The objective of this exercise is to help demonstrate and reinforce the concepts of Trunked Radio Systems. Trunking is used very heavily in the cellular system, emergency response (police, fire, and rescue), as well as certain military communication systems.

Part 1

The Matlab Function `simulate_erlang_b` provides a simple simulation of an Erlang B Trunked Radio System. The inputs to the function are:

```
avholding = Average Holding Time for a call in minutes
inarrival = Call Arrival Rate (i.e., Number of Calls per hour)
channel = Number of Trunked Radio Channels available to the system
nuser = Number of users to simulate
```

The function operates by simulating the effect of `<nuser>` number of users, each of which wish to access the system to place a `<avholding>` duration phone call. The `<inarrival>` rate determines how much time (on average) the system spends “idle” between when User A attempts to place a call and when User B attempts to place a call.

The function provides several outputs:

```
Theo_GoS = Theoretical Grade of Service from the Erlang B Table
Sim_GoS = Simulated Grade of Service
Traffic = Total Offered Traffic (in Erlangs) to the Trunked System
Num_Served = Number of Calls Served
Num_Blocked = Number of Calls Blocked
```

For Part 1, use the `simulate_erlang_b` function and re-create the Erlang B tables from the textbook, along with a comparison to the simulated performance. Simulate the system for 2, 3, 4, 5, and 10 Trunked Channels. In order to run your simulation, you should first recognize that the offered traffic to the system is a function of both `avholding` and `inarrival`. My suggestion is to fix one at a known value, and vary the other to achieve the desired system traffic in Erlangs. The `nuser` parameter will impact the accuracy of your simulation, the higher `nuser` is, the more accurate your simulation will be, but the longer it will take to run. My suggestion is to start with a small number, such as 500, and once you are convinced your code is correct, re-run the simulation for a larger number of users. Your final graph should look similar to the figure below:
**Part 2**

For Part 2, you are going to simulate the behavior of the Mobile Switching Center as it receives phone calls from users and needs to place them on a Trunked line. Simulate the system for 4 Trunked Lines and 40 users over a 2 hour time period. The following segment of code will take in the `<avholding>` average holding time and `<inarrival>` call arrival rate and produce an output matrix `call_matrix`, which contains two columns. The first column is the **absolute start time** (in minutes since the beginning of the simulation) that User X starts a phone call. The second column is the **absolute stop time** of User X’s phone conversation (subtract start from stop to get the duration).

**Note:** Units for `<avholding>` and `<inarrival>` need to match!

```matlab
avholding = ###; % Holding Time (hours)
inarrival = ####; % Arrival Rate (call/minutes)
nuser = ###; % Number of Users
Tmax = nuser*nuser; % maximum time (hours)
T(1)=random('Exponential',1/inarrival);
i=1;
while T(i) < Tmax,
    T(i+1)=T(i)+random('Exponential',1/inarrival);
    i=i+1;
end
T(i)=Tmax;
a = 60*exprnd(avholding,[nuser,1]); % Convert from hours to minutes
call_matrix = zeros(nuser,2); % Convert from hours to minutes
call_matrix(1,1) = T(1);
call_matrix(1,2) = T(1)+a(1);
for i=2:nuser
    call_matrix(i,1)=T(i);
    call_matrix(i,2)=call_matrix(i,1) + a(i);
end
call_matrix(:,2) = call_matrix(:,2)-call_matrix(1,1);
call_matrix(:,1) = call_matrix(:,1)-call_matrix(1,1);
```

An example output of `call_matrix` is given below:

```matlab
call_matrix =
    8.6024 28.0821
    8.6778 43.0005
   12.9782 74.2554
```
This tells me that User 1 places a phone call 8.60 minutes into the simulation, and that his call lasts for 19.48 minutes. User 2 places a phone call 8.67 minutes into the simulation, and that his call lasts for 34.32 minutes. Your objective is to build up a “system usage plot” that illustrates how the system is utilized for two cases: lightly loaded (few or no calls blocked), and heavily loaded (many calls blocked). Your chart should look similar to the following:

Next, keeping the total offered traffic the same (i.e., holding “A” constant), vary the call arrival rate and holding time and re-create the system usage plot. Compare the system usage plots for the various scenarios – notice that the “white space” in your plot represents inefficiency in the Trunked Radio System.

**Interview Grading**

Be prepared to show and demonstrate the following:

- Erlang B Theoretical vs. Simulated Curves (generate the figure and discuss how your code works)
- System Usage Plot for at least 3 configurations of inter-arrival rates and holding times.
- What (if anything) changes for an Erlang C (blocked calls delayed) system.
- Ideas for improving the efficiency of your Trunked Radio System.