# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

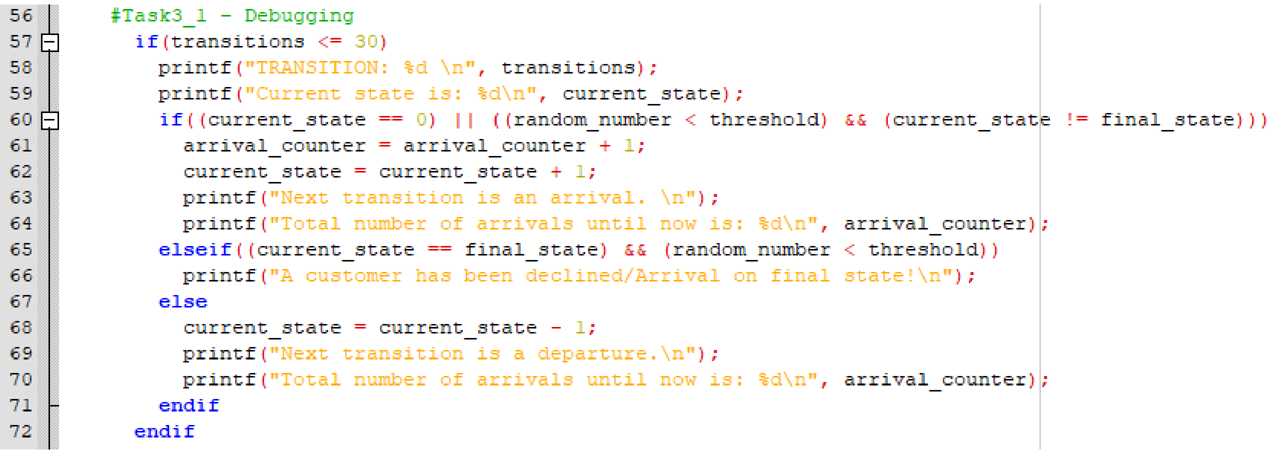
*Ονοματεπώνυμο: Γκούμε Λαουρεντιάν*
*Α.Μ.: 031 18 014*
*Έτος/Εξάμηνο: 3ο/6ο*

# ΣΥΣΤΗΜΑΤΑ ΑΝΑΜΟΝΗΣ (Queuing Systems)

## 3η Σειρά Ασκήσεων

## Προσομοίωση συστήματος M/M/1/10:

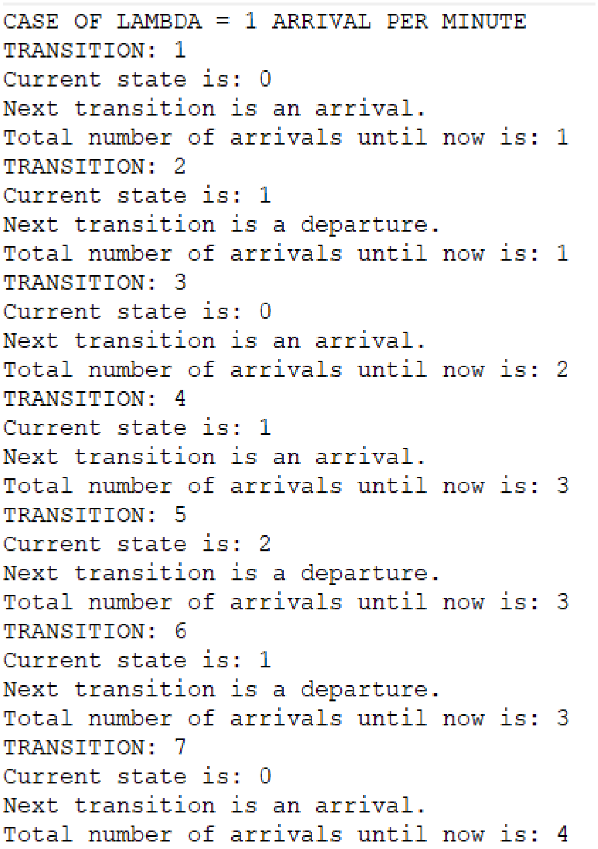
**(1)** Το τμήμα που χρησιμοποιήθηκε για το **debugging** παρουσιάζεται ως πολλαπλά σχόλια παρακάτω:



```
    #Task3_1 - Debugging
      if(transitions <= 30)
        printf("TRANSITION: %d \n", transitions);
        printf("Current state is: %d\n", current_state);
        if((current_state == 0) || ((random_number < threshold) && (current_state != final_state)))
          arrival_counter = arrival_counter + 1;
          current_state = current_state + 1;
          printf("Next transition is an arrival. \n");
          printf("Total number of arrivals until now is: %d\n", arrival_counter);
        elseif((current_state == final_state) && (random_number < threshold))
          printf("A customer has been declined/Arrival on final state!\n");
        else
          current_state = current_state - 1;
          printf("Next transition is a departure.\n");
          printf("Total number of arrivals until now is: %d\n", arrival_counter);
        endif
      endif
```

