

LAB TEST-2

SET: J

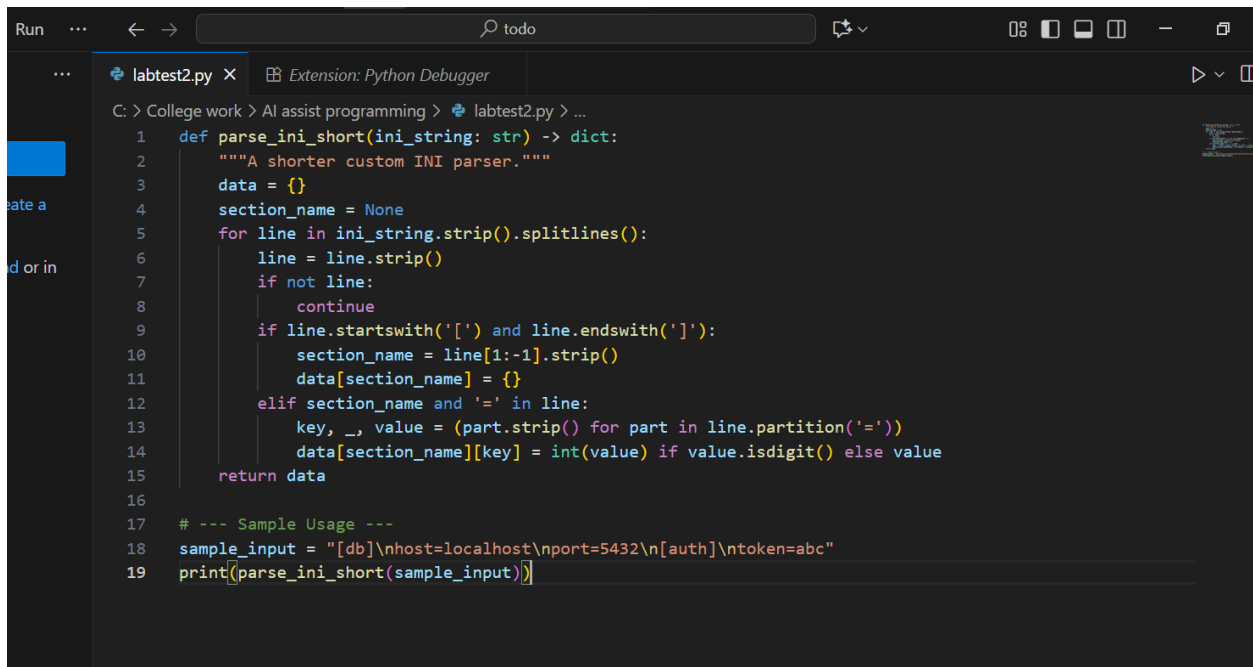
J.1

TASK: Parse INI text into a nested dict; cast numeric ports to int where obvious.

PROMPT:

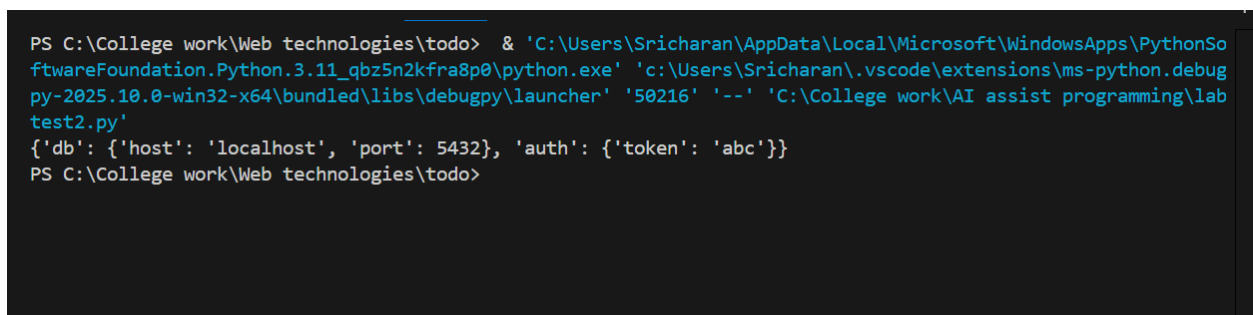
Write a Python function to parse an INI string into a nested dictionary, handling whitespace and casting numeric-only values to integers.

CODE:



```
Run ... < > todo
labtest2.py x Extension: Python Debugger
C: > College work > AI assist programming > labtest2.py > ...
1 def parse_ini_short(ini_string: str) -> dict:
2     """A shorter custom INI parser."""
3     data = {}
4     section_name = None
5     for line in ini_string.strip().splitlines():
6         line = line.strip()
7         if not line:
8             continue
9         if line.startswith('[') and line.endswith(']'):
10            section_name = line[1:-1].strip()
11            data[section_name] = {}
12        elif section_name and '=' in line:
13            key, _, value = (part.strip() for part in line.partition('='))
14            data[section_name][key] = int(value) if value.isdigit() else value
15    return data
16
17 # --- Sample Usage ---
18 sample_input = "[db]\nhost=localhost\nport=5432\n[auth]\ntoken=abc"
19 print(parse_ini_short(sample_input))
```

OUTPUT:



```
PS C:\College work\Web technologies\todo> & 'C:\Users\Sricharan\AppData\Local\Microsoft\WindowsApps\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\python.exe' 'c:\Users\Sricharan\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '50216' '--' 'C:\College work\AI assist programming\labtest2.py'
{'db': {'host': 'localhost', 'port': 5432}, 'auth': {'token': 'abc'}}
PS C:\College work\Web technologies\todo>
```

J.2

TASK:

Compute average duration in minutes from opened -> closed ISO timestamps (naive).

PROMPT: Write a Python function that takes a list of dictionaries and calculates the integer average of the duration in minutes between the 'opened' and 'closed' ISO timestamps in each one.

CODE:

```
labtest2.py x Extension: Python Debugger
C: > College work > AI assist programming > labtest2.py > ...
1 from datetime import datetime
2
3 def calculate_average_sla(tickets: list[dict[str, str]]) -> int:
4     if not tickets:
5         return 0
6
7     total_minutes = 0
8     for ticket in tickets:
9         opened_time = datetime.fromisoformat(ticket['opened'])
10        closed_time = datetime.fromisoformat(ticket['closed'])
11        duration = closed_time - opened_time
12        total_minutes += duration.total_seconds() / 60
13
14    average_minutes = total_minutes / len(tickets)
15    return int(average_minutes)
16
17 sample_input = [
18     {'ticket': 'T1', 'opened': '2025-01-01T10:00:00', 'closed': '2025-01-01T12:15:00'},
19     {'ticket': 'T2', 'opened': '2025-01-01T09:30:00', 'closed': '2025-01-01T10:00:00'}
20 ]
21
22 average_time = calculate_average_sla(sample_input)
23 print(average_time)
```

OUTPUT:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
btest2.py'
82
PS C:\College work\Web technologies\todo>
```