CS3237:PARALLEL PROGRAMMING
FALL2023

DINING PHILOSOPHERS PROBLEM REPORT

200316048 KIYMET NAZLIM

190316040 AYLİN ÇELİK

190316009 İBRAHİM SUDAŞ

Dining philosophers problem:

The dining philosophers problem is a classic synchronization and concurrency problem used to illustrate challenges in concurrent programming. It was formulated by E.W. Dijkstra in 1965 and is often used to discuss issues related to resource allocation and deadlock avoidance in concurrent systems.

The problem is framed around a scenario where a group of philosophers is seated around a dining table. Each philosopher alternates between thinking and eating. The philosophers share a common set of dining utensils, such as forks. To eat, a philosopher needs two forks—one for the left hand and one for the right hand.

The challenge in the dining philosophers problem lies in designing a solution that avoids deadlock and ensures that each philosopher gets a chance to eat without conflicting with others. The potential issues that need to be addressed include:

The dining philosophers problem is a classic synchronization and concurrency problem used to illustrate challenges in concurrent programming. It was formulated by E.W. Dijkstra in 1965 and is often used to discuss issues related to resource allocation and deadlock avoidance in concurrent systems.

The problem is framed around a scenario where a group of philosophers is seated around a dining table. Each philosopher alternates between thinking and eating. The philosophers share a common set of dining utensils, such as forks. To eat, a philosopher needs two forks—one for the left hand and one for the right hand.

The challenge in the dining philosophers problem lies in designing a solution that avoids deadlock and ensures that each philosopher gets a chance to eat without conflicting with others.

STRATEGY: Locked Queue

The strategy of using a "Locked Queue" refers to a solution approach employed to address synchronization issues in problems like the dining philosophers problem. This strategy involves the use of a queue, and this queue, which manages requests for shared resources (such as forks), is protected by lock mechanisms.

Queue: A queue is utilized to manage philosophers' requests for forks. Philosophers add their request to this queue when they want to pick up a fork.

Lock: The queue is protected by a lock mechanism to prevent multiple philosophers from making fork requests simultaneously. This means that when a philosopher is adding a request to the queue or picking up a fork, these operations are performed with the use of a lock.

Fork Acquisition and Release Operations: Philosophers add their fork requests to the queue when they want to pick up a fork. However, to pick up a fork, they need to be in front of the queue. The lock allows only one philosopher at a time to process a fork request. After picking up a fork, when the philosopher wants to release it, the lock is used again to safely add the fork back to the queue.

This strategy ensures that philosophers take turns to perform actions like picking up or releasing forks. As a result, it prevents multiple philosophers from taking the same fork simultaneously, avoiding deadlock situations.

Advantages:

1. **Deadlock Prevention:** This strategy prevents deadlock situations by ensuring that philosophers take turns to perform actions like picking up or releasing forks. Each philosopher processes fork requests by adding them to the queue or releasing forks in a sequential manner using lock mechanisms.
2. **Ease of Understanding:** The use of queues and lock mechanisms makes synchronization and resource sharing easily understandable and manageable. Consequently, code written using this strategy is often more comprehensible and easier to maintain.
3. **General Applicability:** This strategy can be applied in various programming languages and environments. Queues and lock mechanisms are commonly supported features.

Disadvantages:

1. **Performance Cost:** Lock mechanisms may incur a performance cost as they allow only one operation to execute at a time. In large-scale systems, delays can arise due to locking, impacting overall performance.
2. **Slowness and Waiting:** Since philosophers must take turns to perform actions, a philosopher may need to wait to pick up or release a fork. This waiting can potentially decrease the overall performance of the program.
3. **Potential Ordering Issues:** Sequential processing can lead to ordering problems under high workloads. This situation may involve the rapid growth of the queue and frequent changes in ownership of the lock.

STEP BY STEP EXPLANATION OF THE CODE

```python
class Fork:
    def __init__(self, index: int):
        self.index: int = index
        self.lock: threading.Lock = threading.Lock()
        self.picked_up: bool = False
        self.owner: int = -1

    def __enter__(self):
        return self
```

Class representing the state of each fork

What this class includes:

Lock for the fork

Flag indicating whether the fork is picked up

ID of the philosopher who owns the fork

```python
    def __call__(self, owner: int):
        if self.lock.acquire():
            self.owner = owner
            self.picked_up = True
        return self
```

Function indicating the philosopher who picks up the fork

```python
    def __exit__(self, exc_type, exc_value, traceback):
        self.lock.release()
        self.picked_up = False
        self.owner = -1

    def __str__(self):
        return f"F{self.index:2d} ({self.owner:2d})"
```

Function releasing the fork

```python
class DiningPhilosophers:
    def __init__(self, forks):
        self.fork_queue = Queue()
        for fork in forks:
            self.fork_queue.put(fork)

```

Class representing the dining philosophers scenario

```python

    def pick_up_forks(self):
        left_fork = self.fork_queue.get()
        right_fork = self.fork_queue.get()
        return left_fork, right_fork

```

Function for picking up forks

```python

    def put_down_forks(self, left_fork, right_fork):
        self.fork_queue.put(left_fork)
        self.fork_queue.put(right_fork)

```

Function for putting down forks

```python

class Philosopher(threading.Thread):
    def __init__(self, index: int, dining_philosophers: DiningPhilosophers, m: int):
        super().__init__()
        self.index: int = index
        self.dining_philosophers: DiningPhilosophers = dining_philosophers
        self.spaghetti: int = m
        self.state: str = "Thinking"  # Initial state is Thinking
        self.times_eaten: int = 0

```

Thread class representing each philosopher

```python
    def run(self):
        while self.times_eaten < 7:
            self.think()
            self.eat()
```

Main loop of the philosopher

```python

    def think(self):
        time.sleep(3 + random.random() * 3)
        self.state = "Hungry"  # After thinking, philosopher becomes hungry again
```

Function simulating thinking

```python
    def eat(self):
        left_fork, right_fork = self.dining_philosophers.pick_up_forks()

        with left_fork(self.index), right_fork(self.index):
            time.sleep(5 + random.random() * 5)
            self.spaghetti -= 1
            self.state = "Eating"
            time.sleep(5 + random.random() * 5)
            self.state = "Thinking"


        self.dining_philosophers.put_down_forks(left_fork, right_fork)
        self.times_eaten += 1

    def __str__(self):
        return f"P{self.index:2d} ({self.state}, Spaghetti: {self.spaghetti:2d})"

```

Function simulating eating

```python
def print_table(philosophers):
    header = "Philosopher | Spaghetti | Eating | Hungry"
    divider = "-" * len(header)
    rows = [f"{philosopher}" for philosopher in philosophers]

    print(header)
    print(divider)
    for row in rows:
        print(row)
    print(divider)
    print()
```

Helper function to print the table

```python
def main() -> None:
    n: int = 5
    m: int = 7
    forks: list[Fork] = [Fork(i) for i in range(n)]
    dining_philosophers = DiningPhilosophers(forks)
    philosophers: list[Philosopher] = [
        Philosopher(i, dining_philosophers, m) for i in range(n)
    ]
```