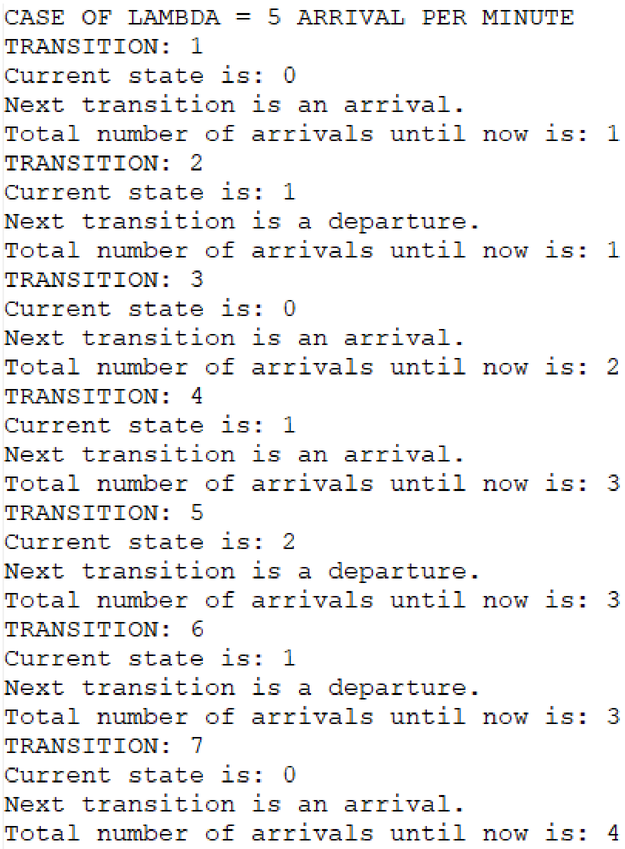
*Εικόνα 1: Τμήμα κώδικα για debugging*

Ενδεικτικά, παρατίθενται οι πρώτες 7 μεταβάσεις για κάθε λ:



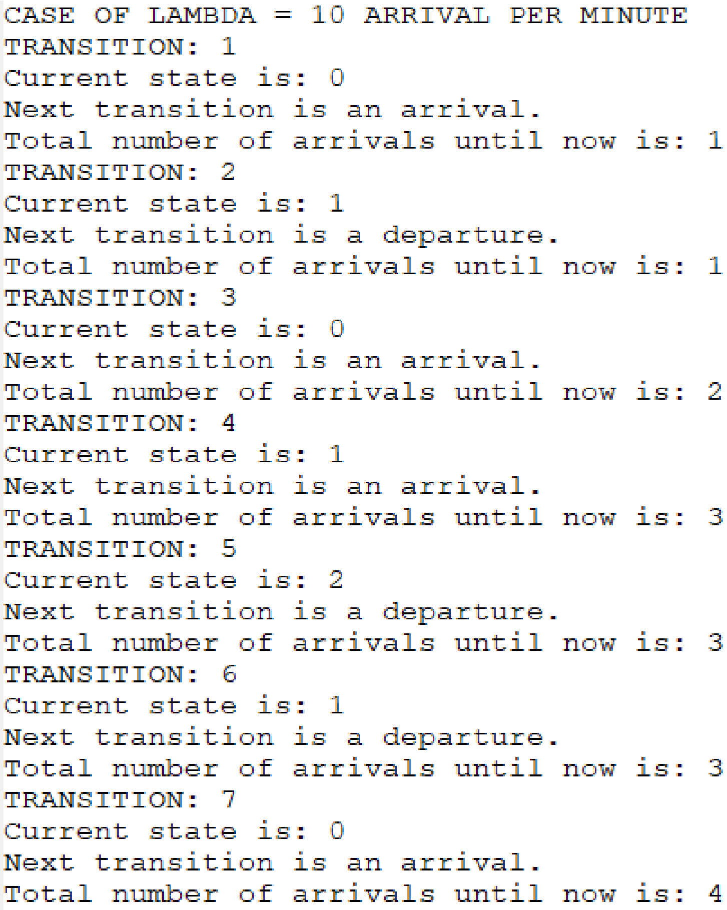
```
CASE OF LAMBDA = 1 ARRIVAL PER MINUTE
TRANSITION: 1
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 1
TRANSITION: 2
Current state is: 1
Next transition is a departure.
Total number of arrivals until now is: 1
TRANSITION: 3
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 2
TRANSITION: 4
Current state is: 1
Next transition is an arrival.
Total number of arrivals until now is: 3
TRANSITION: 5
Current state is: 2
Next transition is a departure.
Total number of arrivals until now is: 3
TRANSITION: 6
Current state is: 1
Next transition is a departure.
Total number of arrivals until now is: 3
TRANSITION: 7
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 4
```

*Εικόνα 2: Αποτελέσματα 7 πρώτων μεταβάσεων για λ = 1*



```
CASE OF LAMBDA = 5 ARRIVAL PER MINUTE
TRANSITION: 1
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 1
TRANSITION: 2
Current state is: 1
Next transition is a departure.
Total number of arrivals until now is: 1
TRANSITION: 3
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 2
TRANSITION: 4
Current state is: 1
Next transition is an arrival.
Total number of arrivals until now is: 3
TRANSITION: 5
Current state is: 2
Next transition is a departure.
Total number of arrivals until now is: 3
TRANSITION: 6
Current state is: 1
Next transition is a departure.
Total number of arrivals until now is: 3
TRANSITION: 7
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 4
```

