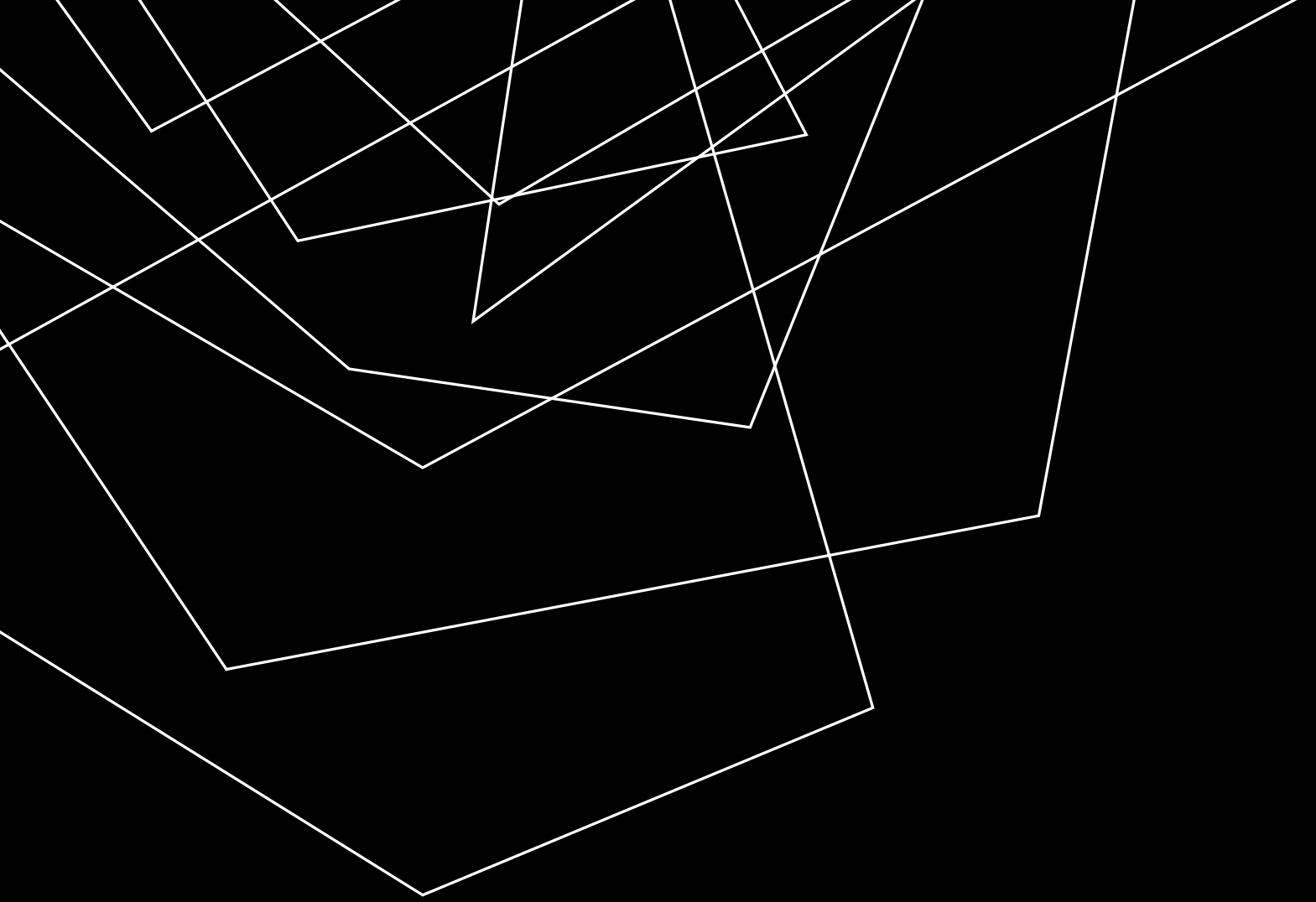
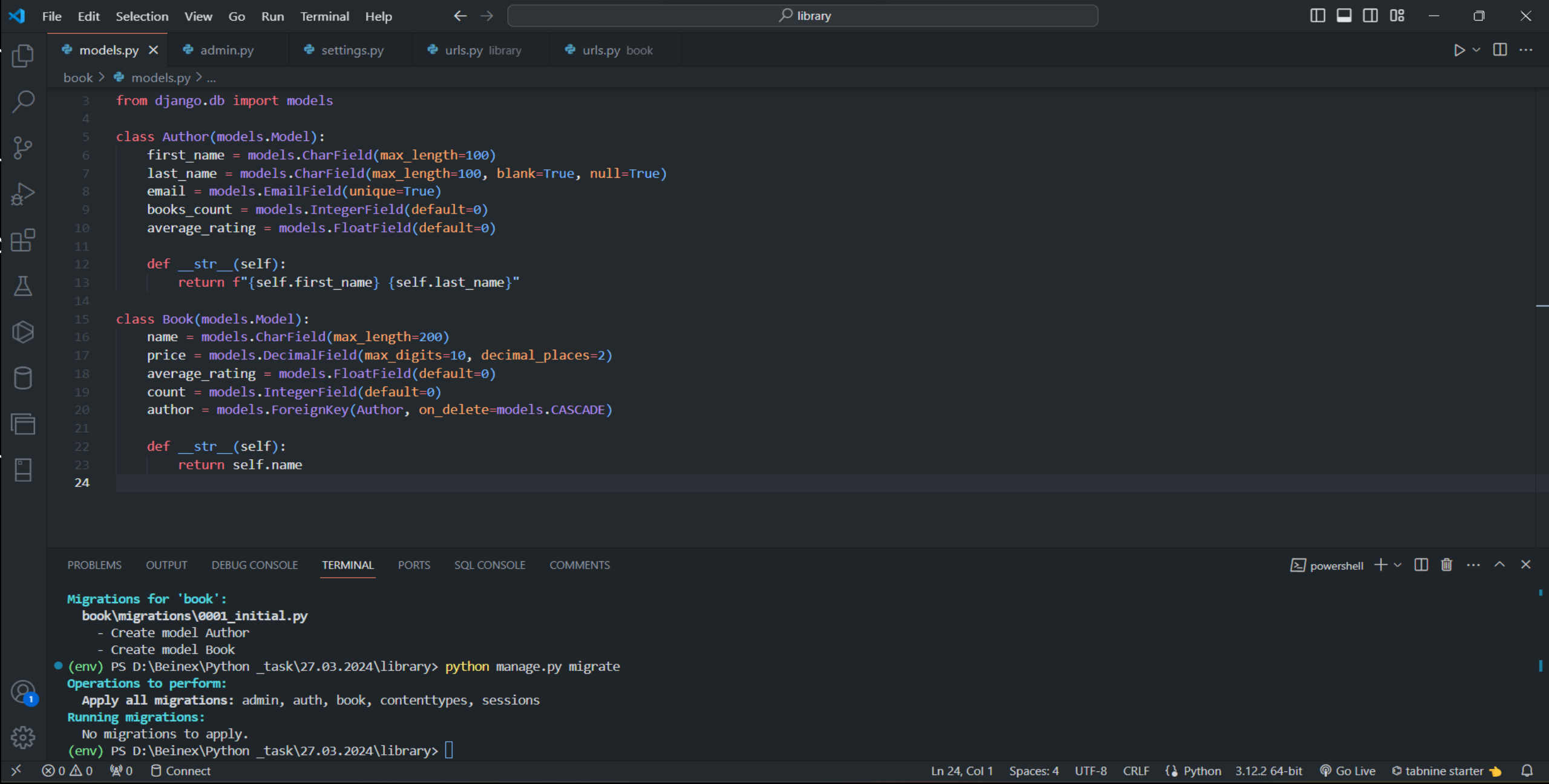