Main function

```python
    for philosopher in philosophers:
        philosopher.start()
```

Start all philosopher threads

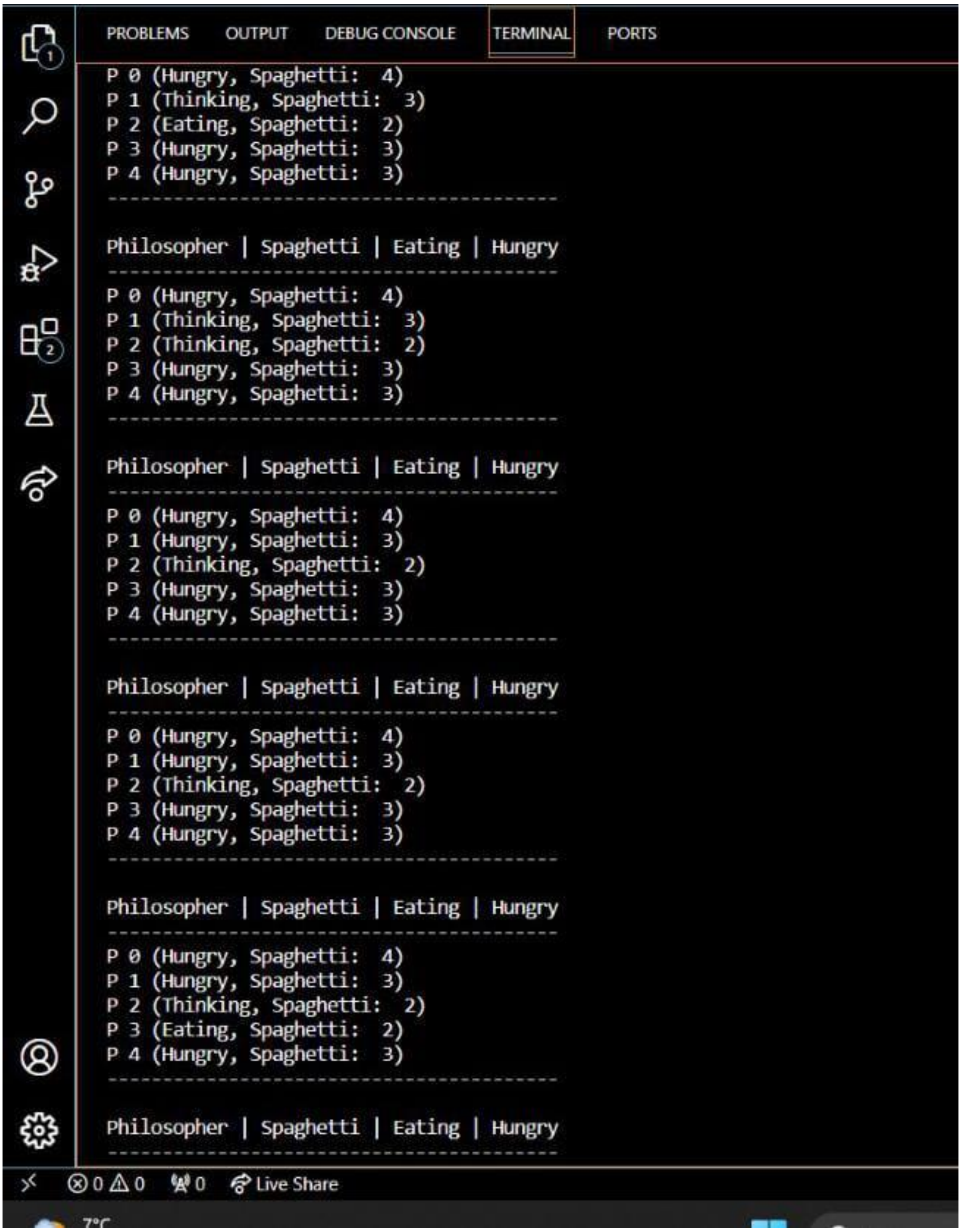
```python
    while any(philosopher.is_alive() for philosopher in philosophers):
        print_table(philosophers)
        time.sleep(1)
```

Print the table until all philosophers finish their meals

```python
    print_table(philosophers)
```

Print the final table

OUTPUT:


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
P 0 (Hungry, Spaghetti: 4)
P 1 (Thinking, Spaghetti: 3)
P 2 (Eating, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 3)
P 4 (Hungry, Spaghetti: 3)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Thinking, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 3)
P 4 (Hungry, Spaghetti: 3)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 3)
P 4 (Hungry, Spaghetti: 3)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 3)
P 4 (Hungry, Spaghetti: 3)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 2)
P 3 (Eating, Spaghetti: 2)
P 4 (Hungry, Spaghetti: 3)

Philosopher | Spaghetti | Eating | Hungry
----
0 0 0
Live Share


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
P 4 (Hungry, Spaghetti: 3)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Eating, Spaghetti: 2)
P 4 (Eating, Spaghetti: 2)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Thinking, Spaghetti: 2)
P 4 (Eating, Spaghetti: 2)