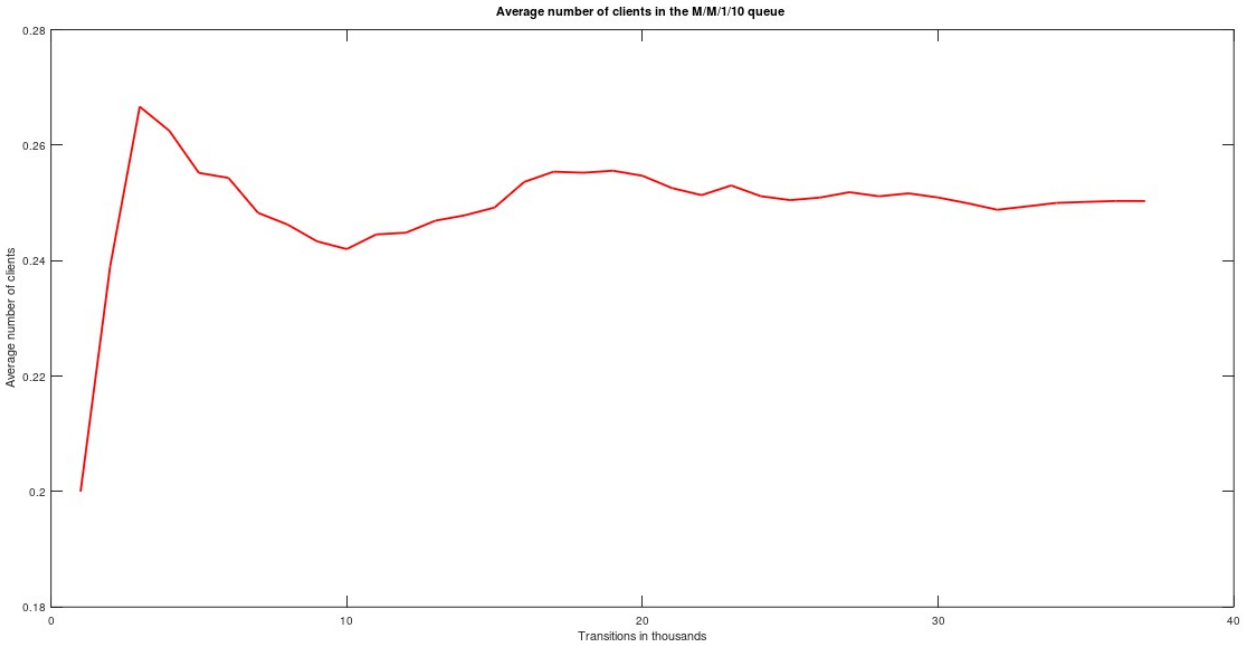
*Εικόνα 3: Αποτελέσματα 7 πρώτων μεταβάσεων για λ = 5*



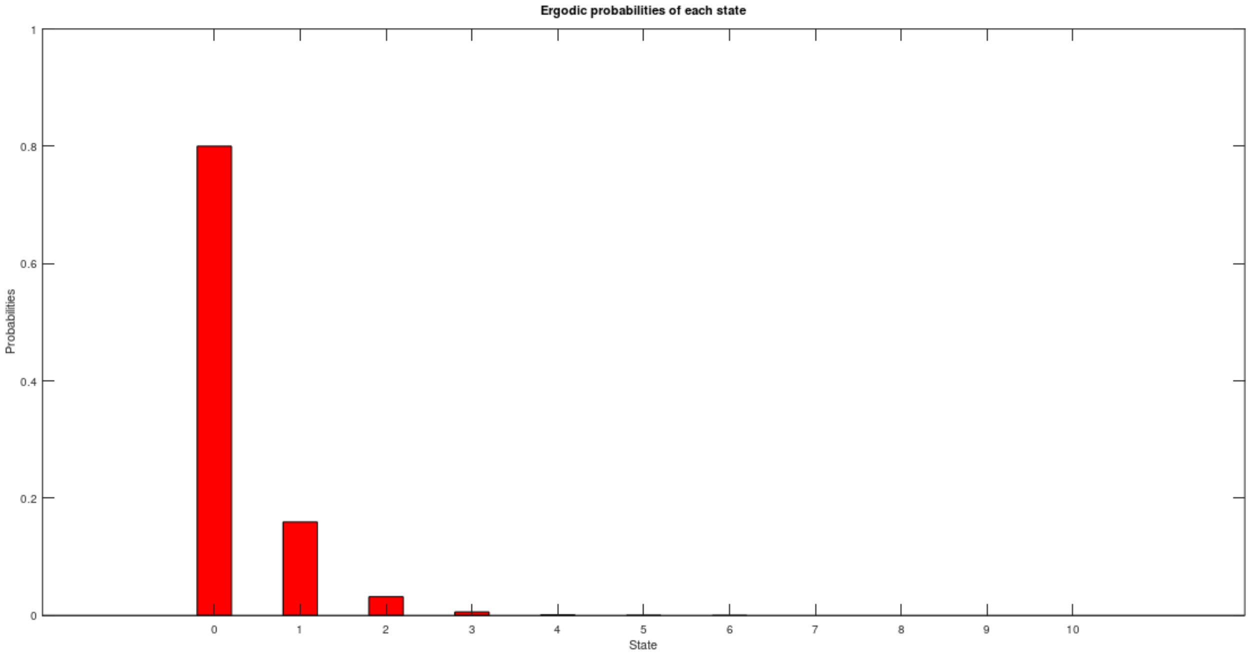
```
CASE OF LAMBDA = 10 ARRIVAL PER MINUTE
TRANSITION: 1
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 1
TRANSITION: 2
Current state is: 1
Next transition is a departure.
Total number of arrivals until now is: 1
TRANSITION: 3
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 2
TRANSITION: 4
Current state is: 1
Next transition is an arrival.
Total number of arrivals until now is: 3
TRANSITION: 5
Current state is: 2
Next transition is a departure.
Total number of arrivals until now is: 3
TRANSITION: 6
Current state is: 1
Next transition is a departure.
Total number of arrivals until now is: 3
TRANSITION: 7
Current state is: 0
Next transition is an arrival.
Total number of arrivals until now is: 4
```

