in range(2,int(x**.5+1)): #time complexity is O(sqrt(n))
68         if x%i==0:
69             return False
70     return True
71 x=int(input('enter a number:'))
72 if is_prime_Opti(x):
73     print(f'given number {x} is prime')
74 else:
75     print(f'given number {x} is not prime')

```

The console window shows the following output:

```

In [4]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
enter a number:9
given number 9 is not prime

In [5]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
enter a number:11
given number 11 is prime

In [6]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
enter a number:63
given number 63 is not prime

In [7]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
enter a number:61
given number 61 is prime

In [8]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
enter a number:91
given number 91 is not prime

In [9]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
enter a number:93
given number 93 is not prime

```

Prime or not with optimization, the image designed by author [Kamireddy Mahendra](#)

Q5) write a program to delete duplicates and return the list without duplicates.

```

def delete_duplicates(arr):
    res = []
    for i in arr:
        if i is not in res:
            res.append(i)
    return res

array = [1, 2, 3, 2, 1, 3, 2, 4, 5, 4]
res = delete_duplicates(array)
reverse_res=sorted(delete_duplicates(array), reverse=True)
print(f'given array is: {array}')
print(f'array without duplicates is: {res}')
print(f'array without duplicates in reverse order is:{reverse_res}')

```

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in

```

Python coding q
9
0 def delete_duplicates(arr):
1     res = []
2     for i in arr:
3         if i not in res:
4             res.append(i)
5     return res
6
7 # Test the function
8 array = [1, 2, 3, 2, 1, 3, 2, 4, 5, 4]
9 res = delete_duplicates(array)
reverse_res=sorted(delete_duplicates(array), reverse=True)
1 print(f'given array is: {array}')
2 print(f'array without duplicates is: {res}')
3 print(f'array without duplicates in reverse order is:{reverse_res}')
4
5
6
7
8
9

```

array list 10 [1, 2, 3, 2, 1, 3, 2, 4, 5, 4]
 Help Variable Explorer Plots Files
 Console 19/A X
 Python 3.10.9 | packaged by Anaconda, Inc. | (main, Mar 1 2023, 18:18:15) [MSC v. 1916 64 bit (AMD64)]
 Type "copyright", "credits" or "license" for more information.
 IPython 8.10.0 -- An enhanced Interactive Python.
 In [1]: runcell(0, 'C:/Users/mahen/Python coding questions.py')
 given array is: [1, 2, 3, 2, 1, 3, 2, 4, 5, 4]
 array without duplicates is: [1, 2, 3, 4, 5]
 array without duplicates in reverse order is:[5, 4, 3, 2, 1]
 In [2]:

return array without duplicates and reverse order, image designed by author [Kamireddy Mahendra](#)

Q6) Write a program to find the second large number in a list.

```

def find_second_largest_num(arr):
    if len(arr) < 2:
        return "List should have at least two elements"

    unique_arr = list(set(arr)) # Remove duplicates
    unique_arr.sort(reverse=True) # time complexity is O(nlogn)

    if len(unique_arr)>= 2:
        return unique_arr[1]
    else:
        "No second-largest number"

array = [2,5,9,3,8,96,25,36,45]
res = find_second_largest_num(array)
print(f"The second-largest number is: {res}")

```

We can Optimize the above code with time complexity from $O(n \log n)$ to $O(n)$.

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in

```

largest = float('-inf')
second_largest = float('-inf')
for i in L:
    if i > largest:
        second_largest = largest
        largest=i
    elif i > second_largest and i != largest:
        second_largest = i
return second_largest
arr = [5,36,25,89,45,17,22,35]
result_optimized = second_largest_number(arr)
print(f" The second-largest number is: {result_optimized}")

```