Philosopher | Spaghetti | Eating | Hungry

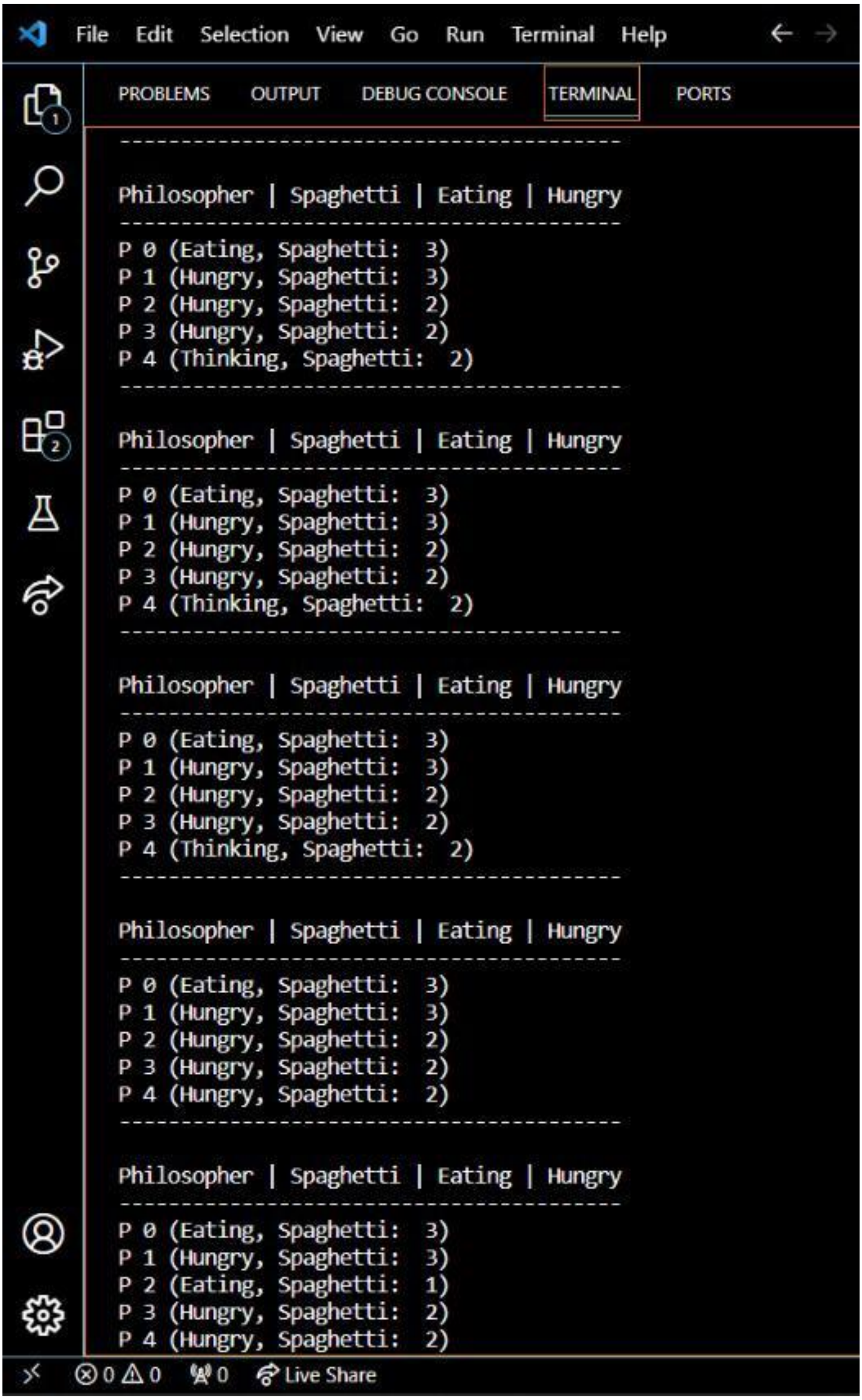
P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Thinking, Spaghetti: 2)
P 4 (Eating, Spaghetti: 2)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Thinking, Spaghetti: 2)
P 4 (Eating, Spaghetti: 2)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 4)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 2)
0 0 0 Live Share
7°C
Kısmen güneşli


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Philosopher | Spaghetti | Eating | Hungry

P 0 (Eating, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 2)
P 4 (Thinking, Spaghetti: 2)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Eating, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 2)
P 4 (Thinking, Spaghetti: 2)

Philosopher | Spaghetti | Eating | Hungry

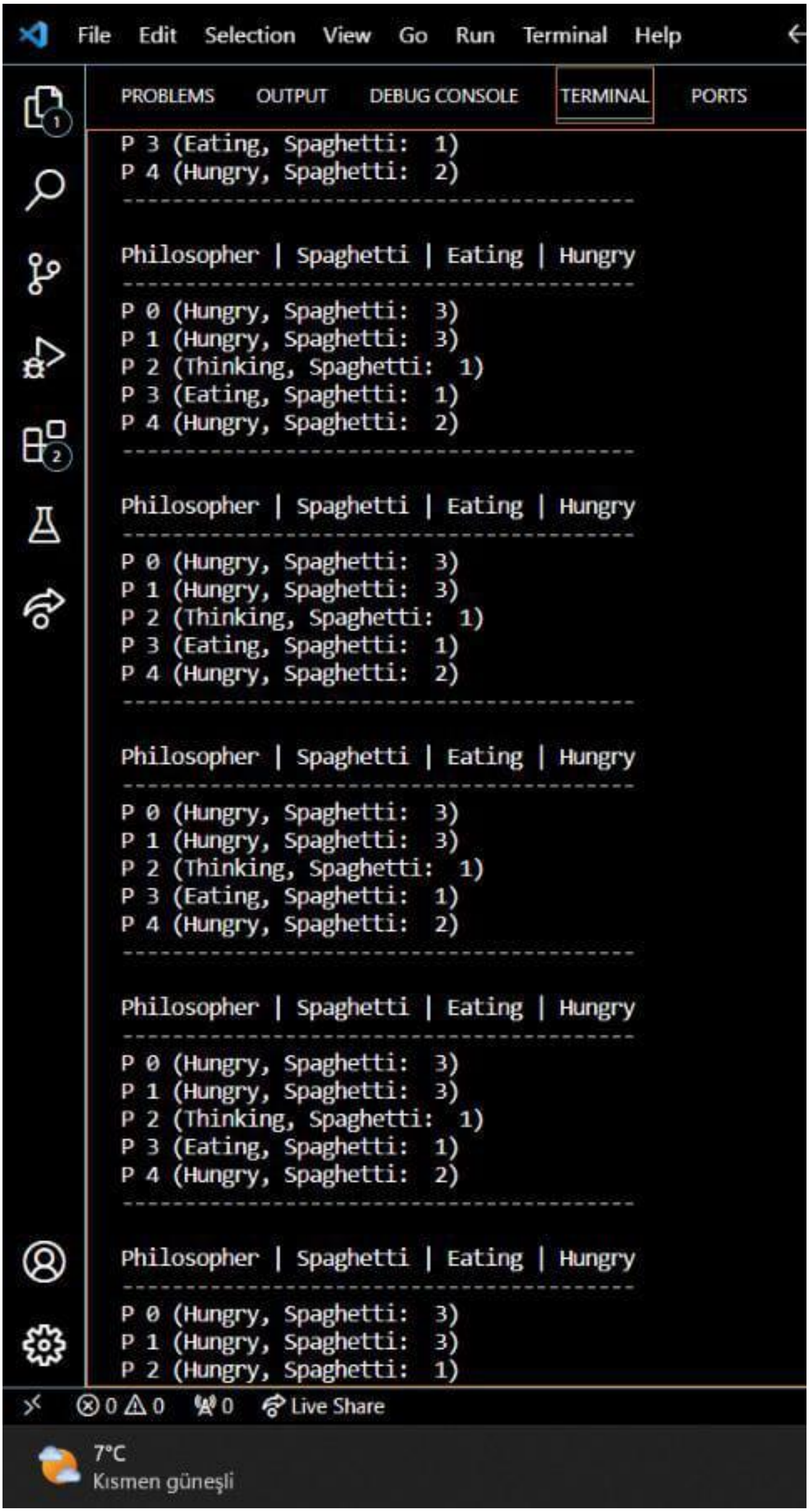
P 0 (Eating, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 2)
P 4 (Thinking, Spaghetti: 2)

