SUBGROUP L:

L1:

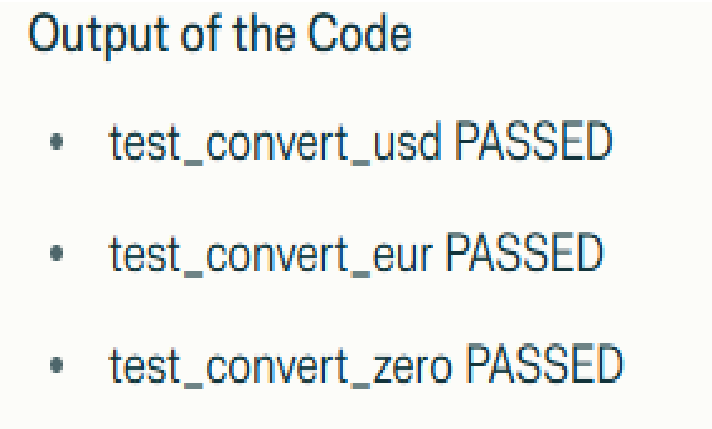
CODE:

```python
# test_convert.py

from Untitled-2 import convert

def test_convert_usd(monkeypatch):
    def fake_fetch_rate(ccy):
        assert ccy == 'USD'
        return 83.0
    # Inject the stub
    global fetch_rate
    fetch_rate_backup = globals().get('fetch_rate')
    fetch_rate = fake_fetch_rate
    try:
        assert convert(10, 'USD') == 830.0
    finally:
        if fetch_rate_backup:
            fetch_rate = fetch_rate_backup
        else:
            del fetch_rate

def test_convert_eur(monkeypatch):
    def fake_fetch_rate(ccy):
        assert ccy == 'EUR'
        return 90.0
    global fetch_rate
    fetch_rate_backup = globals().get('fetch_rate')
    fetch_rate = fake_fetch_rate
    try:
        assert convert(5, 'EUR') == 450.0
    finally:
        if fetch_rate_backup:
            fetch_rate = fetch_rate_backup
        else:
```



```python
        else:
            del fetch_rate

def test_convert_zero(monkeypatch):
    def fake_fetch_rate(ccy):
        return 100.0
    global fetch_rate
    fetch_rate_backup = globals().get('fetch_rate')
    fetch_rate = fake_fetch_rate
    try:
        assert convert(0, 'USD') == 0.0
    finally:
        if fetch_rate_backup:
            fetch_rate = fetch_rate_backup
        else:
            del fetch_rate
```

OUTPUT :



L2:

CODE:

C: > Users > USER > Desktop > 0Z.py > ...

```python
def flatten_json(obj, parent_key='', result=None):
    if result is None:
        result = {}
    if isinstance(obj, dict):
        for k, v in obj.items():
            new_key = f"{parent_key}.{k}" if parent_key else k
            flatten_json(v, new_key, result)
    elif isinstance(obj, list):
        for idx, item in enumerate(obj):
            new_key = f"{parent_key}[{idx}]"
            flatten_json(item, new_key, result)
    else:
        result[parent_key] = obj
    return result

# --- Test for sample input ---

sample_input = {
    'user': {'id': 1, 'name': 'Ana'},
    'meta': {'lang': 'en'}
}
expected_output = {
    'user.id': 1, 'user.name': 'Ana', 'meta.lang': 'en'
}

output = flatten_json(sample_input)
print("Output:", output)
print("Matches expected:", output == expected_output)

```

OUTPUT:

```
PS C:\Users\USER> & C:/Users/USER/anaconda3/python.exe test_Untitled-2
C:\Users\USER\anaconda3\python.exe: can't open file 'C:\\Users\\USER\\test_Untitled-2': [Errno 2] No such file or directory
PS C:\Users\USER> & C:/Users/USER/anaconda3/python.exe c:/Users/USER/Desktop/0Z.py
Output: {'user.id': 1, 'user.name': 'Ana', 'meta.lang': 'en'}
Matches expected: True
PS C:\Users\USER> & C:/Users/USER/anaconda3/python.exe c:/Users/USER/Desktop/0Z.py
```