Indiana Jones Coach Co.

Background

Indiana Jones started a small tourism business taking tourists around historic Harrisonburg, Virginia, USA. A number of tourists **N** arrive to Harrisonburg. Since Harrisonburg is so beautiful in the fall, each tourist will go on a shopping spree then ride the bus. Each tourist repeats the shopping-touring routine **T** times before leaving town. A tourist is either on the street shopping, on board the bus, or out of town altogether. The bus has only **R** seats, and Indiana will start a tour when either:

i. all seats are filled by tourists, or

ii. when all remaining tourists in town are already on the bus (i.e. no more tourists are expected to tour the city.)

Deliverable

Write a program for an operator thread that receives the values for **N** and **T** as command-line arguments, then creates **N+1** child threads representing:

- Indiana (the bus driver) plus
- the **N** tourists.

These threads follow the logic described on the next page. The operator must then wait until Indiana and the tourists have all terminated to clean up afterwards.

Use the srandom() and random() functions for random-number generation. Use usleep() to piut a thread into sleep.

Shared Objects and their initial values

```
semaphore  AvailSeats=0  , arrived=0 , busLoaded=0 ,
int onBoard = 0 ;                  // Number of tourists on board the bus
int shopping ;                     // count of tourists: still on the street, vs. those who started a tour on board the bus
int tickets ;                      // Total tickets for today = numTourists x trips-per-tourist
```

OPERATOR Thread

```
{
    Set up & initialize any shared objects ;
    Start the Driver() , and Tourist(1), Tourist(2), …. , Tourist(N) threads
    Wait for all threads to end then clean-up
}
```

TOURIST (*j*) Thread

```
tweet("Tourist < j >: Arrived" );
Notify Indiana;
REPEAT  T  times
{
    Tweet( "Tourist < j >: Going to shop") ;
    Sleep( 500 ≤ random ≤ 2500 mSec);          // Simulate shopping session
    Wait for an available seat on the bus;
    // update counters
    Tweet( "Tourist < j >: I got a seat on the bus") ;
    IF ( there are no more vacant seats OR NO other tourists are still shopping on the street
    THEN
        Alert Driver that bus is "as-full-as-possible"
    ENDIF
    Fasten seatbelt and inform Driver of that.
    Wait for the bus to actually move ;
    Tweet("Tourist < j >: The Wheels on the Bus go Round and Round!") ;
    Inform Driver my song is over ;
    Wait for the tour to finish ;
    Tweet("Tourist < j >: I got off the bus") ;
    IF I am last tourist to get off the bus
        Tweet("Tourist < j >: The bus is now vacant!") ;
        Inform Driver that this group of tourists got off the bus ;
    ENDIF
}
Tweet(" Tourist <j>: Leaving Town");
```

INDIANA Thread

```
Tweet("Driver: Started My Day")
Wait for all tourists to arrive to town.
REPEAT indefinitely
{
    IF ( all tourists left town )
        break out of this loop ;
    ENDIF
    Declare all the seats on the bus now available ;
    Take a nap until the bus is "as-full-as-possible";
    Tweet("Indy: Welcome on Board Everyone  <count> !") ;
    Wait for all tourists on board to fasten their seatbelts;
    Tweet("Indy: Thank you");
    duration = random: 1500 to 4000 mSec ;
    Tweet("Indy: Tour will last <duration> msec");
    Tweet("Indy: Bus is now moving. Sing Everyone!") ;
    Tweet("Indy: Bus! Bus! On the street! Who is the fastest driver to beat?");
    Inform all tourists on board that bus has moved ;
    Sleep( duration ) ;  // simulate driving the tour
    Wait for all tourists on board to finish their songs ;
    Tweet("Driver: Tour is over. Thanks you for Riding Indiana-Jones Coach") ;
    Inform all tourists on board that tour is finished ;
    Wait for last tourist of this group of tourists to get off the bus ;
}
Tweet ("I did <count> tours today")
```