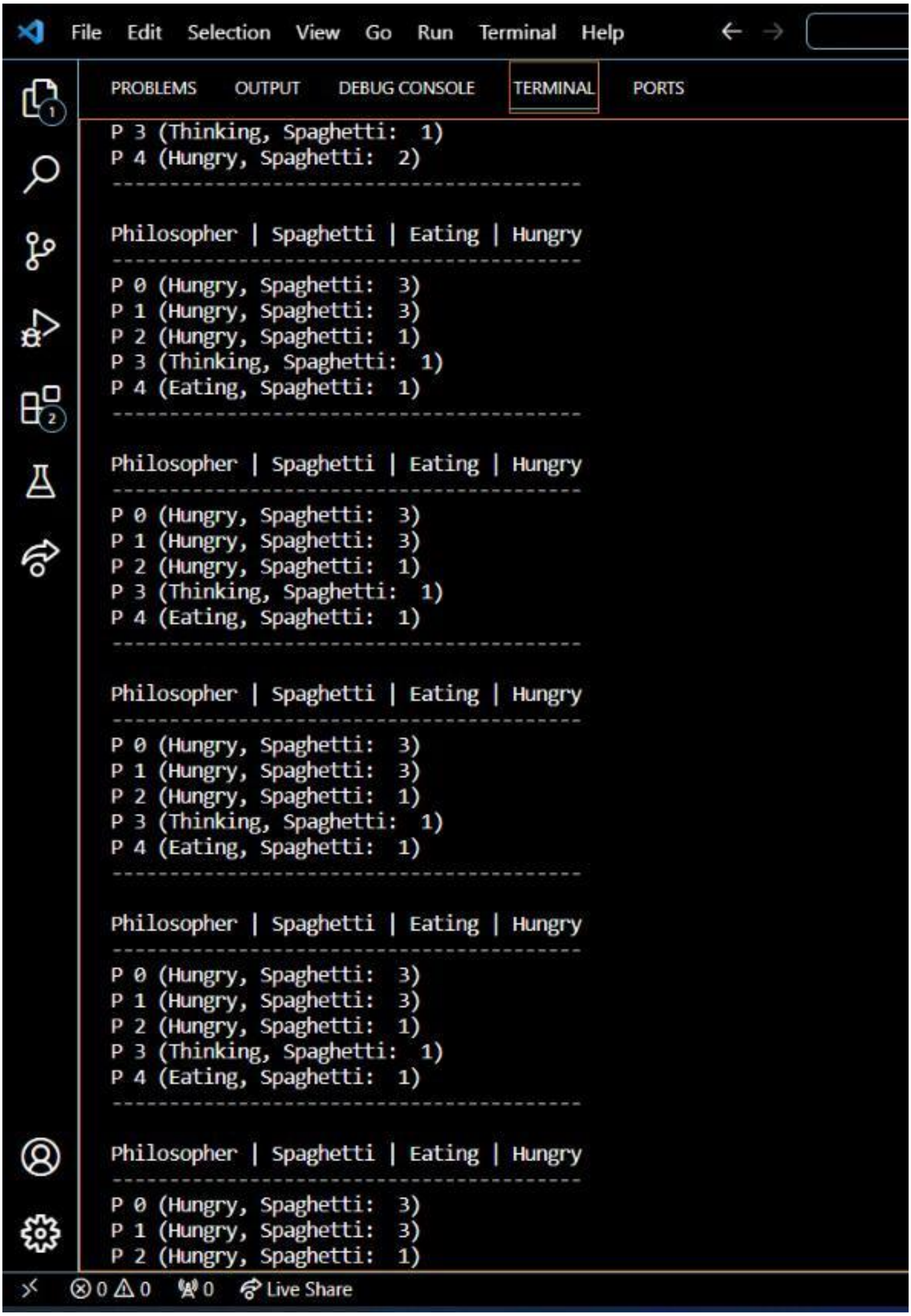
Philosopher | Spaghetti | Eating | Hungry

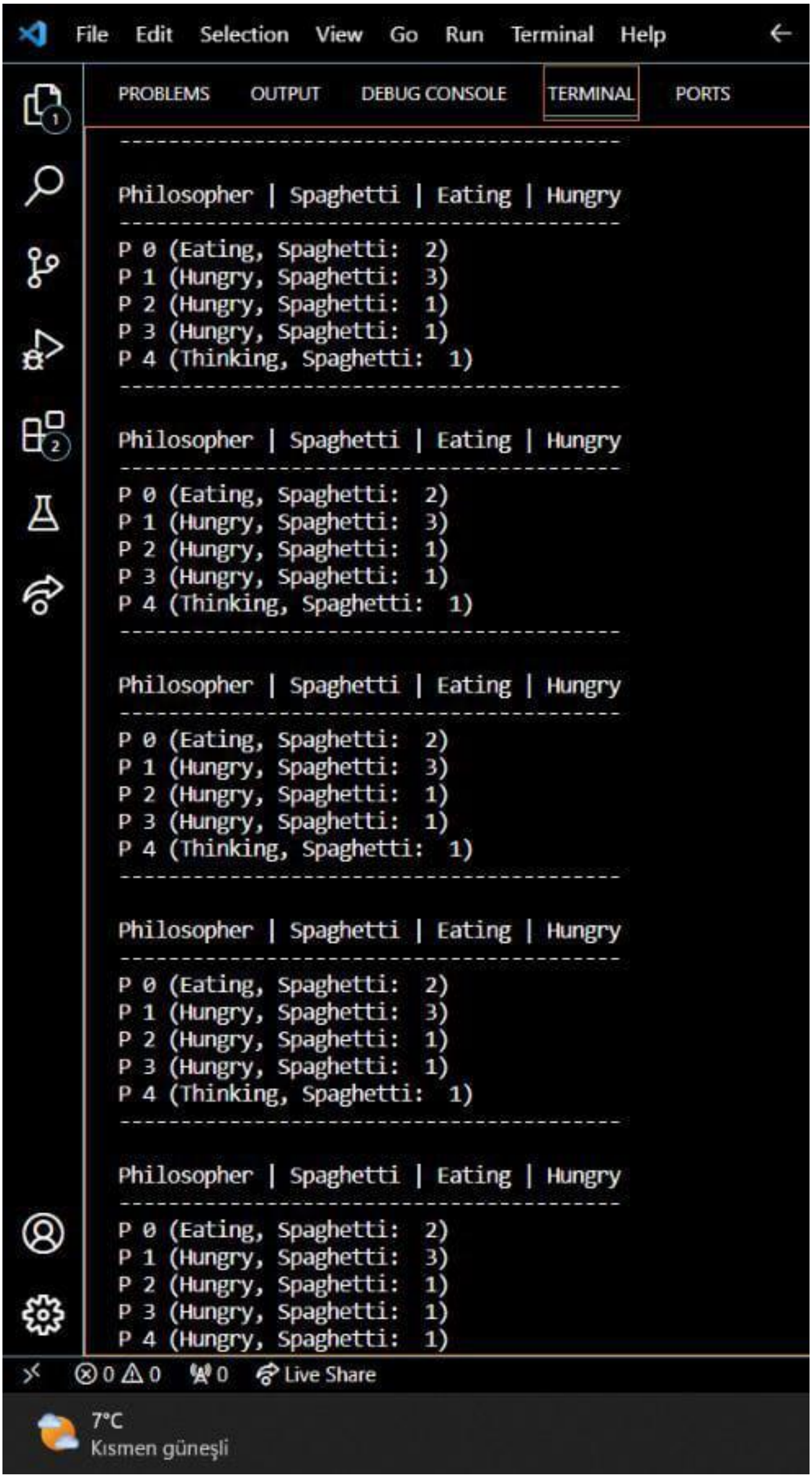
P 0 (Eating, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 2)
P 3 (Hungry, Spaghetti: 2)
P 4 (Hungry, Spaghetti: 2)

