Question-1)producer-consumer

```
from queue import Queue
from threading import Thread

def producer(q):
    for i in range(5):
        q.put(i)
        print(f"Produced: {i} ")

def consumer(q):
    while not q.empty():
        print(f"Consumed: {q.get()}")

q = Queue()
Thread(target=producer, args=(q,)).start()
Thread(target=consumer, args=(q,)).start()
```

output:

```
Produced: 0Consumed: 0

Produced: 1Consumed: 1

Produced: 2Consumed: 2

Produced: 3Consumed: 3

Produced: 4Consumed: 4
```

question-2)RR(round robin) Scheduling algorithm

```
def rr(processes, bt, q):
    t, rem_bt = 0, bt[:]
    while any(rem_bt):
        for i in range(len(processes)):
            if rem_bt[i] > 0:
                t += min(rem_bt[i], q)
                print(f"P{processes[i]}: {min(rem_bt[i], q)} exec, {rem_bt[i]-q
if rem_bt[i]>q else 0} left, t={t}")
                rem_bt[i] = max(0, rem_bt[i] - q)

# Example
rr([1, 2, 3],[10,5,8],3)
```

output:

```
P1: 3 exec, 7 left, t=3
P2: 3 exec, 2 left, t=6
P3: 3 exec, 5 left, t=9
P1: 3 exec, 4 left, t=12
P2: 2 exec, 0 left, t=14
P3: 3 exec, 2 left, t=17
P1: 3 exec, 1 left, t=20
P3: 2 exec, 0 left, t=22
P1: 1 exec, 0 left, t=23
```

question-3) WAP to design a File System

```
fs = {}
fs["file1"] = "Hello"
fs["file1"] += " World"
print(fs["file1"])  # Output: Hello World
print(list(fs.keys()))  # Output: ['file1']
```

output:

```
Hello World
['file1']
```

question-4)wAP to work with a single thread

```
import threading

def task():
    for i in range(5):
        print(f"Task running: {i}")

thread = threading.Thread(target=task)
thread.start()
thread.join()
print("Task completed.")
```

output:

```
Task running: 0
Task running: 1
Task running: 2
Task running: 3
Task running: 4
Task completed.
```

question-5)wAP to work with a multi threads

```
from threading import Thread

def task(name):
    for i in range(3):
        print(f"{name} running:{i} ")

threads = [Thread(target=task, args=(f"Thread-{i+1}",)) for i in range(2)]
for t in threads:t.start()
for t in threads:t.join()
```

output:

Thread-1 running:0 Thread-2 running:0

Thread-1 running:1 Thread-2 running:1

Thread-1 running:2 Thread-2 running:2