DJANGO ORM



INSTALL DJANGO-EXTENSIONS

```
C:\Windows\System32\cmd.e

(env) D:\Beinex\Python _task\27.03.2024\library>python manage.py shell_plus
# Shell Plus Model Imports
from book.models import Author, Book
from django.contrib.admin.models import LogEntry
from django.contrib.auth.models import Group, Permission, User
from django.contrib.contenttypes.models import ContentType
from django.contrib.sessions.models import Session
# Shell Plus Django Imports
from django.core.cache import cache
from django.conf import settings
from django.contrib.auth import get_user_model
from django.db import transaction
from django.db.models import Avg, Case, Count, F, Max, Min, Prefetch, Q, Sum, When
from django.utils import timezone
from django.urls import reverse
from django.db.models import Exists, OuterRef, Subquery
Python 3.12.2 (tags/v3.12.2:6abddd9, Feb  6 2024, 21:26:36) [MSC v.1937 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>>
>>>
>>>
>>> from book.models import Author, Book
>>>
>>> Author.objects.bulk_create([
...     Author(first_name='Albin', last_name='AK', email='albin@example.com', books_count=10 ,average_rating=7),
...     Author(first_name='Rahul', last_name='', email='rahul@example.com', books_count=50 ,average_rating=1.2),
...     Author(first_name='Arun', last_name='Kumar', email='arun@example.com', books_count=25 ,average_rating=9),
...     Author(first_name='Akhila', last_name='Jhon', email='akhila@example.com',books_count=3 ,average_rating=5.3),
...     Author(first_name='Madavi', last_name='Kutty', email='madavi@example.com',books_count=8 ,average_rating=4.1),
... ])
[<Author: Albin>, <Author: Rahul>, <Author: Arun>, <Author: Akhila>, <Author: Madavi>]
>>>
>>>
>>>
>>> Book.objects.bulk_create([
...     Book(name='Book 1', price=100, average_rating=4.5, count=50, author=Author.objects.get(first_name='Albin')),
...     Book(name='Book 2', price=150, average_rating=3.8, count=30, author=Author.objects.get(first_name='Albin')),
...     Book(name='Book 3', price=200, average_rating=4.2, count=45, author=Author.objects.get(first_name='Rahul')),
...     Book(name='Book 4', price=250, average_rating=4.7, count=60, author=Author.objects.get(first_name='Rahul')),
...     Book(name='Book 5', price=300, average_rating=3.9, count=25, author=Author.objects.get(first_name='Arun')),
...     Book(name='Book 6', price=350, average_rating=4.0, count=40, author=Author.objects.get(first_name='Arun')),
...     Book(name='Book 7', price=400, average_rating=4.8, count=55, author=Author.objects.get(first_name='Akhila')),
...     Book(name='Book 8', price=450, average_rating=3.5, count=20, author=Author.objects.get(first_name='Akhila')),
...     Book(name='Book 9', price=500, average_rating=4.6, count=70, author=Author.objects.get(first_name='Madavi')),
...     Book(name='Book 10', price=550, average_rating=4.3, count=0, author=Author.objects.get(first_name='Madavi')),
... ])
[<Book: Book 1>, <Book: Book 2>, <Book: Book 3>, <Book: Book 4>, <Book: Book 5>, <Book: Book 6>, <Book: Book 7>, <Book: Book 8>, <Book: Book 9>, <Book: Book 10>]
>>>
```

CREATE 5 AUTHORS , CREATE 10 BOOKS

```
>>> from book.models import Author, Book
>>> all_authors = Author.objects.all()
>>> print("Authors:")
Authors:
>>> for author in all_authors:
...     print(author)
...
Albin AK
Rahul
Arun Kumar
Akhila Jhon
Madavi Kutty
>>>
>>>
>>> all_books = Book.objects.all()
>>> print("\nBooks:")

Books:
>>> for book in all_books:
...     print(book)
...
Book 1
Book 2
Book 3
Book 4
Book 5
Book 6
Book 7
Book 8
Book 9
Book 10
```

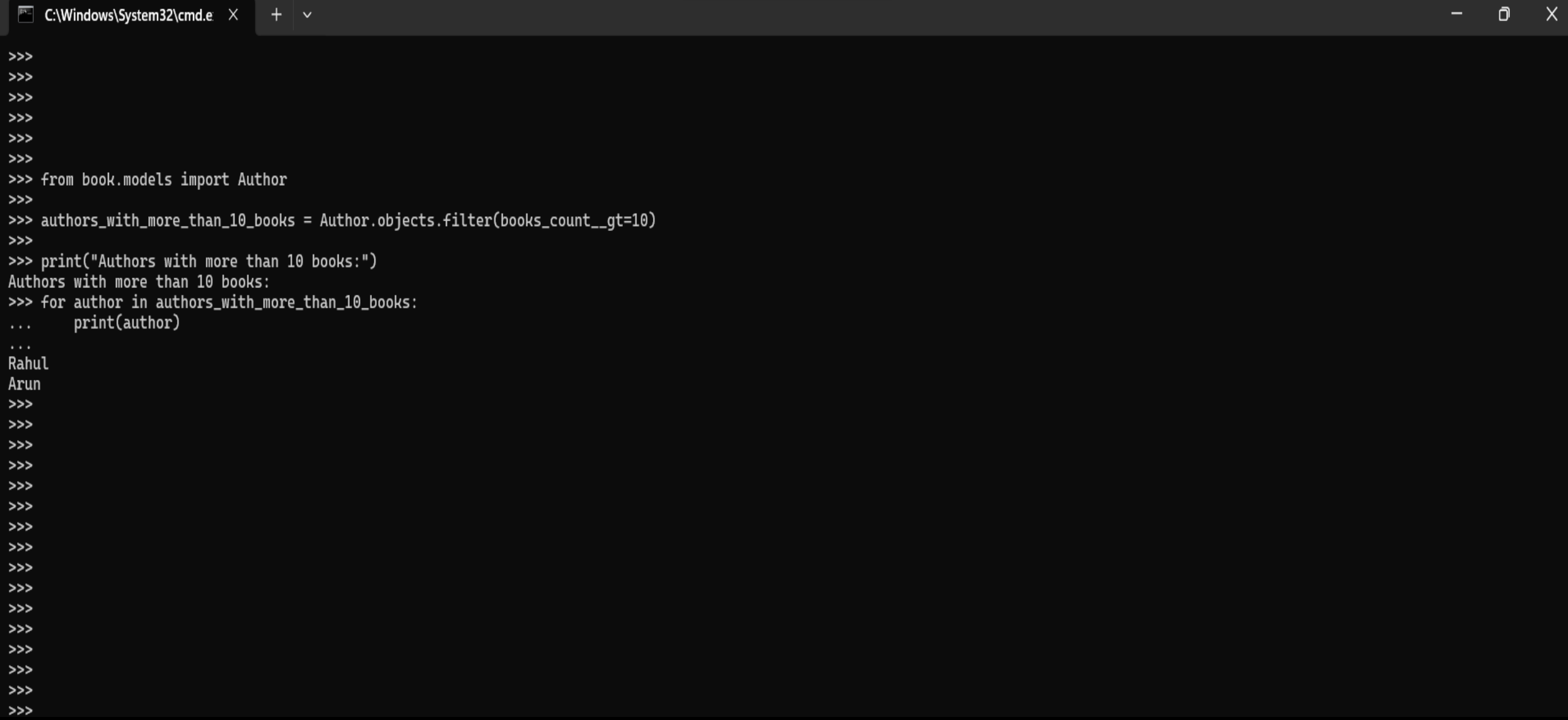
SHOW ALL THE OBJECTS IN BOOKS AND AUTHORS MODEL

```
>>>
>>> from book.models import Author
>>>
>>> filtered_authors = Author.objects.filter(first_name__icontains='A')
>>> print("Filtered Authors:")
Filtered Authors:
>>> for author in filtered_authors:
...     print(author)
...
Albin AK
Rahul
Arun Kumar
Akhila Jhon
Madavi Kutty
>>>
>>>
>>>
>>>
```