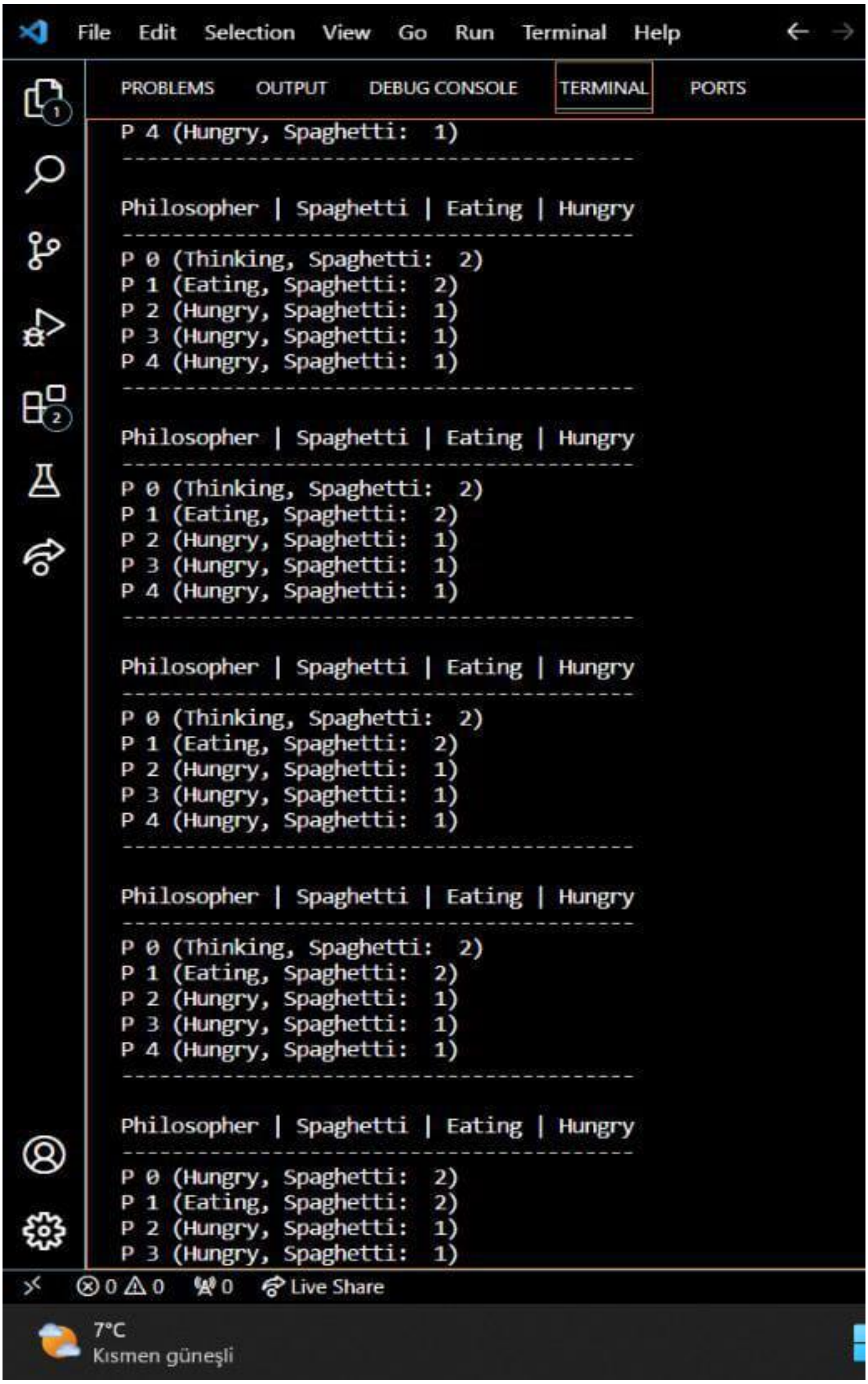
Philosopher | Spaghetti | Eating | Hungry

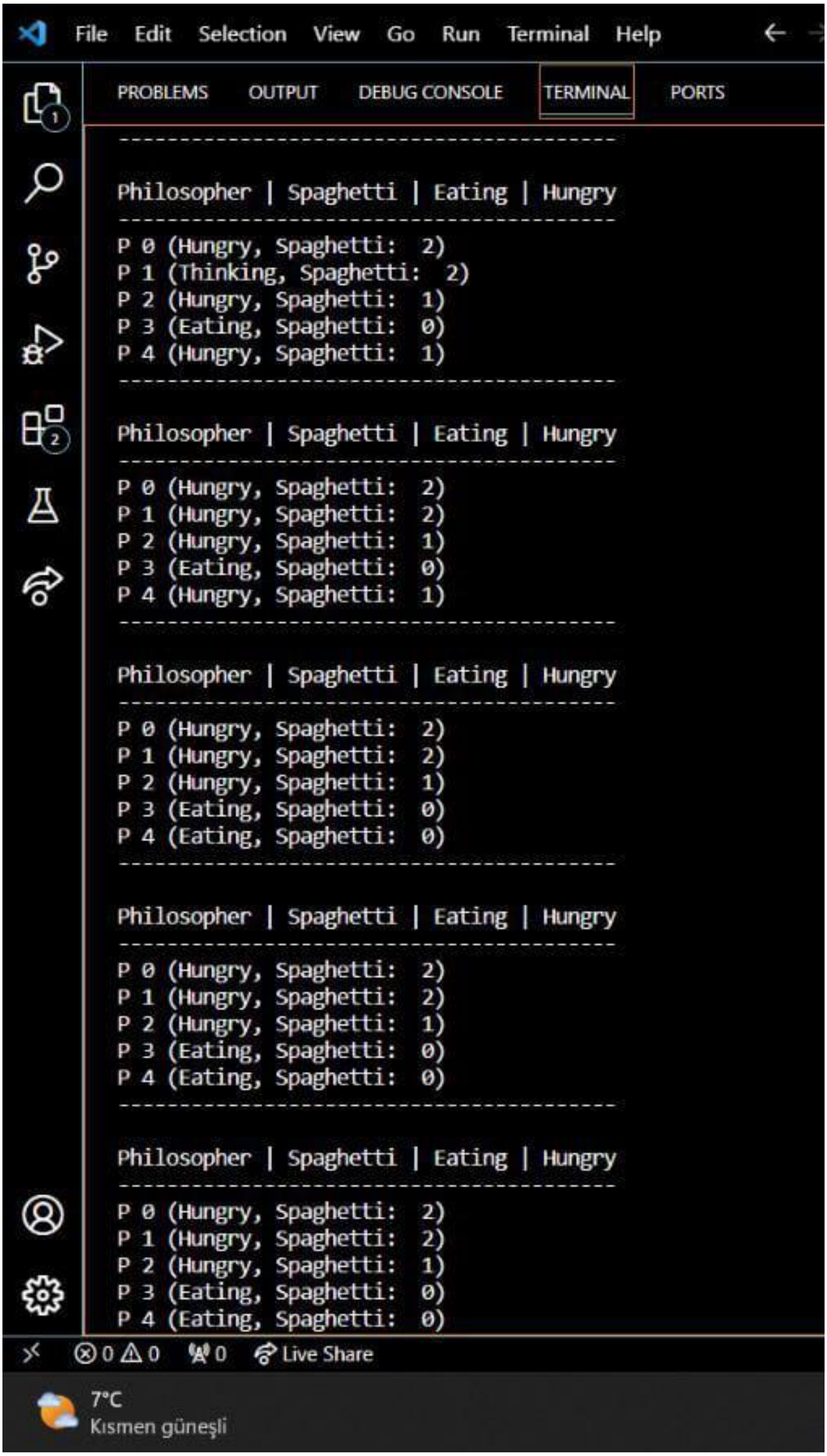
P 0 (Eating, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Eating, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 2)
P 4 (Hungry, Spaghetti: 2)
⊗ 0 ⚠ 0 0 Live Share


