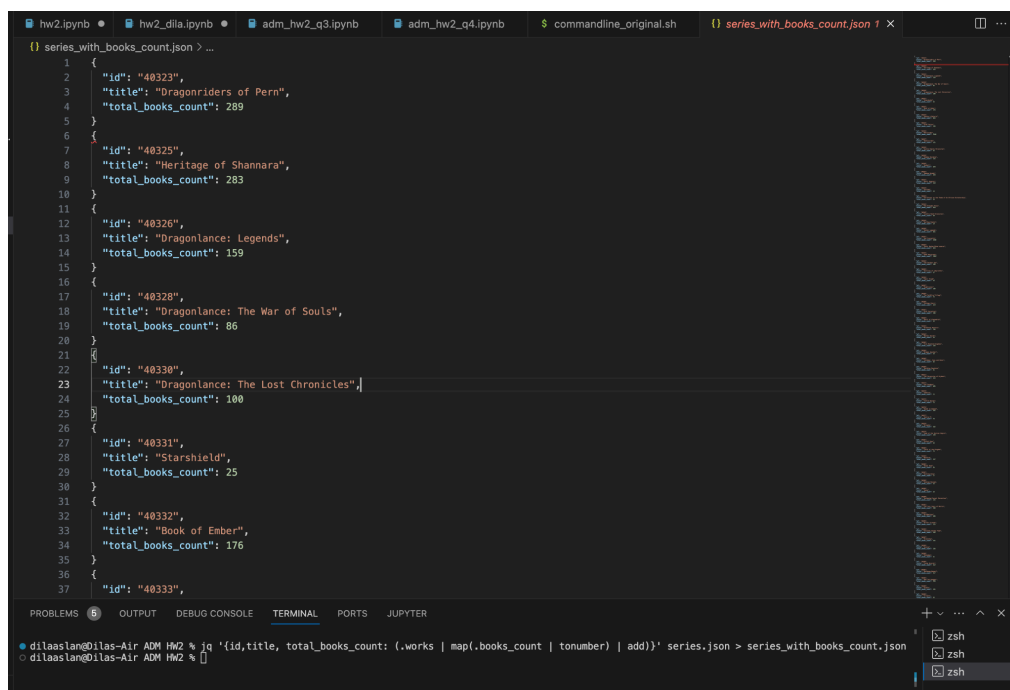


Command Line Question

JQ (<https://jqlang.github.io/jq/>) is used to read json data from the command line and perform operations on it.

1. First, with the following code, we access the 'books_count' data in the 'works' dictionary and convert it into a numeric value, allowing us to sum the counts of books. After the code runs, a file named 'series_with_books_count.json' will be created.

```
jq '{id,title,total_books_count: (.works | map(.books_count | tonumber) | add)}' series.json > series_with_books_count.json
```