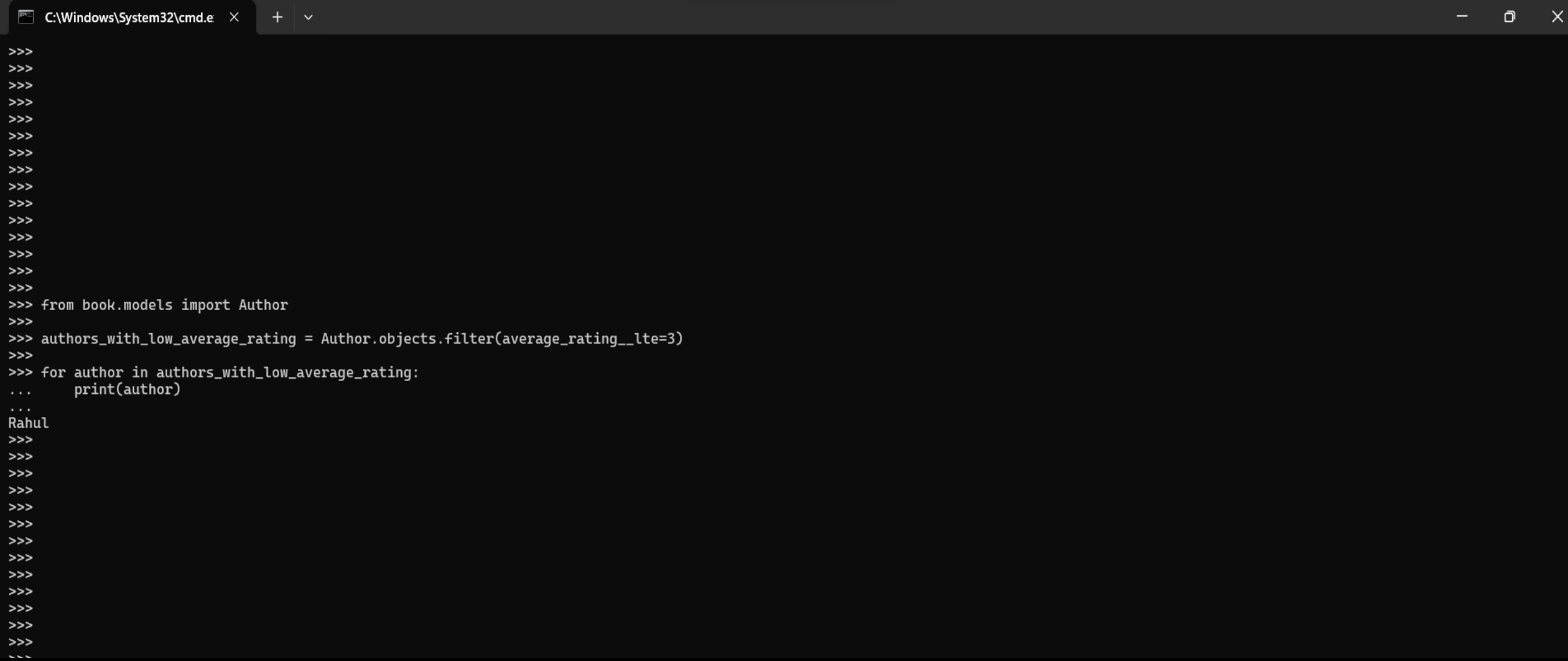
FILTER AND SHOW SOME AUTHORS USING FIRST NAME FIELD

```
>>>
>>>
>>> from book.models import Author
>>>
>>> authors_without_last_name = Author.objects.filter(last_name__isnull=True)
>>> print("Authors without a last name:")
Authors without a last name:
>>> for author in authors_without_last_name:
...     print(author)
...
Rahul
>>>
>>>
>>>
>>>
```