```

78 def delete_duplicates(arr):
79     res = []
80     for i in arr:
81         if i not in res:
82             res.append(i)
83     return res
84
85 # Test the function
86 array = [5,36,25,89,45,17,22,35]
87 res = delete_duplicates(array)
88 reverse_res=sorted(delete_duplicates(array), reverse=True)
89 print(f'given array is: {array}')
90 print(f'array without duplicates is: {res}')
91 print(f'array without duplicates in reverse order is:{reverse_res}'')
92
93
94 def second_largest_number(L):
95     if len(L) < 2:
96         return 'List should have more than one number'
97     largest = float('-inf')
98     second_largest = float('-inf')
99     for i in L:
100         if i > largest:
101             second_largest = largest
102             largest=i
103         elif i > second_largest and i != largest:
104             second_largest = i
105     return second_largest
106 arr = [5,36,25,89,45,17,22,35]
107 result_optimized = second_largest_number(arr)
108 print(f" The second-Largest number is: {result_optimized}")
109

```

The Second Largest number with Optimization, image designed by author [Kamireddy Mahendra](#)

Q7) Write a program to find the second-largest negative number in a list.

```

def second_largest_negative(arr):
    largest_neg = float('-inf')
    second_largest_neg = float('-inf')

```

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in

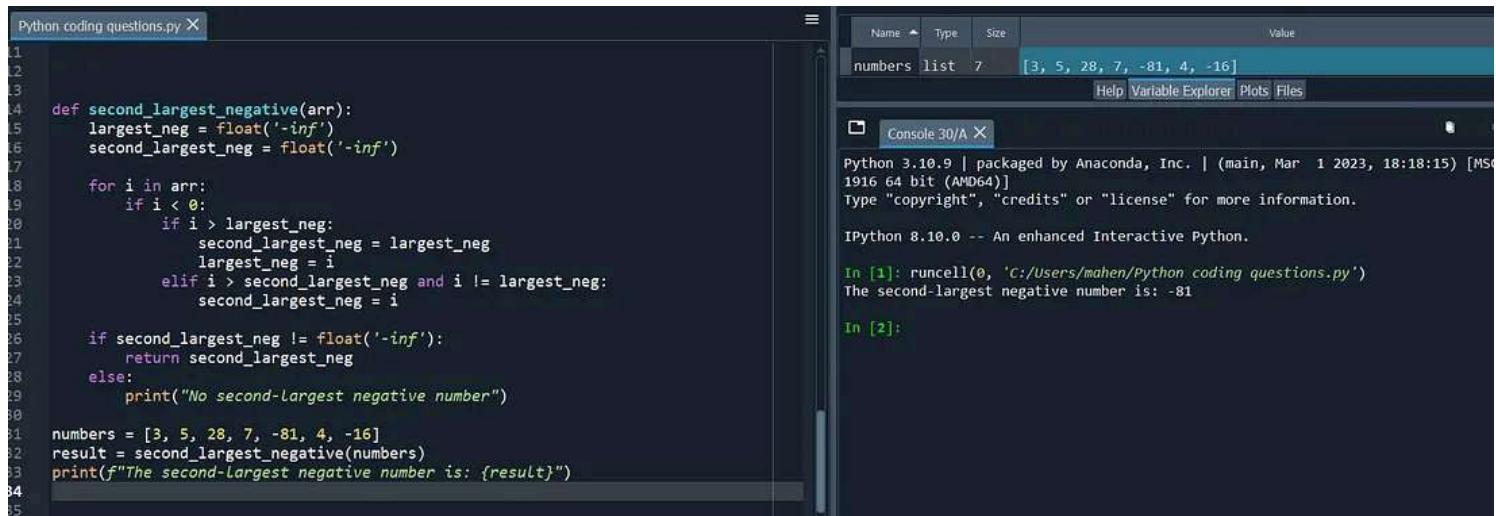
```

        second_largest_neg = largest_neg
        largest_neg = i
    elif i > second_largest_neg and i != largest_neg:
        second_largest_neg = i

if second_largest_neg != float('-inf'):
    return second_largest_neg
else:
    "No second-largest negative number"