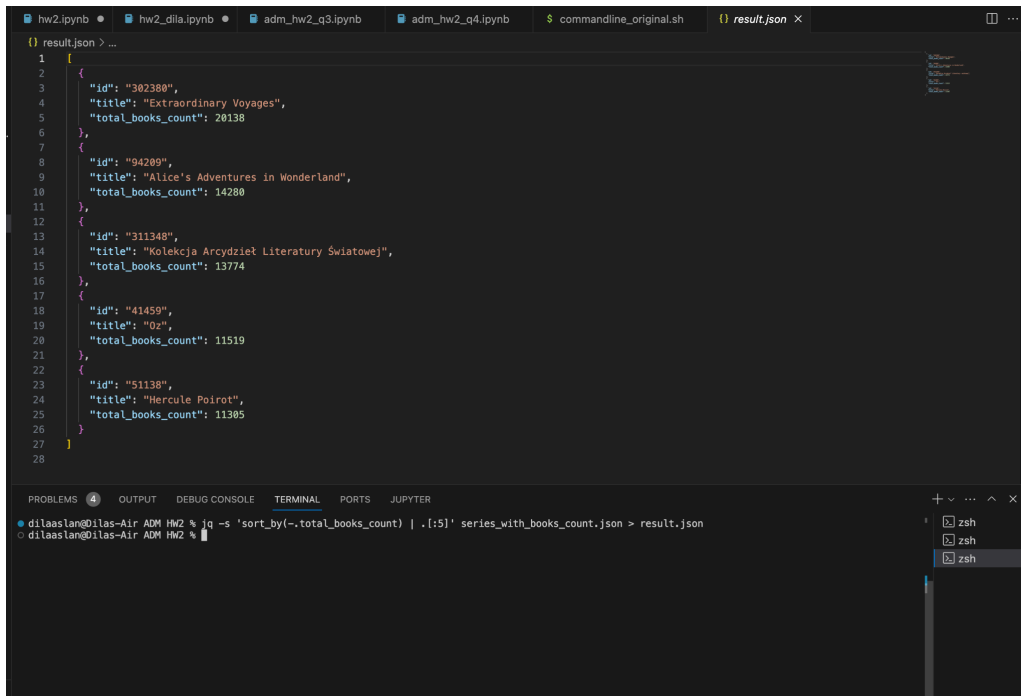


The screenshot shows a JupyterLab interface with a terminal window. The terminal displays the output of a jq command, which has created a file named 'series_with_books_count.json'. The output is a JSON array of objects, each containing 'id', 'title', and 'total_books_count'.

```
{
  "id": "40323",
  "title": "Dragonriders of Pern",
  "total_books_count": 289
},
{
  "id": "40325",
  "title": "Heritage of Shannara",
  "total_books_count": 283
},
{
  "id": "40326",
  "title": "Dragonlance: Legends",
  "total_books_count": 159
},
{
  "id": "40328",
  "title": "Dragonlance: The War of Souls",
  "total_books_count": 86
},
{
  "id": "40330",
  "title": "Dragonlance: The Lost Chronicles",
  "total_books_count": 100
},
{
  "id": "40331",
  "title": "Starshield",
  "total_books_count": 25
},
{
  "id": "40332",
  "title": "Book of Ember",
  "total_books_count": 176
},
{
  "id": "40333",
  "title": "The Hobbit",
  "total_books_count": 95
}
```

2. Secondly, to find the top 5 highest 'total_books_count,' we sort the 'series_with_books_count.json' file and write the first 5 entries into the 'result.json' file.

```
jq -s 'sort_by(-.total_books_count) | .[:5]' series_with_books_count.json > result.json
```



The screenshot shows a JupyterLab interface with a file browser at the top displaying several files: 'hw2.ipynb', 'hw2_dila.ipynb', 'adm_hw2_q3.ipynb', 'adm_hw2_q4.ipynb', 'commandline_original.sh', and 'result.json'. The 'result.json' file is selected and its content is displayed in the main editor area. The JSON content is as follows:

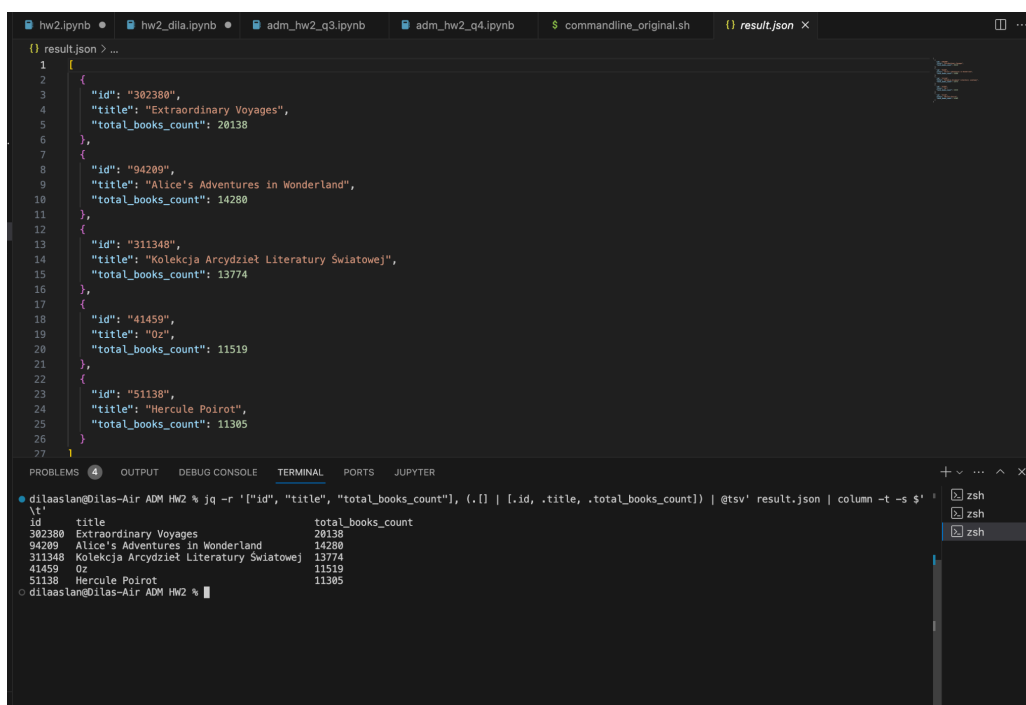
```
{
  "id": "302380",
  "title": "Extraordinary Voyages",
  "total_books_count": 20138
},
{
  "id": "94289",
  "title": "Alice's Adventures in Wonderland",
  "total_books_count": 14280
},
{
  "id": "311348",
  "title": "Kolekcja Arcydzieł Literatury Światowej",
  "total_books_count": 13774
},
{
  "id": "41459",
  "title": "Oz",
  "total_books_count": 11519
},
{
  "id": "51138",
  "title": "Hercule Poirot",
  "total_books_count": 11305
}
```

Below the editor, there is a terminal window with the following command and output:

```
dilaaslan@dilas-Air ADM HW2 % jq -s 'sort_by(-.total_books_count) | .[:5]' series_with_books_count.json > result.json
dilaaslan@dilas-Air ADM HW2 %
```