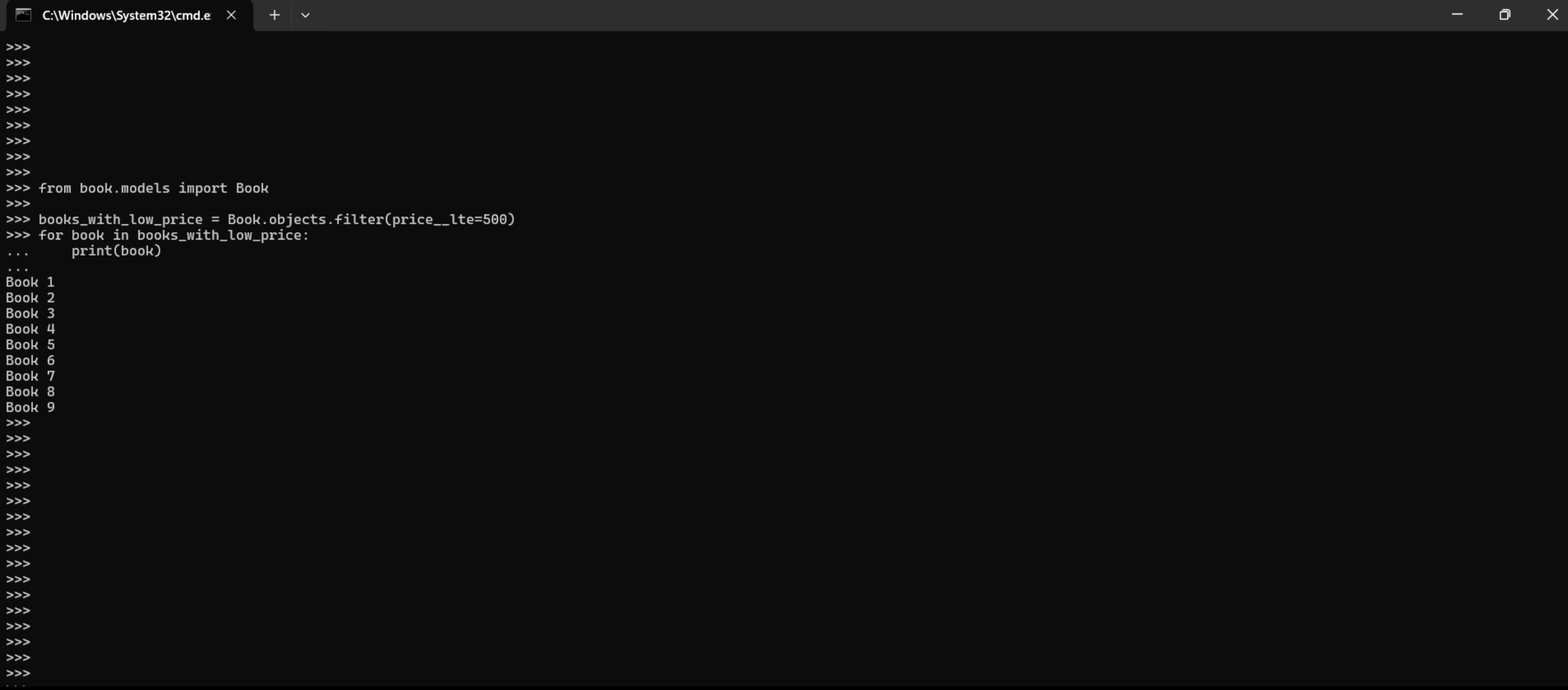
FILTER AND SHOW SOME AUTHORS THAT DOES NOT HAVE A LAST_NAME



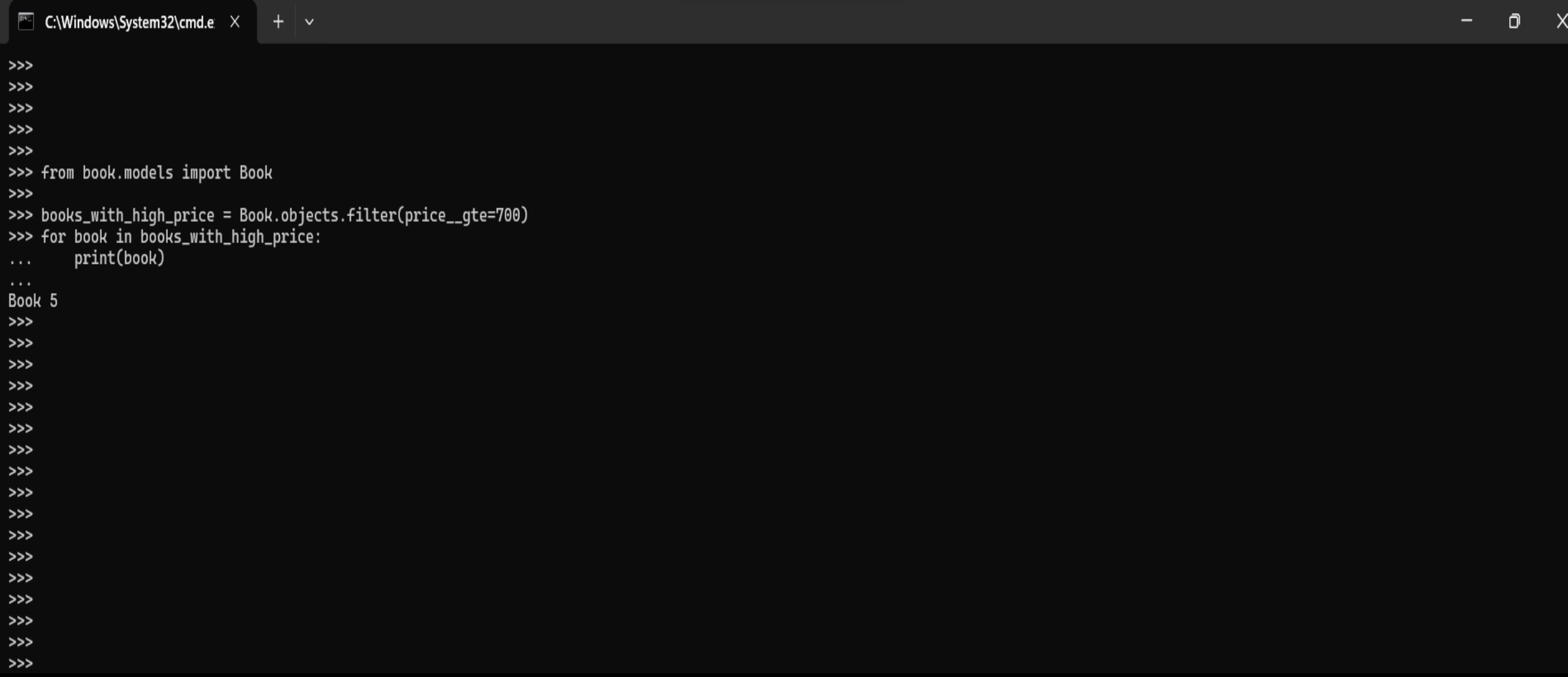
FILTER AND SHOW SOME AUTHORS THAT HAVE BOOKS COUNT GREATER THAN 10



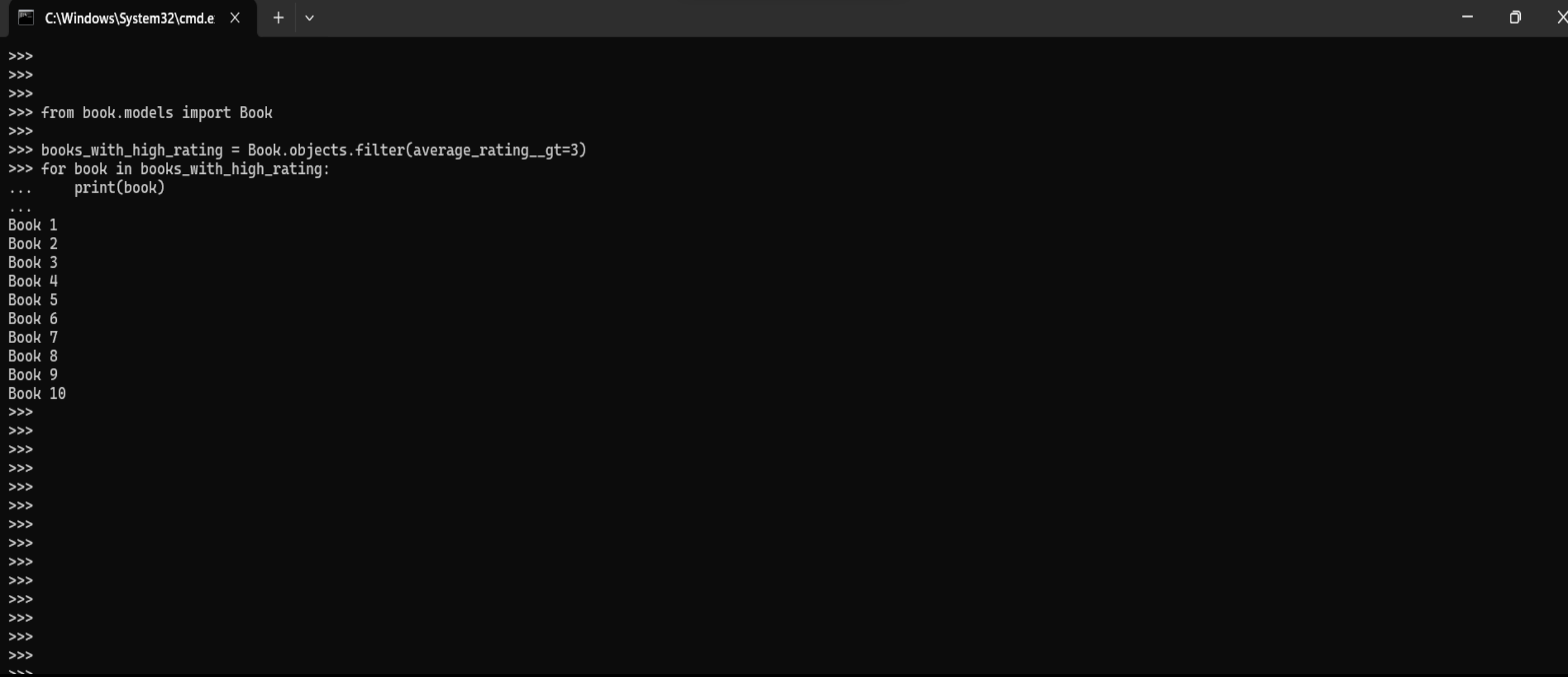
FILTER AND SHOW SOME AUTHORS THAT HAVE AN AVERAGE RATING LESS THAN OR EQUAL TO 3