numbers = [3, -5, -28, 7, -81, 4, -16]
result = second_largest_negative(numbers)
print(f"The second-largest negative number is: {result}")

```



The screenshot shows a Jupyter Notebook interface with the following details:

- Code Editor:** The code is written in Python, defining a function `second_largest_negative` that takes a list of numbers and returns the second largest negative number. It handles edge cases where there might be no second largest negative number.
- Variable Explorer:** Shows a variable `numbers` of type `list` with a size of 7, containing the values `[3, 5, 28, 7, -81, 4, -16]`.
- Console:** The output of the code execution shows the result: `The second-largest negative number is: -81`.

second largest negative number, image designed by author [Kamireddy Mahendra](#)

Q8) Write a program to reverse the order of words in a sentence.

```

def reverse_words(text):
    words = text.split()
    reversed_sentence = ' '.join(reversed(words))
    return reversed_sentence

```

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in

Printed: REVERSED SENTENCE. RESULTS.



The screenshot shows a Jupyter Notebook interface. On the left, the code in `Python coding questions.py` is displayed:

```

38
39
40 def reverse_words(text):
41     words = text.split()
42     reversed_sentence = ' '.join(reversed(words))
43     return reversed_sentence
44
45
46 input_sentence = "Hi welcome to the world of python programming"
47 result = reverse_words(input_sentence)
48 print(f"Given sentence: {input_sentence}")
49 print(f"Reversed sentence: {result}")
50
51
52
53
54
55
56

```

On the right, the `Variable Explorer` shows the variable `input_sentence` with the value "Hi welcome to the world of python programming". The `Console 28/A` tab shows the execution of the script:

```

Python 3.10.9 | packaged by Anaconda, Inc. | (main, Mar 1 2023, 18:18:15) [MSC v. 1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.10.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/mahen/Python coding questions.py', wdir='C:/Users/mahen')
Given sentence: Hi welcome to the world of python programming
Reversed sentence: programming python of world the to welcome Hi

In [2]:

```

Reverse words in a sentence or text, image designed by author [Kamireddy Mahendra](#)

Q9) Write a program to count the lower and upper case characters in any text or sentence.

```

def count_case_characters(sentence):
    upper = 0
    lower = 0

    for char in sentence:
        if char.isupper():
            upper += 1
        elif char.islower():
            lower += 1
    return upper, lower

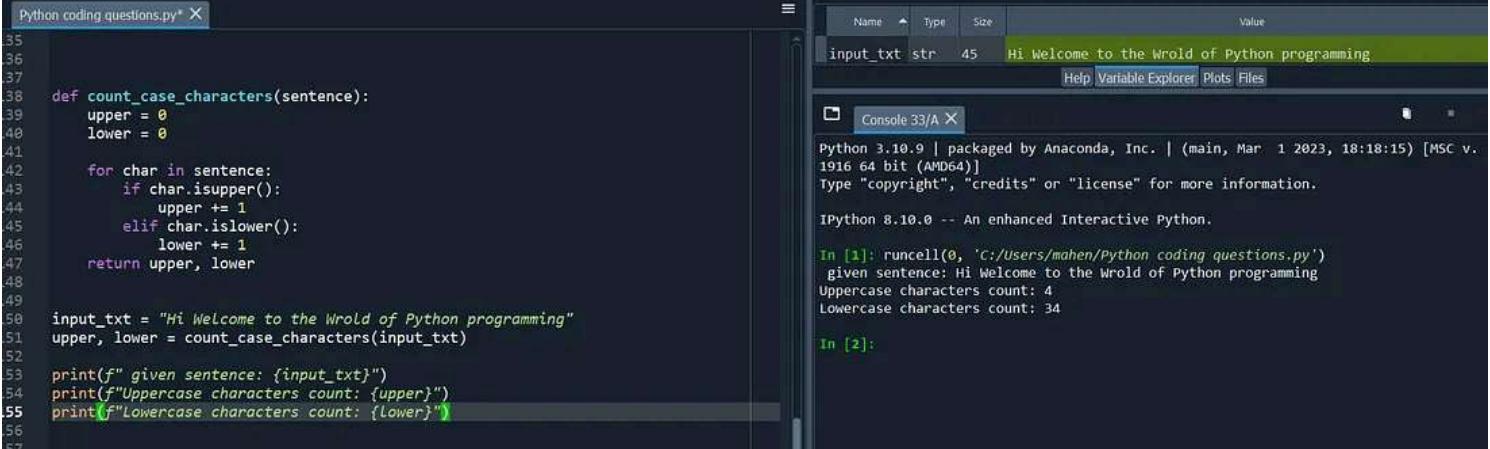
input_txt = "Hi Welcome to the Wrold of Python programming"
upper, lower = count_case_characters(input_txt)