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
P 3 (Eating, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 2)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 1)
P 3 (Eating, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 2)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 1)
P 3 (Eating, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 2)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 1)
P 3 (Eating, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 2)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Thinking, Spaghetti: 1)
P 3 (Eating, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 2)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
0 0 0 Live Share
7°C
Kısmen güneşli


File
Edit
Selection
View
Go
Run
Terminal
Help
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
P 3 (Thinking, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 2)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 1)
P 4 (Eating, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 1)
P 4 (Eating, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 1)
P 4 (Eating, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 1)
P 4 (Eating, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 3)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
0 0
0
Live Share


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Thinking, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Thinking, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Thinking, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Thinking, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 3)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 1)
0 0 0 Live Share
7°C
Kısmen güneşli


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
P 4 (Hungry, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Thinking, Spaghetti: 2)
P 1 (Eating, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Thinking, Spaghetti: 2)
P 1 (Eating, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Thinking, Spaghetti: 2)
P 1 (Eating, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Thinking, Spaghetti: 2)
P 1 (Eating, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
P 4 (Hungry, Spaghetti: 1)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 2)
P 1 (Eating, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Hungry, Spaghetti: 1)
0 0 0 Live Share
7°C
Kısmen güneşli


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 2)
P 1 (Thinking, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Eating, Spaghetti: 0)
P 4 (Hungry, Spaghetti: 1)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Eating, Spaghetti: 0)
P 4 (Hungry, Spaghetti: 1)

Philosopher | Spaghetti | Eating | Hungry

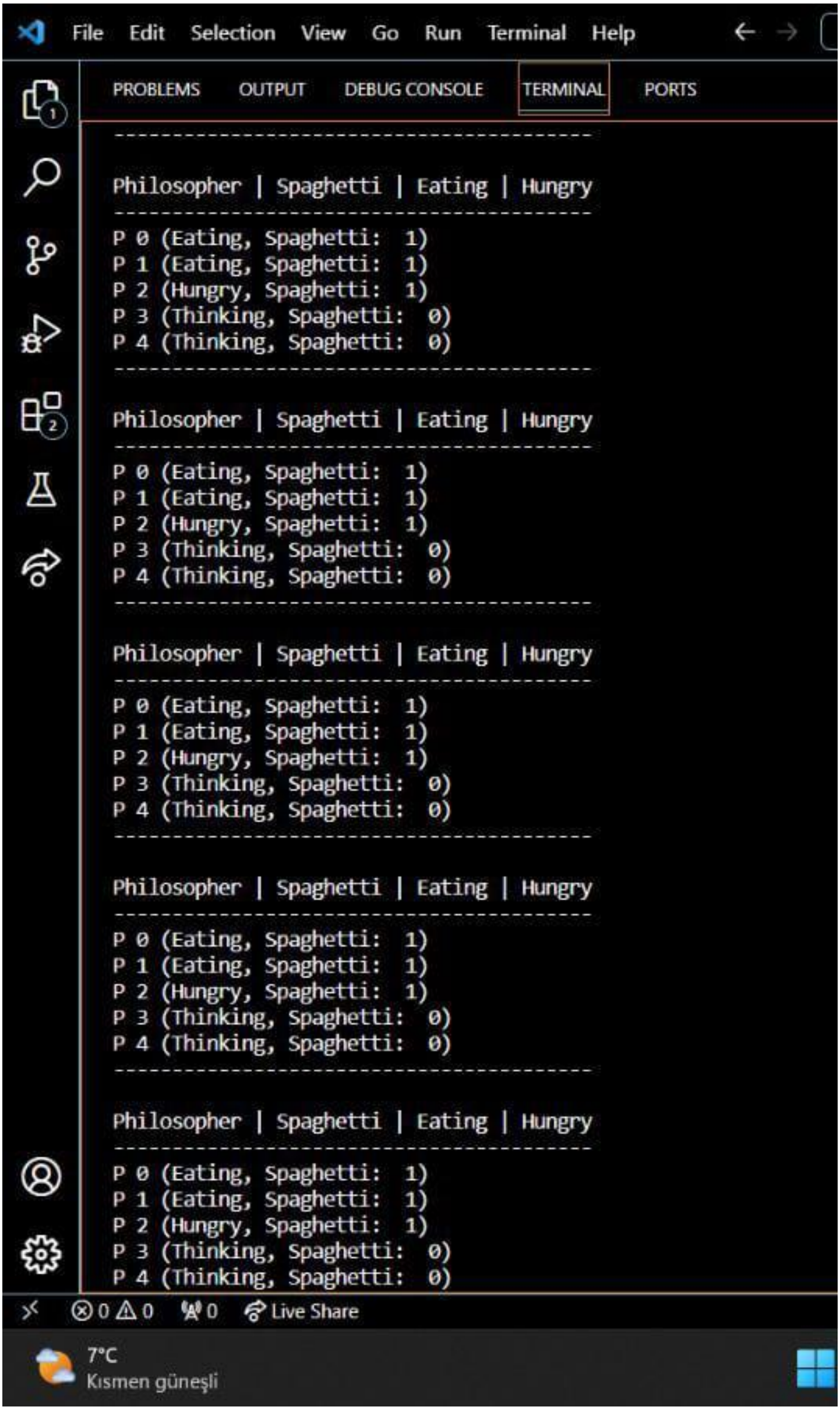
P 0 (Hungry, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Eating, Spaghetti: 0)
P 4 (Eating, Spaghetti: 0)