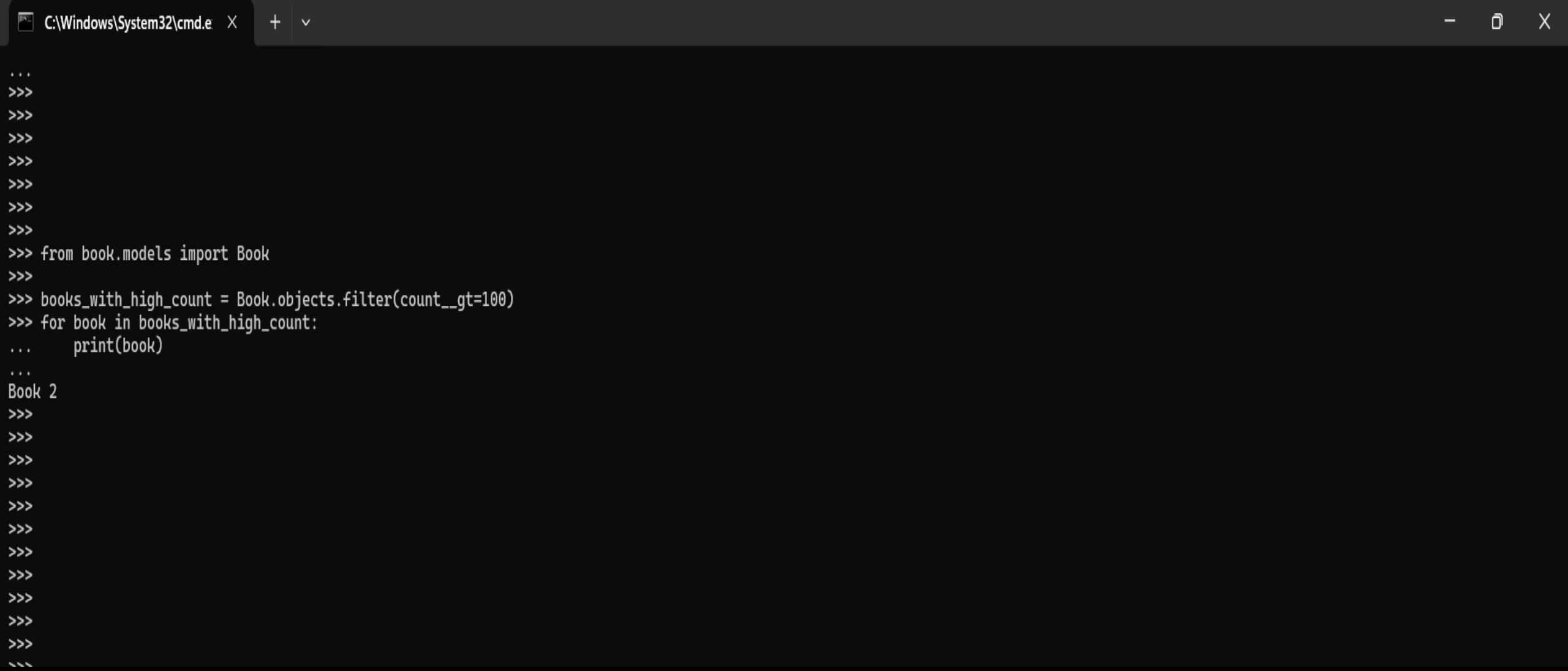
FILTER AND SHOW SOME BOOKS THAT HAVE PRICE LESS THAN OR EQUAL TO 500



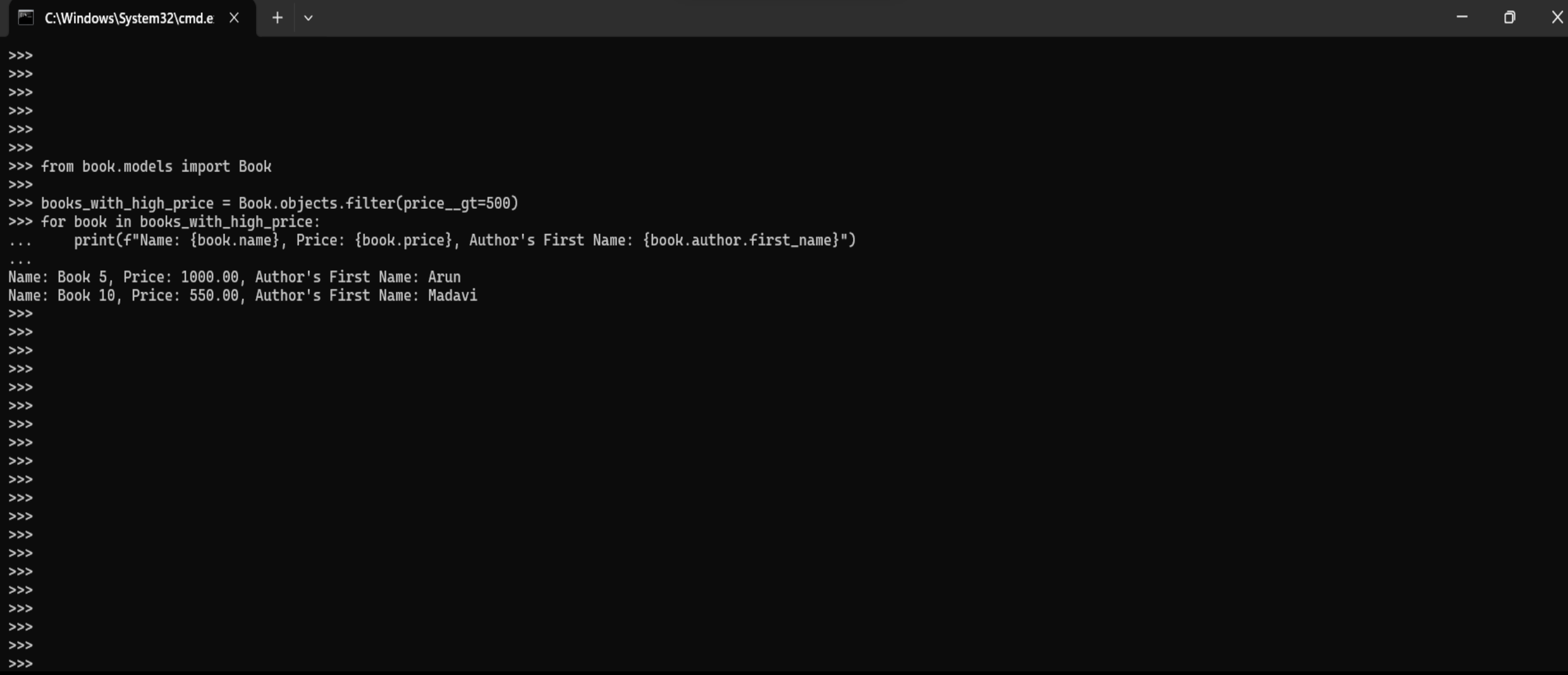
FILTER AND SHOW SOME BOOKS THAT HAVE PRICE GREATER THAN OR EQUAL TO 700



FILTER AND SHOW SOME BOOKS THAT HAS AN AVERAGE RATING GREATER THAN 3



FILTER AND SHOW SOME BOOKS THAT HAS COUNT GREATER THAN 100

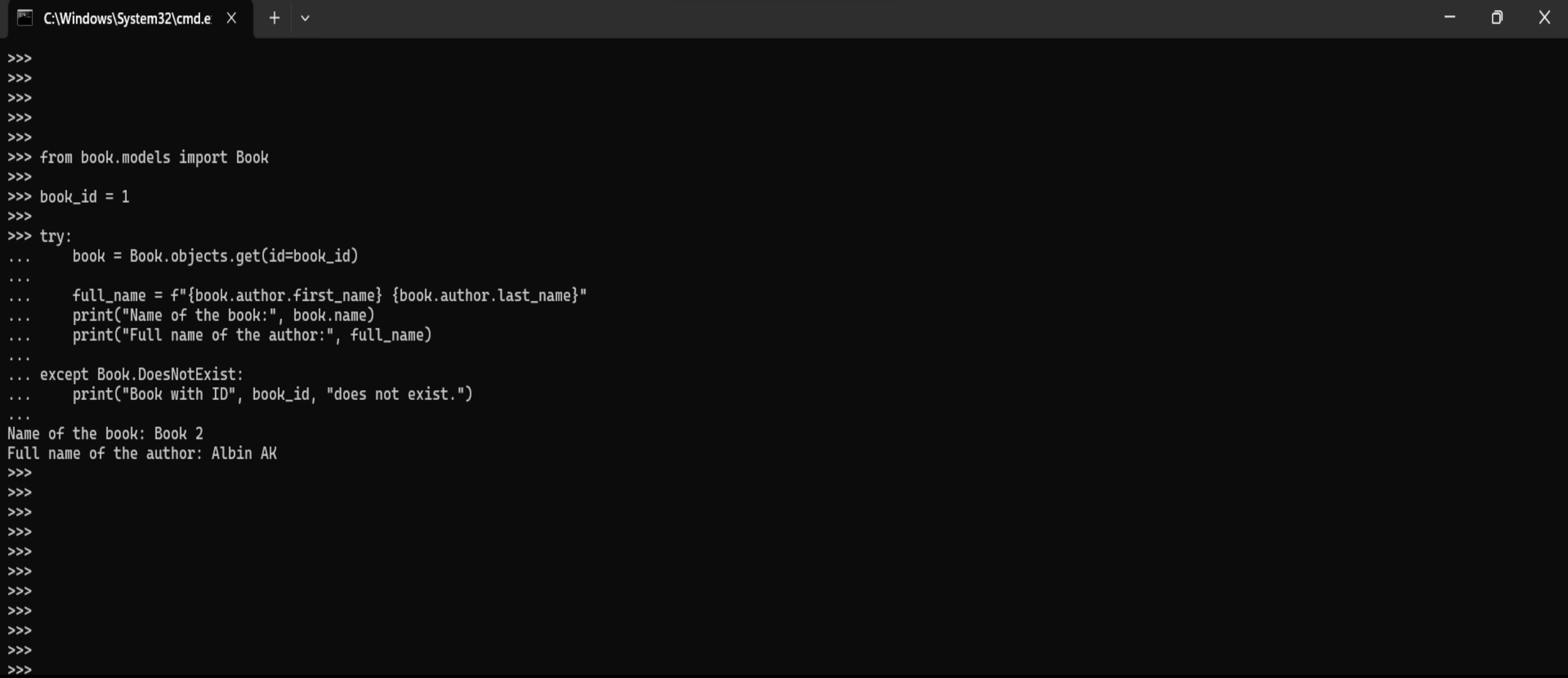


FILTER SOME BOOKS THAT HAS PRICE GREATER THAN 500 AND SHOW THE NAMES OF ALL ONE BY ONE ALONG WITH ITS PRICE AND AUTHOR'S FIRST NAME

C:\Windows\System32\cmd.e

```
>>> from book.models import Book
>>>
>>> last_book = Book.objects.last()
>>>
>>> if last_book:
...     print("Full name of the author of the last book:", last_book.author.first_name, last_book.author.last_name)
... else:
...     print("No books found.")
...
Full name of the author of the last book: Madavi Kutty
>>>
```

