

```
from unittest.mock import Mock, patch
```

```
import pytest
```

```
from swarms.structs.agent import Agent
```

```
from swarms.structs.swarm_net import SwarmNetwork
```

```
@pytest.fixture
```

```
def swarm_network():
```

```
    agents = [Agent(id=f"Agent_{i}") for i in range(5)]
```

```
    return SwarmNetwork(agents=agents)
```

```
def test_swarm_network_init(swarm_network):
```

```
    assert isinstance(swarm_network.agents, list)
```

```
    assert len(swarm_network.agents) == 5
```

```
@patch("swarms.structs.swarm_net.SwarmNetwork.logger")
```

```
def test_run(mock_logger, swarm_network):
```

```
    swarm_network.run()
```

```
    assert (
```

```
        mock_logger.info.call_count == 10
```

```
    ) # 2 log messages per agent
```

```
def test_run_with_mocked_agents(mock, swarm_network):  
    mock_agents = [Mock(spec=Agent) for _ in range(5)]  
    mock.patch.object(swarm_network, "agents", mock_agents)  
    swarm_network.run()  
    for mock_agent in mock_agents:  
        assert mock_agent.run.called
```

```
def test_swarm_network_with_no_agents():  
    swarm_network = SwarmNetwork(agents=[])  
    assert swarm_network.agents == []
```

```
def test_swarm_network_add_agent(swarm_network):  
    new_agent = Agent(id="Agent_5")  
    swarm_network.add_agent(new_agent)  
    assert len(swarm_network.agents) == 6  
    assert swarm_network.agents[-1] == new_agent
```

```
def test_swarm_network_remove_agent(swarm_network):  
    agent_to_remove = swarm_network.agents[0]  
    swarm_network.remove_agent(agent_to_remove)  
    assert len(swarm_network.agents) == 4  
    assert agent_to_remove not in swarm_network.agents
```