print(f" given sentence: {input_txt}")

```

```
print(f"Uppercase characters count: {upper}")
```

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



```

35
36
37
38 def count_case_characters(sentence):
39     upper = 0
40     lower = 0
41
42     for char in sentence:
43         if char.isupper():
44             upper += 1
45         elif char.islower():
46             lower += 1
47     return upper, lower
48
49
50 input_txt = "Hi Welcome to the Wrold of Python programming"
51 upper, lower = count_case_characters(input_txt)
52
53 print(f" given sentence: {input_txt}")
54 print(f"Uppercase characters count: {upper}")
55 print(f"Lowercase characters count: {lower}")
56

```

Count of upper and lower characters, image designed by author [Kamireddy Mahendra](#)

I hope this article will serve as a solid foundation for preparing for basic-level data engineering coding rounds. In the next phase, I'll provide material covering slightly more complex questions to further enhance your readiness.

Bring your hands together to create a resounding clap, fostering support and encouragement for me to share even more valuable content in the future.

Here is the continuation of this article and added a few more coding problems. Click on [Python Coding Questions for Data Engineer Interview Part II \(Easy Level\)](#) to explore a few more problems.

[**Learning “Web Scraping” Tech Stack Can Make Us a Millionaire!!**](#)

[**Follow me and subscribe**](#) to catch any updates from me instantly.

Thank you :-)

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. [Sign in](#)

Data Analytics/ Data Analyst / Business Analyst / Data Engineer / Data Scientist

Python Coding Questions for a Data Engineer Interview

1. Factorial of a number using recursion
2. Square of largest element in an array
3. Number & its frequency in a list
4. Given number is Prime or not
5. Return a list without Duplicates in descending order
6. Second largest Number in a list
7. Second Largest negative number in a list
8. Reversing the order of words
9. Count upper and lower characters in a text



image designed by author [Kamireddy Mahendra](#)

Academic

Thank you for reading until the end. Before you go:

- Please consider **clapping** and **following** the writer! [clapping](#)
- Follow us [X](#) | [LinkedIn](#) | [YouTube](#) | [Discord](#)
- Visit our other platforms: [In Plain English](#) | [CoFeed](#) | [Venture](#)

[Data Engineer](#)[Data Engineering](#)[Python](#)[Python Programming](#)[Coding](#)

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



Written by Kamireddy Mahendra

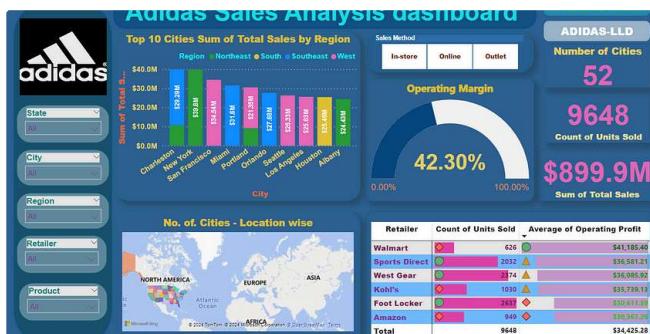
158 Followers · Writer for Stackademic