FETCH THE LAST BOOK AND SHOW ITS AUTHOR'S FULL NAME

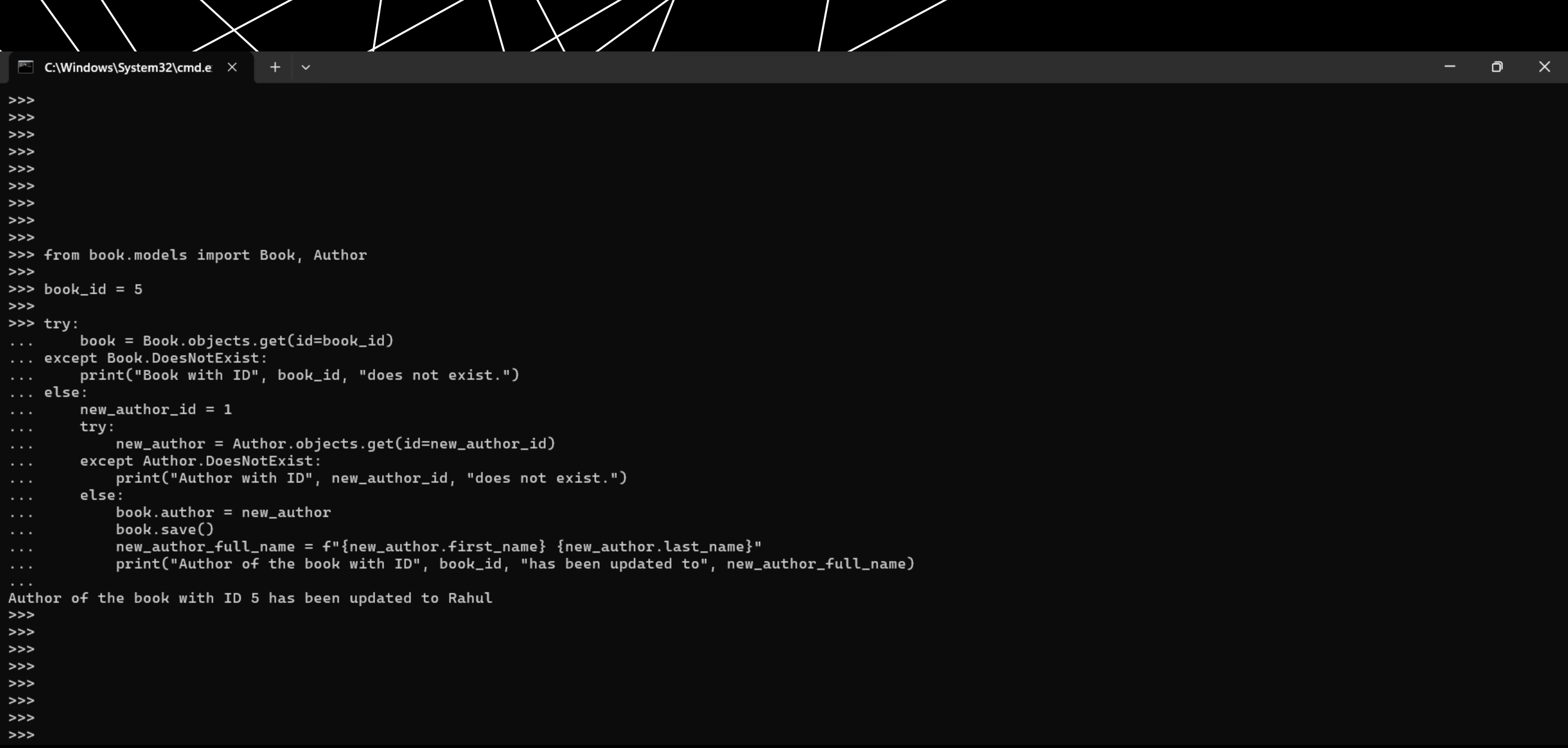


FETCH A BOOK USING ITS ID, SHOW ITS NAME AND AUTHOR'S FULL NAME

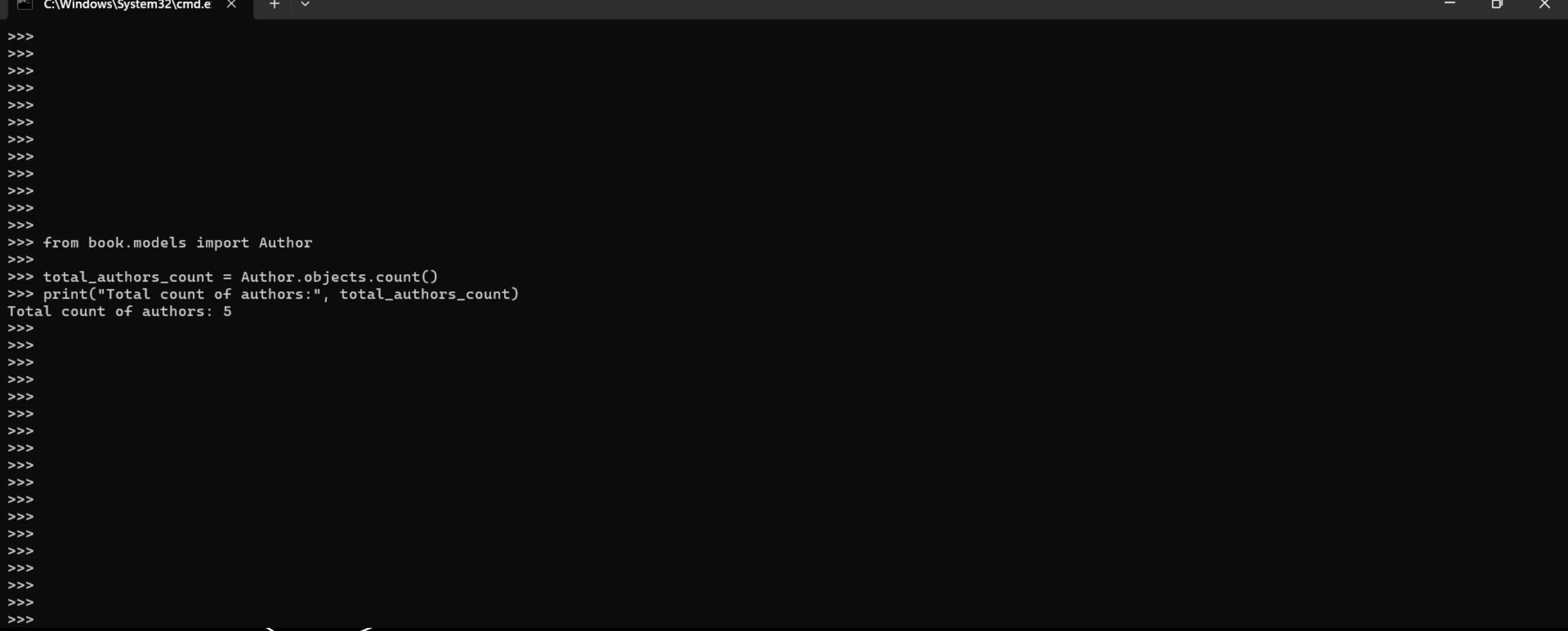
C:\Windows\System32\cmd.e

```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> from book.models import Book, Author
>>>
>>> total_books_count = Book.objects.count()
>>> total_authors_count = Author.objects.count()
>>>
>>> print("Total count of books:", total_books_count)
Total count of books: 10
>>> print("Total count of authors:", total_authors_count)
Total count of authors: 5
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

SHOW THE TOTAL COUNT OF BOOKS AND AUTHORS



FETCH A BOOK BY ITS ID AND UPDATE THE AUTHOR AND SAVE



SHOW THE TOTAL COUNT OF AUTHORS