*Εικόνα 4: Αποτελέσματα 7 πρώτων μεταβάσεων για λ = 10*

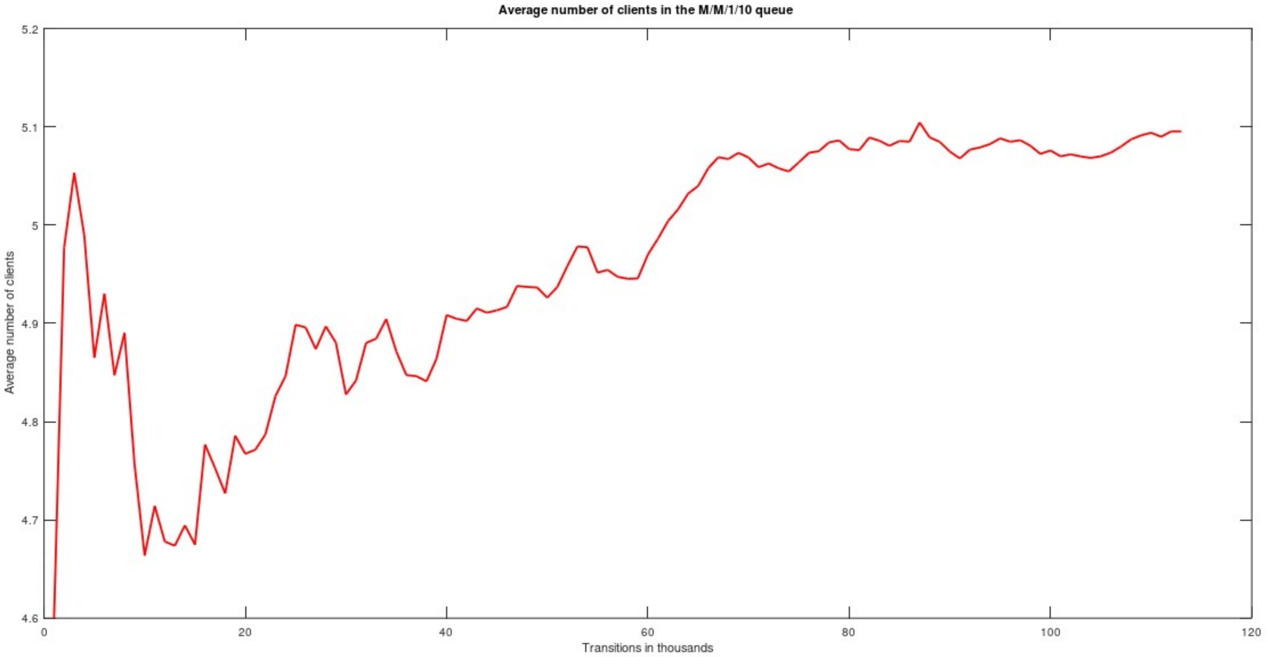
**(2)** Η εκτέλεση της προσομοίωσης για ρυθμούς άφιξης **λ = {1, 5, 10}** μας δίνει τα παρακάτω αποτελέσματα:



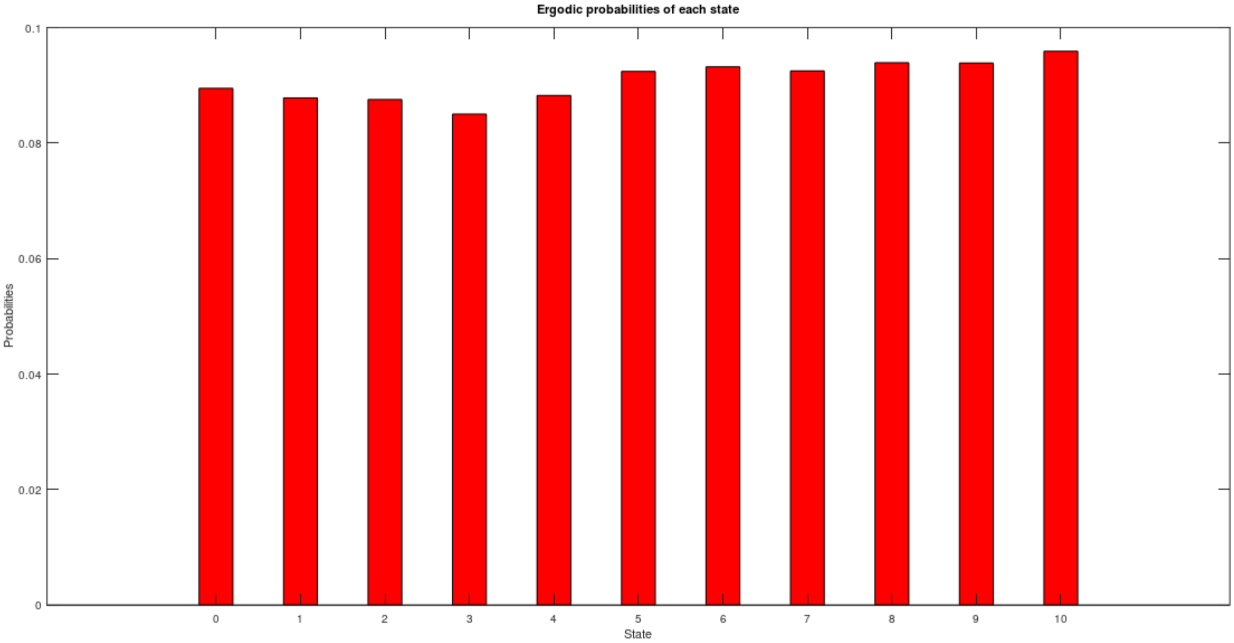
*Εικόνα 5: Εξέλιξη μέσου αριθμού πελατών στο σύστημα για λ = 1*



