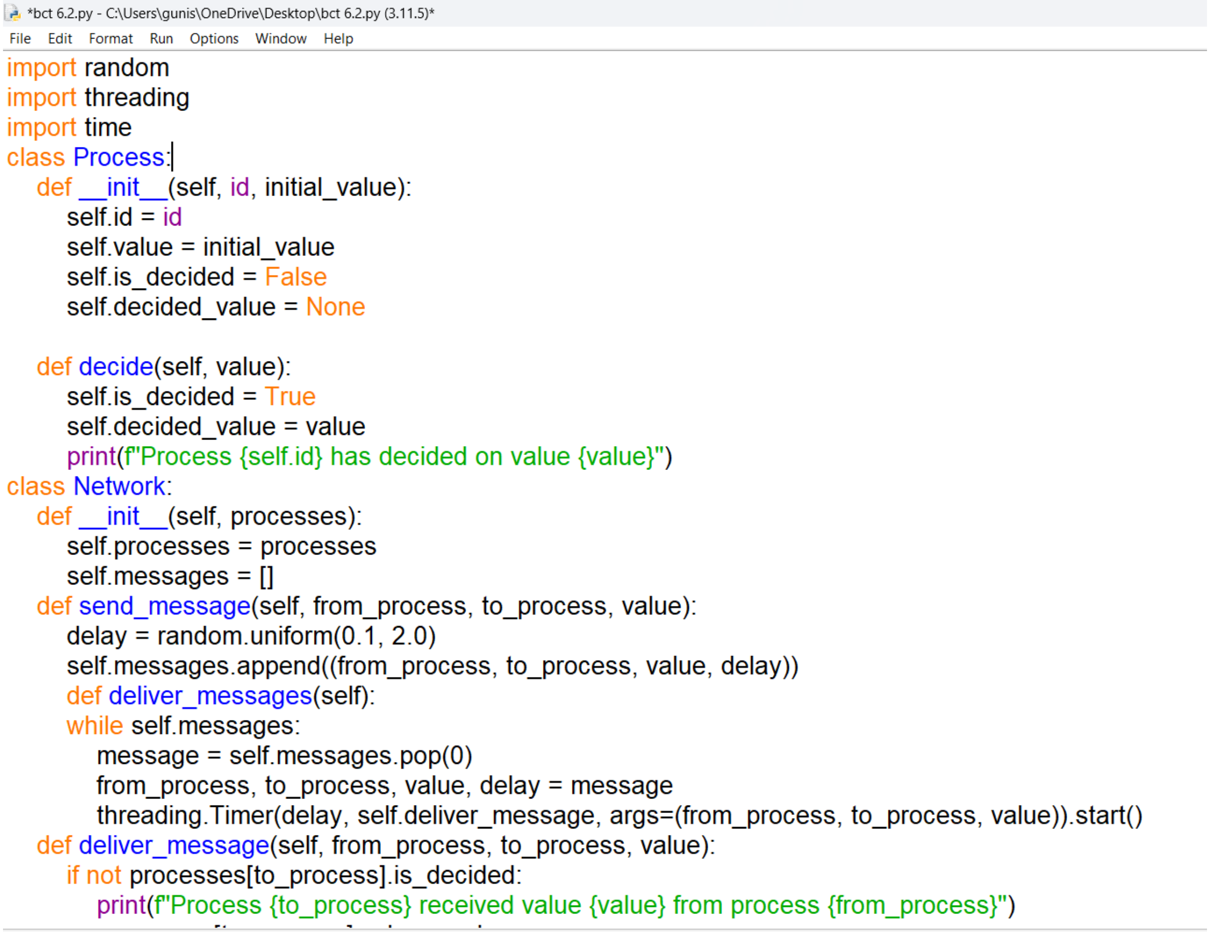
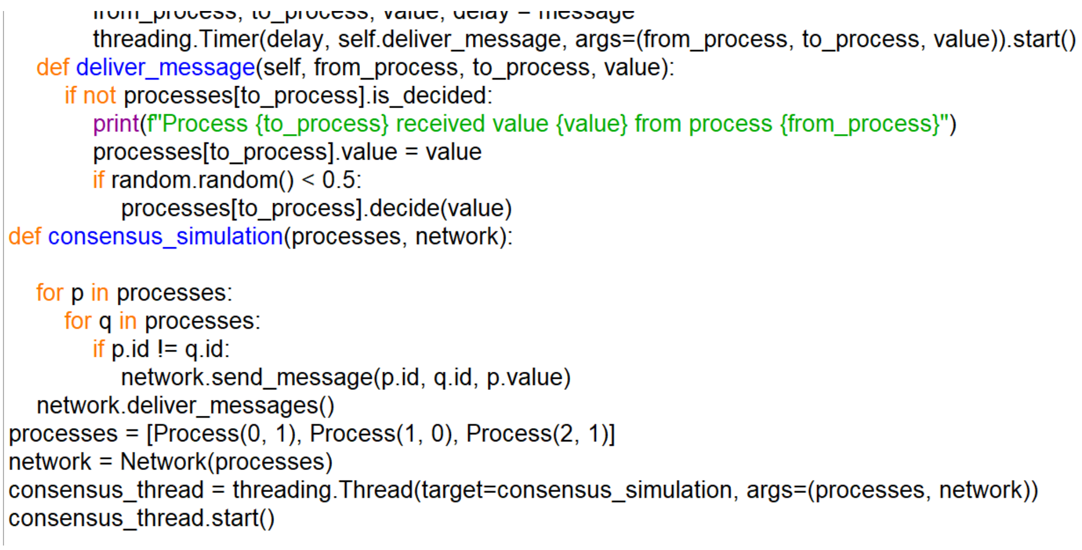
Lab 6