```
>>> from book.models import Author
>>> authors_with_high_rating_count = Author.objects.filter(average_rating__gte=3).count()
>>> print("Count of authors with an average rating greater than or equal to 3:", authors_with_high_rating_count)
Count of authors with an average rating greater than or equal to 3: 4
```

SHOW THE COUNT OF AUTHORS THAT HAS AN AVERAGE RATING GREATER THAN OR EQUAL TO 3

C:\Windows\System32\cmd.e

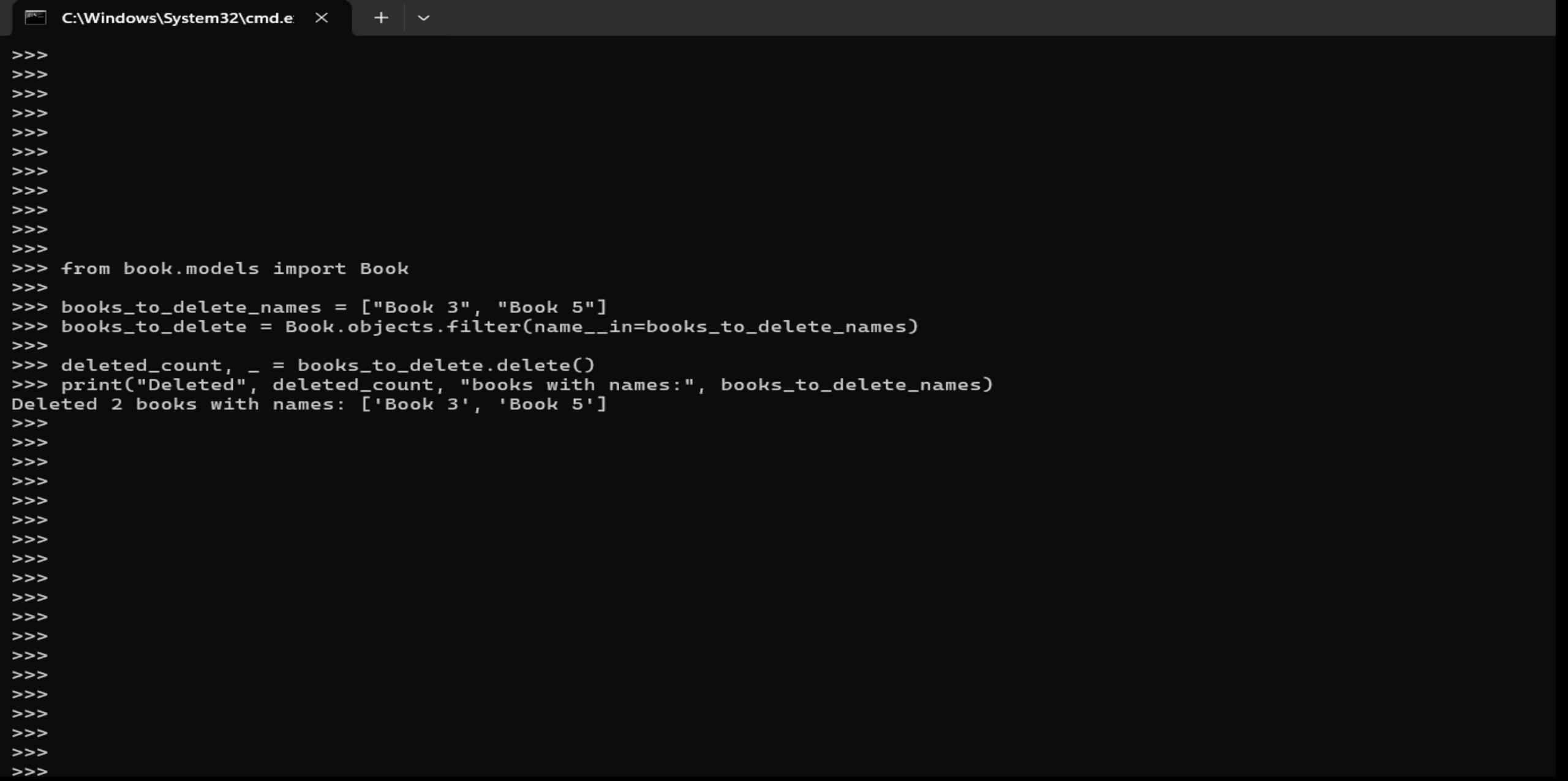
```
>>>
>>>
>>>
>>>
>>> from book.models import Author
>>>
>>> author_email = "albin@example.com"
>>> author_exists = Author.objects.filter(email=author_email).exists()
>>>
>>> if author_exists:
...     print("Author with email", author_email, "exists.")
... else:
...     print("Author with email", author_email, "does not exist.")
...
Author with email albin@example.com exists.
>>>
>>>
>>>
>>>
>>>
>>>
>>> from book.models import Author
>>>
>>> author_email = "ram@example.com"
>>> author_exists = Author.objects.filter(email=author_email).exists()
>>>
>>> if author_exists:
...     print("Author with email", author_email, "exists.")
... else:
...     print("Author with email", author_email, "does not exist.")
...
Author with email ram@example.com does not exist.
>>>
```

CHECK WHETHER AN AUTHOR EXISTS IN THE TABLE USING EMAIL FIELD

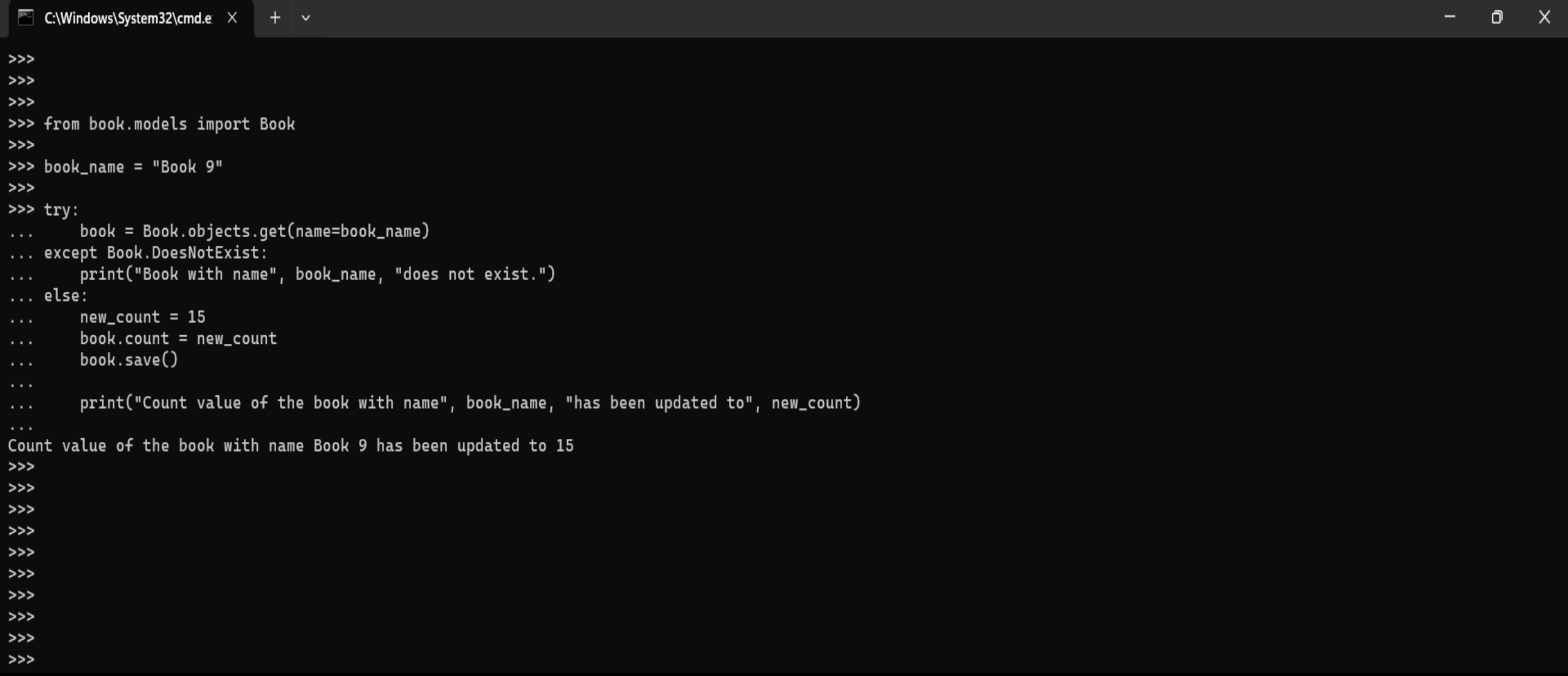
CHECK WHETHER A BOOK EXISTS IN THE TABLE WITH ITS NAME

```
>>>
>>>
>>>
>>>
>>> from book.models import Book
>>>
>>> books_ordered_by_name = Book.objects.all().order_by('name')
>>>
>>> print("Books ordered by name:")
Books ordered by name:
>>> for book in books_ordered_by_name:
...     print(book.name)
...
Book 1
Book 10
Book 2
Book 3
Book 4
Book 5
Book 6
Book 7
Book 8
Book 9
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