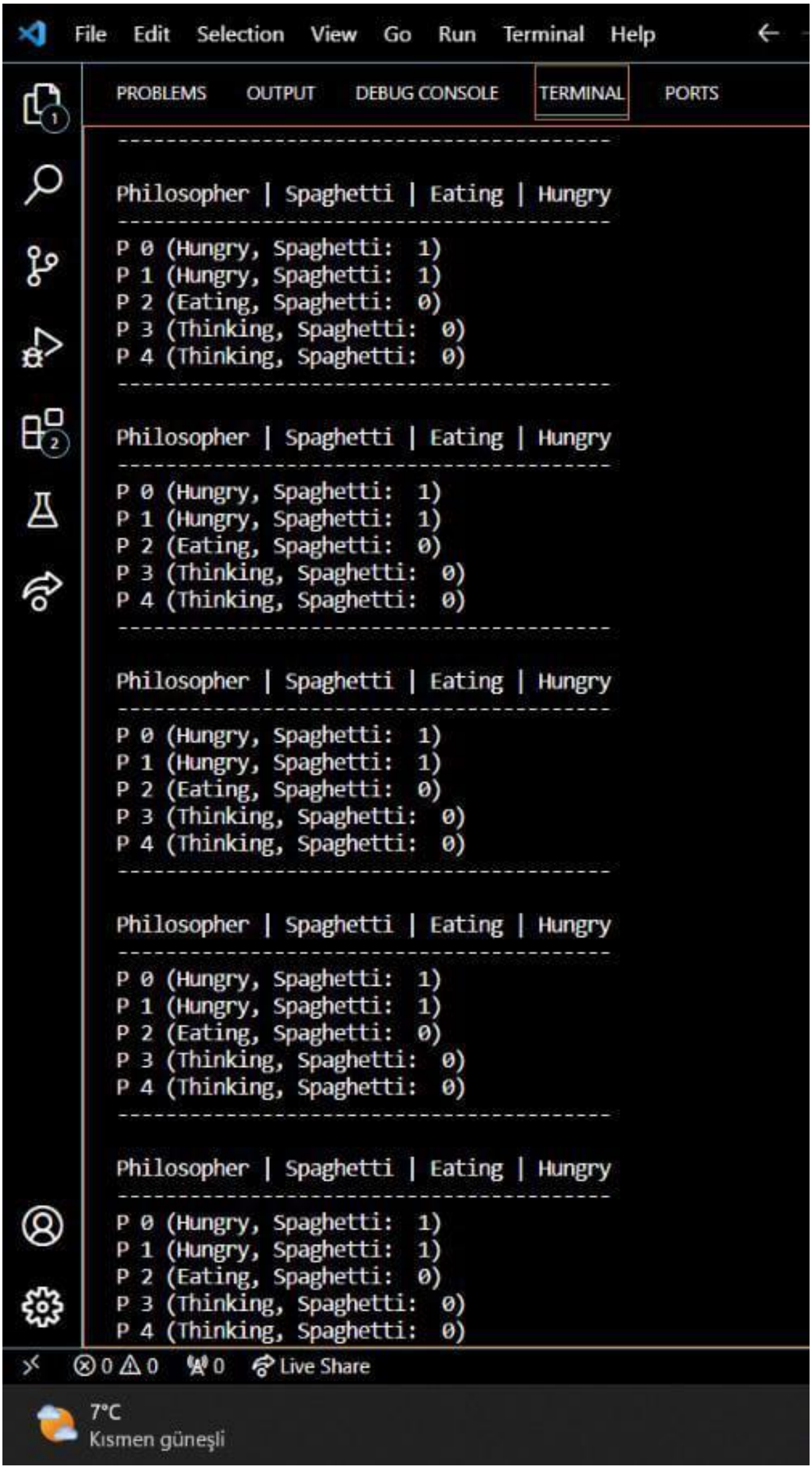
Philosopher | Spaghetti | Eating | Hungry

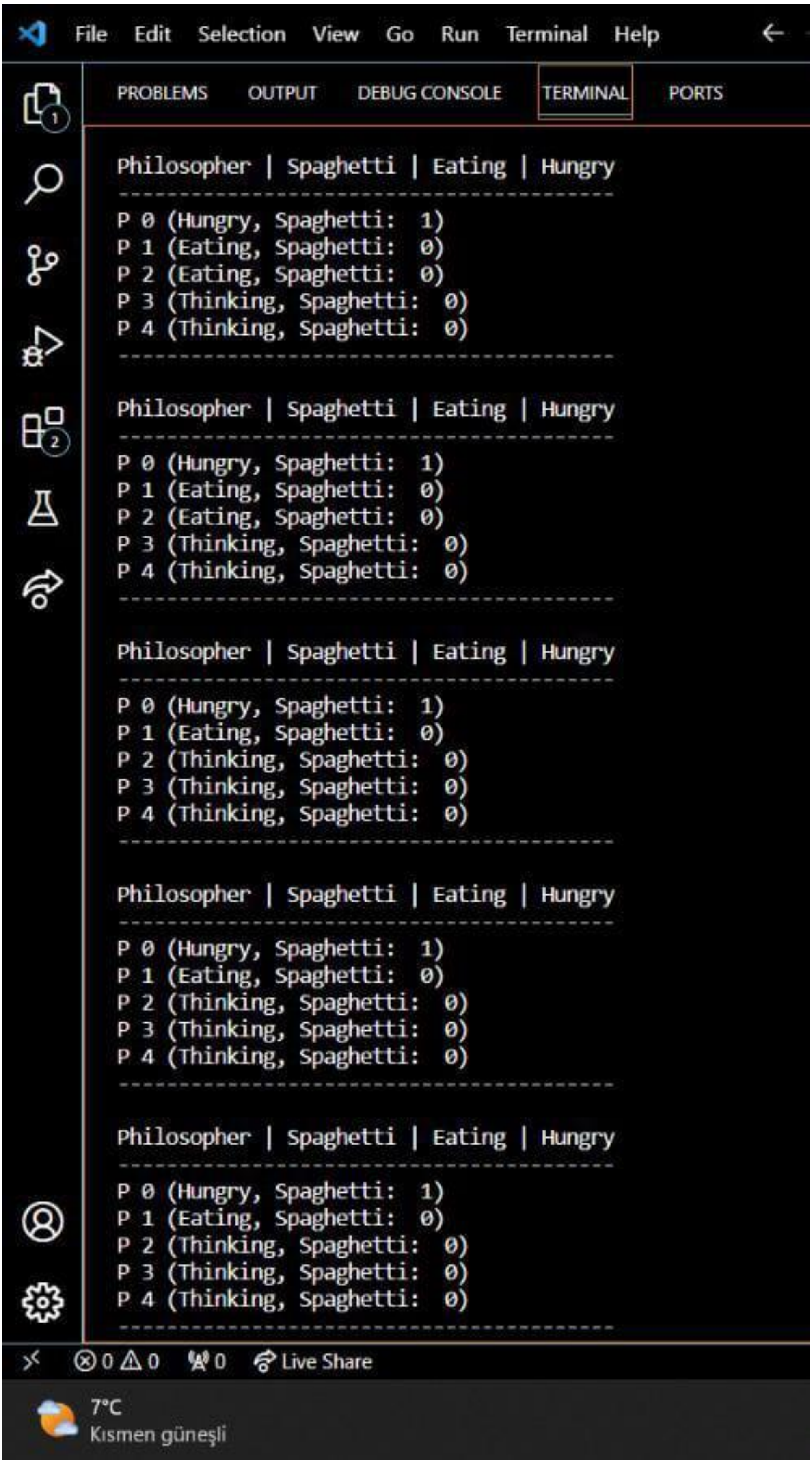
P 0 (Hungry, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Eating, Spaghetti: 0)
P 4 (Eating, Spaghetti: 0)

