```
from multiprocessing import Process
from multiprocessing import current_process
from multiprocessing import Value, Array

N = 8
def task(common, tid, turn):
    a = 0
    for i in range(100):
        print(f’{tid}−{i}: Non−critical Section’)
        a += 1
        print(f’{tid}−{i}: End of non−critical Section’)
        while turn.value!=tid:
            pass
        print(f’{tid}−{i}: Critical section’)
        v = common.value + 1
        print(f’{tid}−{i}: Inside critical section’)
        common.value = v
        print(f’{tid}−{i}: End of critical section’)
        turn.value = (tid + 1) % N

def main():
    lp = []
    common = Value(’i’, 0)
    turn = Value(’i’, 0)
    for tid in range(N):
        lp.append(Process(target=task, args=(common, tid, turn)))
    print (f"Valor inicial del contador {common.value}")
    for p in lp:
        p.start()

    for p in lp:
        p.join()

    print (f"Valor final del contador {common.value}")
    print ("fin")


if __name__ == "__main__":
    main()
```

```python
from multiprocessing import Process
from multiprocessing import current_process
from multiprocessing import Value, Array

N = 8

def is_anybody_inside(critical, tid):
    found = False
    i = 0
    while i<len(critical) and not found:
        found = tid!=i and critical[i]==1
        i += 1
    return found

def task(common, tid, critical):
    a = 0
    for i in range(100):
        print(f'{tid}-{i}: Non-critical Section')
        a += 1
        print(f'{tid}-{i}: End of non-critical Section')
        while is_anybody_inside(critical, tid):
            pass
        critical[tid] = 1
        print(f'{tid}-{i}: Critical section')
        v = common.value + 1
        print(f'{tid}-{i}: Inside critical section')
        common.value = v
        print(f'{tid}-{i}: End of critical section')
        critical[tid] = 0

def main():
    lp = []
    common = Value('i', 0)
    critical = Array('i', [0]*N)
    for tid in range(N):
        lp.append(Process(target=task, args=(common, tid, critical)))
    print (f"Valor inicial del contador {common.value}")
    for p in lp:
        p.start()

    for p in lp:
        p.join()

    print (f"Valor final del contador {common.value}")
    print ("fin")


if __name__ == "__main__":
    main()
```

```
from multiprocessing import Process
from multiprocessing import current_process
from multiprocessing import Value, Array

N = 8

def is_anybody_inside(critical, tid):
    found = False
    i = 0
    while i<len(critical) and not found:
        found = tid!=i and critical[i]==1
        i += 1
    return found

def task(common, tid, critical):
    a = 0
    for i in range(100):
        print(f’{tid}–{i}: Non–critical Section’)
        a += 1
        print(f’{tid}–{i}: End of non–critical Section’)
        critical[tid] = 1
        while is_anybody_inside(critical, tid):
            critical[tid] = 0
            print(f’{tid}–{i}: Giving up’)
            critical[tid] = 1
        print(f’{tid}–{i}: Critical section’)
        v = common.value + 1
        print(f’{tid}–{i}: Inside critical section’)
        common.value = v
        print(f’{tid}–{i}: End of critical section’)
        critical[tid] = 0

def main():
    lp = []
    common = Value(’i’, 0)
    critical = Array(’i’, [0]*N)
    for tid in range(N):
        lp.append(Process(target=task, args=(common, tid, critical)))
    print (f"Valor inicial del contador {common.value}")
    for p in lp:
        p.start()

    for p in lp:
        p.join()

    print (f"Valor final del contador {common.value}")
    print ("fin")


if __name__ == "__main__":
    main()
```

```
from multiprocessing import Process
from multiprocessing import current_process
from multiprocessing import Value, Array

N = 8

def is_anybody_inside(critical, tid):
    found = False
    i = 0
    while i<len(critical) and not found:
        found = tid!=i and critical[i]==1
        i += 1
    return found

def task(common, tid, critical, turn):
    a = 0
    for i in range(100):
        print(f’{tid}-{i}: Non-critical Section’)
        a += 1
        print(f’{tid}-{i}: End of non-critical Section’)
        critical[tid] = 1
        while is_anybody_inside(critical, tid):
            critical[tid] = 0
            print(f’{tid}-{i}: Giving up’)
            while turn.value==tid:
                pass
            critical[tid] = 1
        print(f’{tid}-{i}: Critical section’)
        v = common.value + 1
        print(f’{tid}-{i}: Inside critical section’)
        common.value = v
        print(f’{tid}-{i}: End of critical section’)
        critical[tid] = 0
        turn.value = tid

def main():
    lp = []
    common = Value(’i’, 0)
    critical = Array(’i’, [0]*N)
    turn = Value(’i’, 0)
    for tid in range(N):
        lp.append(Process(target=task, args=(common, tid, critical, turn)))
    print (f"Valor inicial del contador {common.value}")
    for p in lp:
        p.start()

    for p in lp:
        p.join()

    print (f"Valor final del contador {common.value}")
    print ("fin")


if __name__ == "__main__":
    main()
```