README

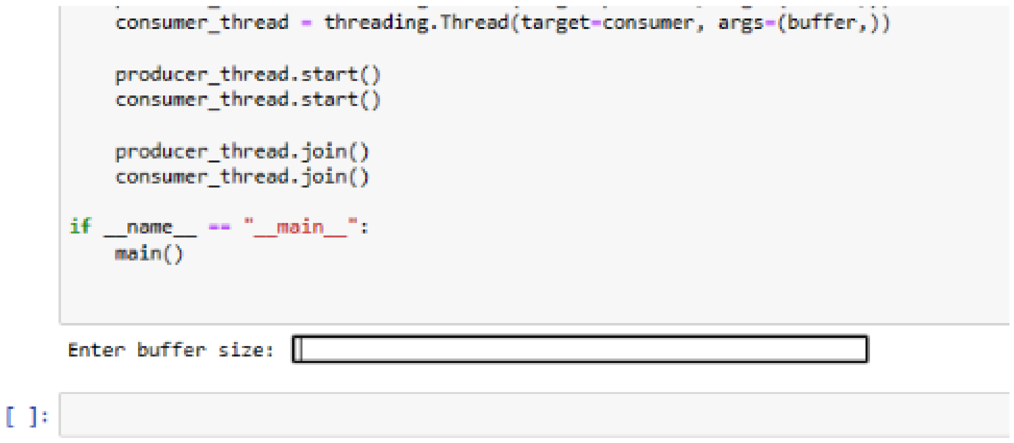
This Python code implements a simple producer and consumer problem using socket programming. The code creates a server and client that communicate through sockets to produce and consume items from a buffer.

The code begins by importing the socket and threading modules. It then defines a Buffer class with methods for producing and consuming items from the buffer. The produce() method appends an item to the buffer as long as the buffer is not full, printing the produced item. If the buffer is full, it prints a message and waits for the consumer to consume an item. The consume() method removes an item from the buffer if it is not empty, printing the consumed item. If the buffer is empty, it prints a message and waits for the producer to produce an item.

The code then defines the producer() and consumer() functions. The producer() function takes a buffer argument and repeatedly prompts the user to input an item to produce. It calls the produce() method of the buffer to add the item to the buffer. The consumer() function takes a buffer argument and repeatedly prompts the user to enter to consume an item. It calls the consume() method of the buffer to remove an item from the buffer.

Finally, the main() function prompts the user to input the size of the buffer. It creates a buffer object with the specified size and creates threads for the producer and consumer functions, passing the buffer object as an argument. It starts the threads, waits for them to finish, and exits the program.

To use this code, run the file and follow the prompts to enter the buffer size and produce and consume items. Use "exit" to quit producing items.

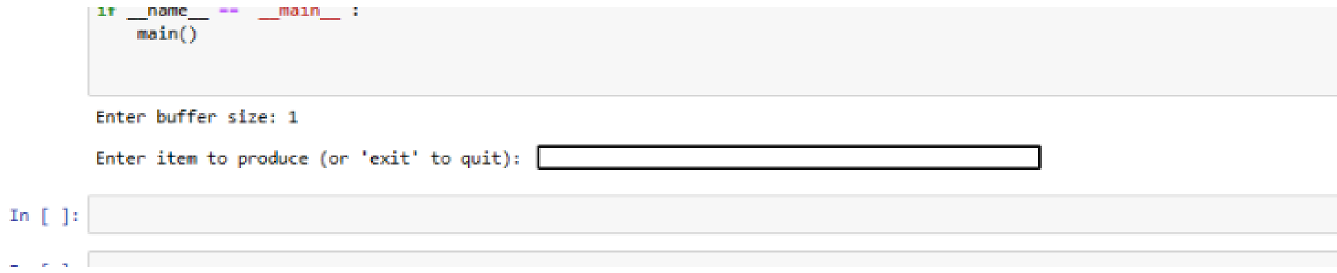


```
    consumer_thread = threading.Thread(target=consumer, args=(buffer,))

    producer_thread.start()
    consumer_thread.start()

    producer_thread.join()
    consumer_thread.join()

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

```
Enter buffer size:
```

```
[ ]:
```

The program prompts for the buffer size to be entered as shown above.

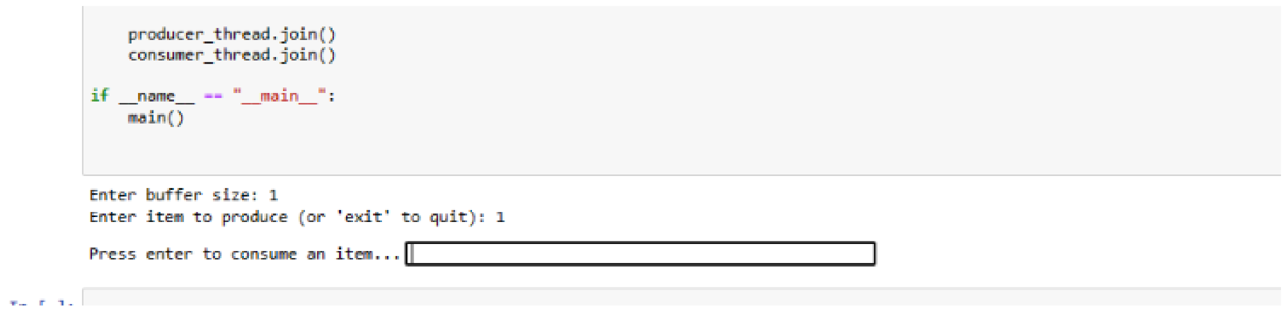


```
if __name__ == "__main__":
    main()
```

```
Enter buffer size: 1

Enter item to produce (or 'exit' to quit):
```

```
In [ ]:
```

Next, the program requires for the item to be produced to be entered as shown above.



```
    producer_thread.join()
    consumer_thread.join()

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

```
Enter buffer size: 1
Enter item to produce (or 'exit' to quit): 1

Press enter to consume an item...
```

It then prompts you to press enter so that it can consume an item as shown above.