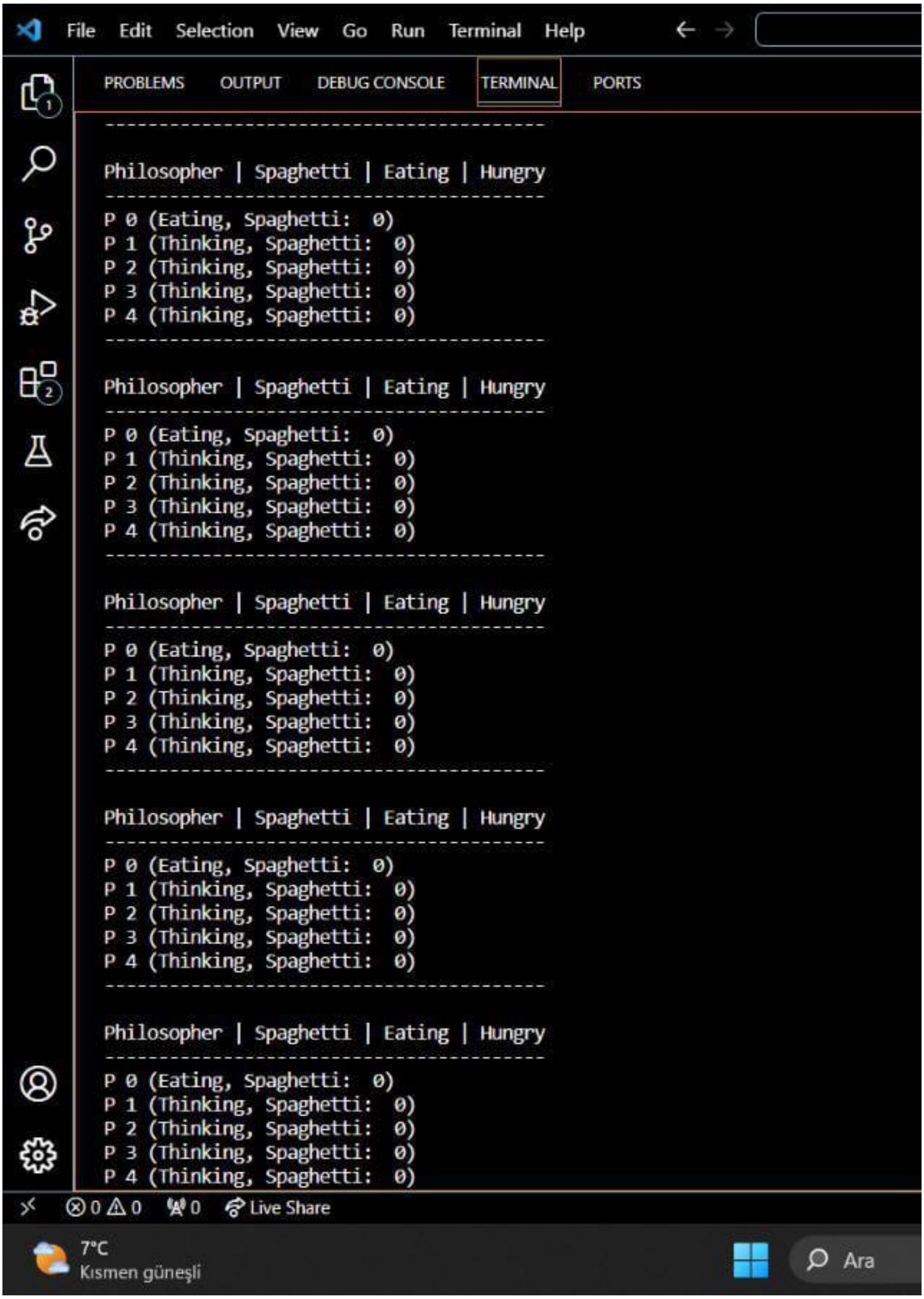
Philosopher | Spaghetti | Eating | Hungry

P 0 (Hungry, Spaghetti: 2)
P 1 (Hungry, Spaghetti: 2)
P 2 (Hungry, Spaghetti: 1)
P 3 (Eating, Spaghetti: 0)
P 4 (Eating, Spaghetti: 0)
0 0 0
Live Share
7°C
Kısmen güneşli


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 1)
P 1 (Eating, Spaghetti: 1)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 1)
P 1 (Eating, Spaghetti: 1)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 1)
P 1 (Eating, Spaghetti: 1)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 1)
P 1 (Eating, Spaghetti: 1)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Eating, Spaghetti: 1)
P 1 (Eating, Spaghetti: 1)
P 2 (Hungry, Spaghetti: 1)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
0 0 0 Live Share
7°C
Kısmen güneşli


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Hungry, Spaghetti: 1)
P 2 (Eating, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Hungry, Spaghetti: 1)
P 2 (Eating, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Hungry, Spaghetti: 1)
P 2 (Eating, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Hungry, Spaghetti: 1)
P 2 (Eating, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Hungry, Spaghetti: 1)
P 2 (Eating, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
0 0 0 Live Share
7°C
Kısmen güneşli


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Eating, Spaghetti: 0)
P 2 (Eating, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Eating, Spaghetti: 0)
P 2 (Eating, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Eating, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Eating, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
Philosopher | Spaghetti | Eating | Hungry
P 0 (Hungry, Spaghetti: 1)
P 1 (Eating, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
0 0 0 Live Share
7°C
Kısmen güneşli


File Edit Selection View Go Run Terminal Help
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Philosopher | Spaghetti | Eating | Hungry

P 0 (Eating, Spaghetti: 0)
P 1 (Thinking, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Eating, Spaghetti: 0)
P 1 (Thinking, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Eating, Spaghetti: 0)
P 1 (Thinking, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Eating, Spaghetti: 0)
P 1 (Thinking, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)

Philosopher | Spaghetti | Eating | Hungry

P 0 (Eating, Spaghetti: 0)
P 1 (Thinking, Spaghetti: 0)
P 2 (Thinking, Spaghetti: 0)
P 3 (Thinking, Spaghetti: 0)
P 4 (Thinking, Spaghetti: 0)
0 0 0 Live Share
7°C
Kısmen güneşli
Ara