3. Finally, we print the results from the 'result.json' file to the console, displaying the relevant values under their respective columns.

```
jq -r '["id", "title", "total_books_count"], (.[] | [.id, .title, .total_books_count]) | @tsv' result.json | column -t -s $'\t'
```




The screenshot shows the same JupyterLab interface as before, but the terminal window now displays the output of the jq command. The output is a table with three columns: 'id', 'title', and 'total_books_count'. The data is as follows:

id	title	total_books_count
302380	Extraordinary Voyages	20138
94289	Alice's Adventures in Wonderland	14280
311348	Kolekcja Arcydzieł Literatury Światowej	13774
41459	Oz	11519
51138	Hercule Poirot	11305


ChatGPT Answer

When we posed the same question to ChatGPT, the suggested answer provided by ChatGPT was found to be incorrect, as shown below.

Default (GPT-3.5)

 jq '{id,title,total_books_count:(.works | map(.books_count | tonumber) | add))}' series.json > series_with_books_count.json && jq -s 'sort_by(-.total_books_count) | .[:5]' series_with_books_count.json > result.json && jq -r '["id","title","total_books_count"]' | @tsv result.json | column -t -s \$'\t'

"can you make this more robust solution in max 3 lines of script: (script code)"

 Certainly! You can achieve the same result in a more concise way using the following three lines of script:

```
bash
```

```
jq 'map(.total_books_count = (.works | map(.books_count | tonumber) | add))'
```

Copy code

This script will take your input file "series.json," calculate the total books count, sort by it, select the top 5 series, format the output as a TSV, and write the result to "result.json."

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER
o dilaaslan@dilas-Air ADM HW2 % jq 'map(.total_books_count = (.works | map(.books_count | tonumber) | add)) | sort_by(-.total_books_count) | .[:5] | [{"id" | '
{id, title, total_books_count}]] | @tsv' series.json | column -t -s $'\t' > result.json
```

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS JUPYTER
jq: error (at series.json:5250): Cannot index string with string "works"
jq: error (at series.json:5251): Cannot index string with string "works"
jq: error (at series.json:5252): Cannot index string with string "works"
jq: error (at series.json:5253): Cannot index string with string "works"
jq: error (at series.json:5254): Cannot index string with string "works"
jq: error (at series.json:5255): Cannot index string with string "works"
jq: error (at series.json:5256): Cannot index string with string "works"
jq: error (at series.json:5257): Cannot index string with string "works"
jq: error (at series.json:5258): Cannot index string with string "works"
jq: error (at series.json:5259): Cannot index string with string "works"
jq: error (at series.json:5260): Cannot index string with string "works"
jq: error (at series.json:5261): Cannot index string with string "works"
jq: error (at series.json:5262): Cannot index string with string "works"
jq: error (at series.json:5263): Cannot index string with string "works"
jq: error (at series.json:5264): Cannot index string with string "works"
jq: error (at series.json:5265): Cannot index string with string "works"
jq: error (at series.json:5266): Cannot index string with string "works"
jq: error (at series.json:5267): Cannot index string with string "works"
jq: error (at series.json:5268): Cannot index string with string "works"
jq: error (at series.json:5269): Cannot index string with string "works"
jq: error (at series.json:5270): Cannot index string with string "works"
jq: error (at series.json:5271): Cannot index string with string "works"
jq: error (at series.json:5272): Cannot index string with string "works"
jq: error (at series.json:5273): Cannot index string with string "works"
jq: error (at series.json:5274): Cannot index string with string "works"
jq: error (at series.json:5275): Cannot index string with string "works"
jq: error (at series.json:5276): Cannot index string with string "works"
jq: error (at series.json:5277): Cannot index string with string "works"
jq: error (at series.json:5278): Cannot index string with string "works"
jq: error (at series.json:5279): Cannot index string with string "works"
jq: error (at series.json:5280): Cannot index string with string "works"
jq: error (at series.json:5281): Cannot index string with string "works"
jq: error (at series.json:5282): Cannot index string with string "works"
jq: error (at series.json:5283): Cannot index string with string "works"
jq: error (at series.json:5284): Cannot index string with string "works"
jq: error (at series.json:5285): Cannot index string with string "works"
jq: error (at series.json:5286): Cannot index string with string "works"
jq: error (at series.json:5287): Cannot index string with string "works"
jq: error (at series.json:5288): Cannot index string with string "works"
jq: error (at series.json:5289): Cannot index string with string "works"
jq: error (at series.json:5290): Cannot index string with string "works"
jq: error (at series.json:5291): Cannot index string with string "works"
jq: error (at series.json:5292): Cannot index string with string "works"
jq: error (at series.json:5293): Cannot index string with string "works"
jq: error (at series.json:5294): Cannot index string with string "works"
jq: error (at series.json:5295): Cannot index string with string "works"
jq: error (at series.json:5296): Cannot index string with string "works"
jq: error (at series.json:5297): Cannot index string with string "works"
jq: error (at series.json:5298): Cannot index string with string "works"
Ln 1, Col 347 Spaces: 4 UTF-8 LF Shell Script
```