Follow



Data Analytics Engineer & Content Creator (Upcoming Data Scientist).I write content on Data, Analytics, Engineering & Sciences and Other useful information.

More from Kamireddy Mahendra and Stackademic



Kamireddy Mahendra in Towards AI

A Data Analysis Project—Adidas Sales Report.

Sales Analysis of Adidas products using Power BI Dashboard.

5 min read · Mar 12, 2024



7 min read · Feb 28, 2024



React Forget

An auto-memoizing compiler
 Research project in React Labs
 It is the same familiar React

Wayne Jun in Stackademic

React 19 is coming, what's new?

React 19 will be another milestone release after the introduction of hooks!

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in



Cheat Sheet

 Oliver Foster in Stackademic

What's the Difference Between localhost and 127.0.0.1?

My article is open to everyone; non-member readers can click this link to read the full text.

◆ · 8 min read · Feb 1, 2024

 2.5K  11



The Essential PySpark Cheat Sheet for Data Engineers.

The Essential PySpark Cheat Sheet for Data Engineers.

6 min read · Feb 2, 2024

 107 



[See all from Kamireddy Mahendra](#)

[See all from Stackademic](#)

Recommended from Medium





Kamireddy Mahendra in ILLUMINATION'S MIRROR



Nikita Silaparasetty

**Py
En**

You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. Sign in

Mastering Python For Data Engineer Interview—#Python

4 min read · Jan 31, 2024

105



154



Get ready for your next Python interview with these practice questions!

8 min read · Jan 20, 2024

Lists



Coding & Development

11 stories · 522 saves



Predictive Modeling w/ Python

20 stories · 1034 saves



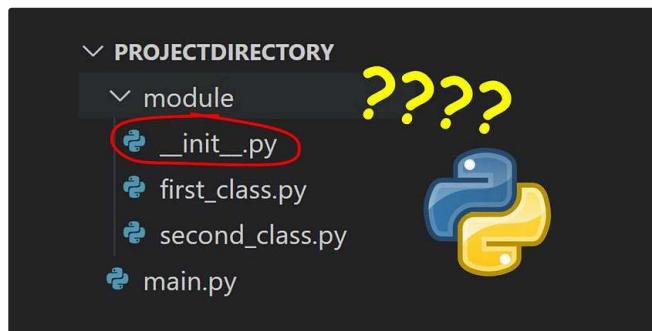
Practical Guides to Machine Learning

10 stories · 1236 saves



Stories to Help You Grow as a Software Developer

19 stories · 930 saves



Sarang Surve

Python Basics: Why use __init__.py?

Hey There! Today, we're diving into the understanding of __init__.py in Python, a tiny...

3 min read · Feb 3, 2024



Artturi Jalli

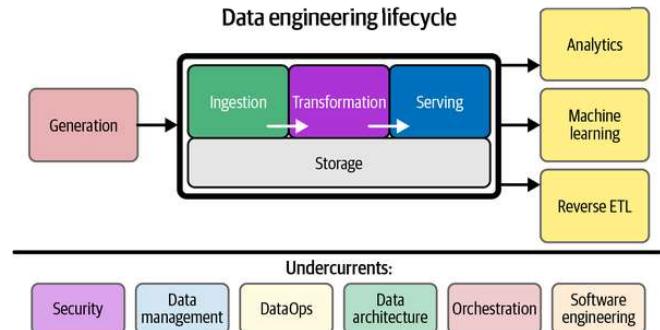
25 Key Python Interview Questions (for 2024)

This is a comprehensive list of Python coding interview questions. Make sure to nail these...

33 min read · Dec 4, 2023



You are signed out. Sign in with your member account (va__@g__.com) to view other member-only stories. [Sign in](#)



Hussein Jundi in Towards Data Science



Mudra Patel

Data Engineering: Incremental Data Loading Strategies

Outlining strategies and solution architectures to incrementally load data fro...

◆ 10 min read · Mar 18, 2024

516

3



5 min read · Feb 12, 2024

72

1



[See more recommendations](#)