SHOW ALL THE BOOKS IN THE ORDER OF ITS NAME

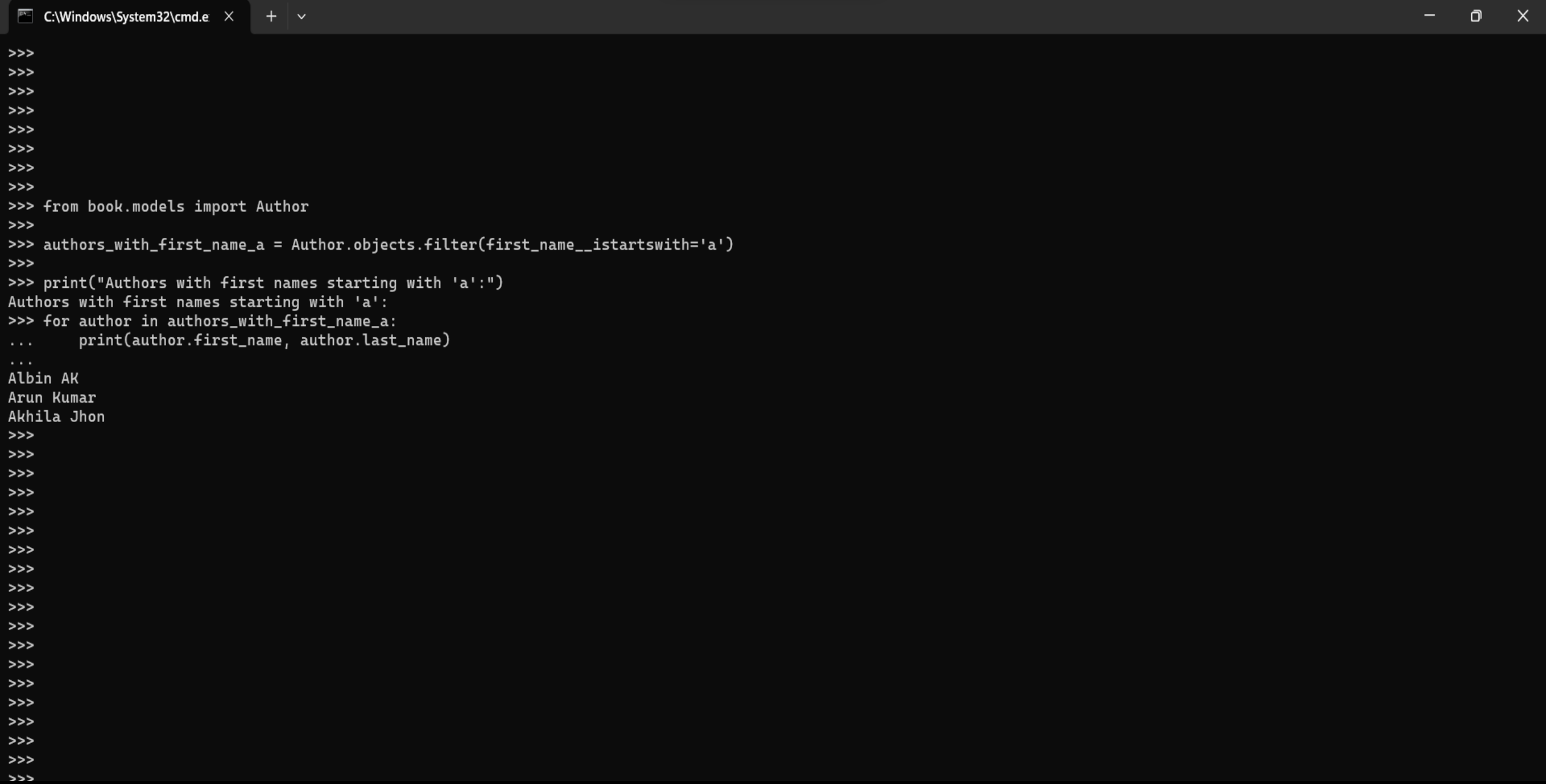


FETCH THE FIRST BOOK AND SHOW ITS NAME

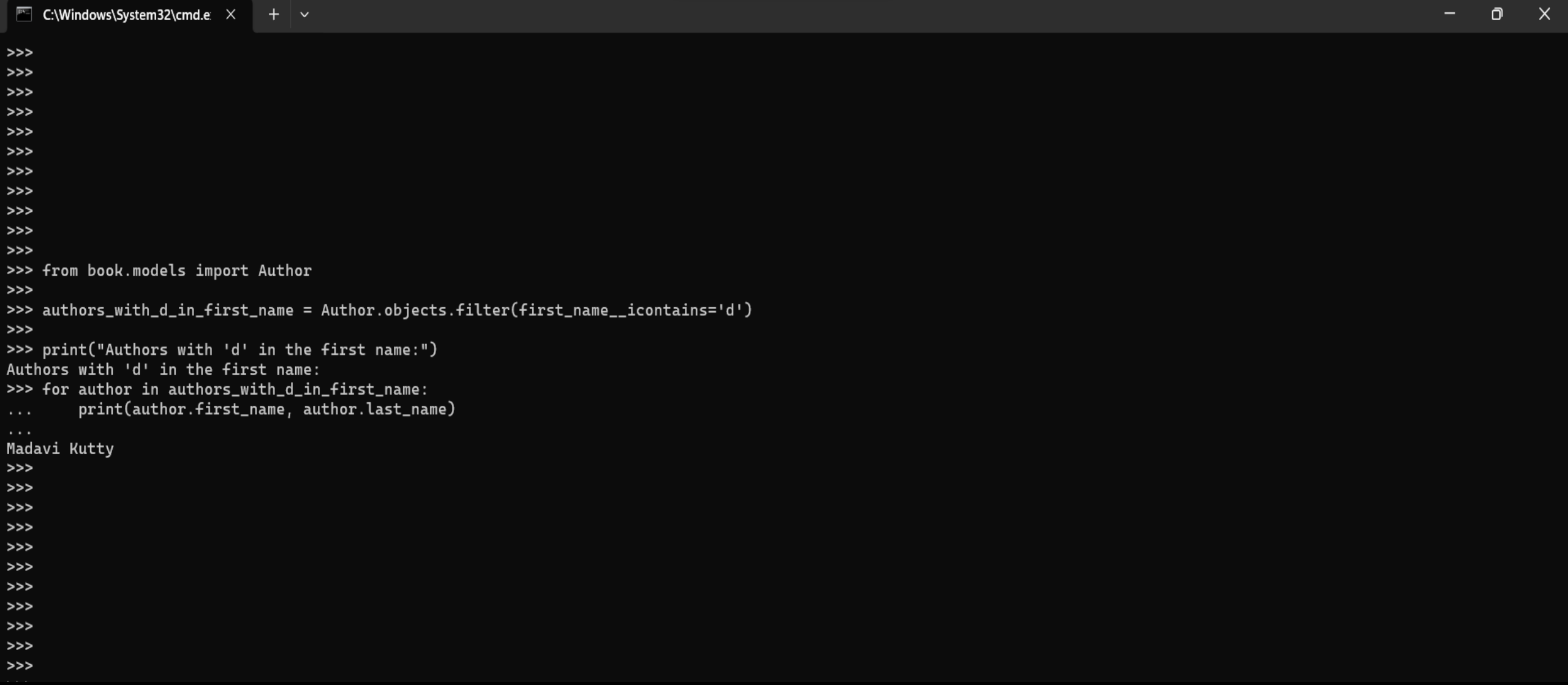


UPDATE THE BOOKS COUNT VALUE AFTER FETCHING A BOOK USING ITS NAME

FILTER AND SHOW THE NAME OF BOOKS THAT HAS BOOKS COUNT LESS THAN OR EQUAL TO 20



SHOW ALL THE AUTHORS THAT HAS FIRST NAME STARTING WITH "A"



SHOW ALL THE AUTHORS THAT CONTAINS "D" IN THE FIRST NAME

C:\Windows\System32\cmd.e

```
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>> from book.models import Book
>>>
>>> books_to_delete = Book.objects.filter(count=0)
>>>
>>> deleted_count, _ = books_to_delete.delete()
>>>
>>> print("Deleted", deleted_count, "books with count equal to 0.")
Deleted 1 books with count equal to 0.
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

DELETE THE BOOKS THAT HAS BOOK COUNT 0