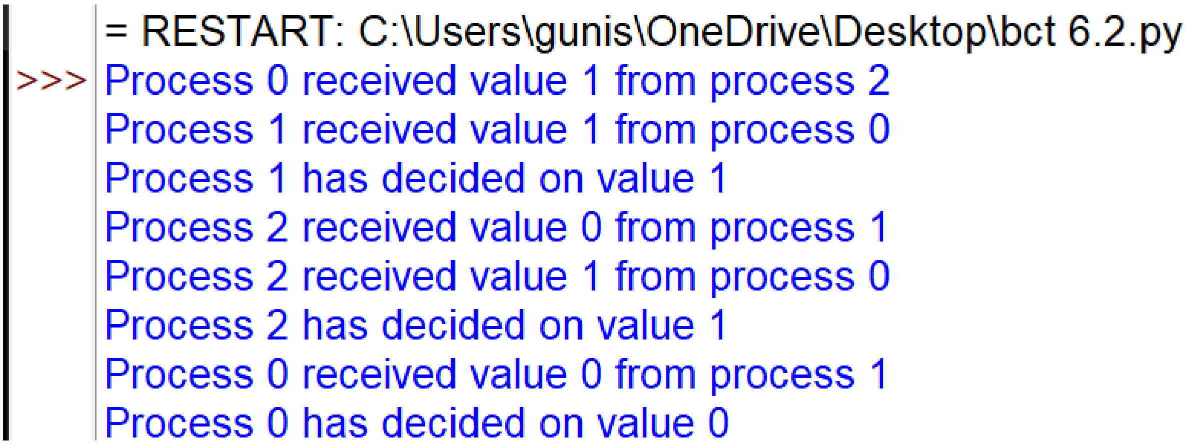

bct 6.2.py - C:\Users\gunis\OneDrive\Desktop\bct 6.2.py (3.11.5)
File Edit Format Run Options Window Help

```
import random
import threading
import time
class Process:
    def __init__(self, id, initial_value):
        self.id = id
        self.value = initial_value
        self.is_decided = False
        self.decided_value = None

    def decide(self, value):
        self.is_decided = True
        self.decided_value = value
        print(f"Process {self.id} has decided on value {value}")
class Network:
    def __init__(self, processes):
        self.processes = processes
        self.messages = []
    def send_message(self, from_process, to_process, value):
        delay = random.uniform(0.1, 2.0)
        self.messages.append((from_process, to_process, value, delay))
        def deliver_messages(self):
        while self.messages:
            message = self.messages.pop(0)
            from_process, to_process, value, delay = message
            threading.Timer(delay, self.deliver_message, args=(from_process, to_process, value)).start()
    def deliver_message(self, from_process, to_process, value):
        if not processes[to_process].is_decided:
            print(f"Process {to_process} received value {value} from process {from_process}")
```



```
            from_process, to_process, value, delay = message
            threading.Timer(delay, self.deliver_message, args=(from_process, to_process, value)).start()
    def deliver_message(self, from_process, to_process, value):
        if not processes[to_process].is_decided:
            print(f"Process {to_process} received value {value} from process {from_process}")
            processes[to_process].value = value
            if random.random() < 0.5:
                processes[to_process].decide(value)
def consensus_simulation(processes, network):

    for p in processes:
        for q in processes:
            if p.id != q.id:
                network.send_message(p.id, q.id, p.value)
    network.deliver_messages()
processes = [Process(0, 1), Process(1, 0), Process(2, 1)]
network = Network(processes)
consensus_thread = threading.Thread(target=consensus_simulation, args=(processes, network))
consensus_thread.start()
```

Output



```
    = RESTART: C:\Users\gunis\OneDrive\Desktop\bct 6.2.py
>>> Process 0 received value 1 from process 2
    Process 1 received value 1 from process 0
    Process 1 has decided on value 1
    Process 2 received value 0 from process 1
    Process 2 received value 1 from process 0
    Process 2 has decided on value 1
    Process 0 received value 0 from process 1
    Process 0 has decided on value 0
```