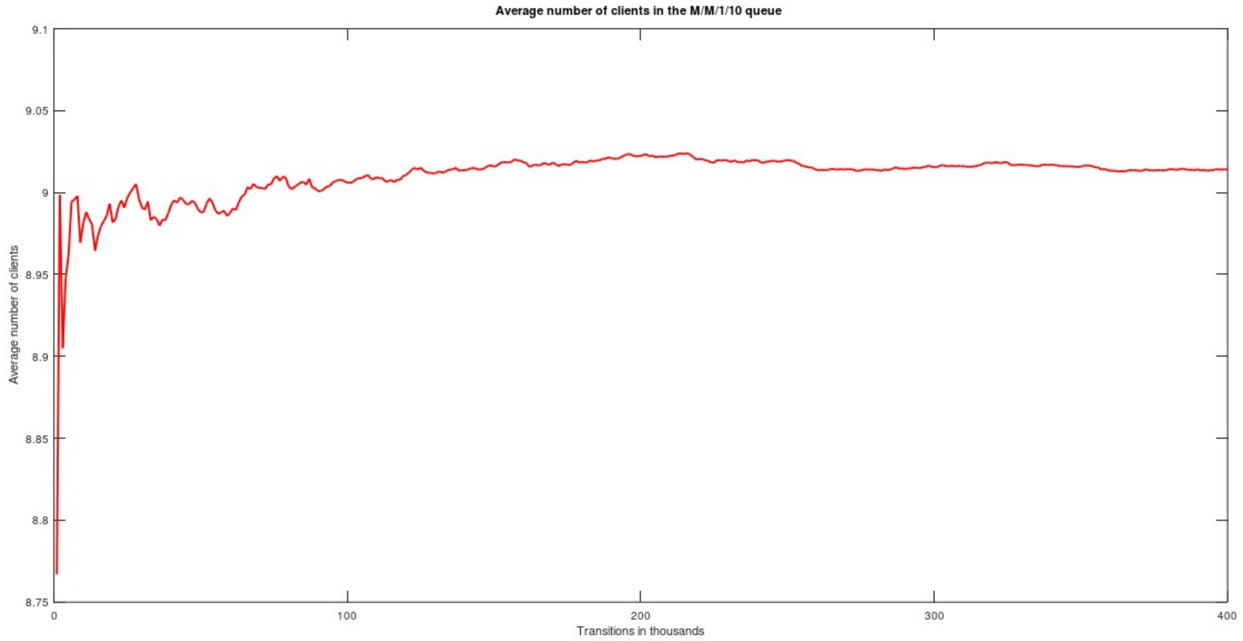
*Εικόνα 6: Εργοδικές πιθανότητες καταστάσεων για λ = 1*



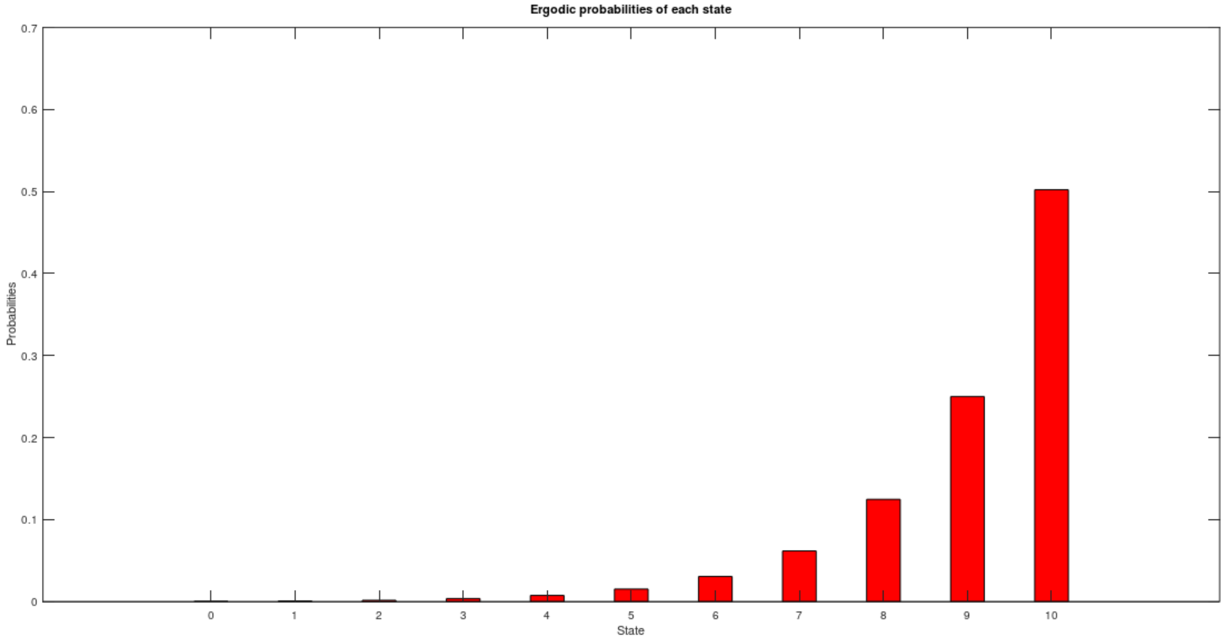
*Εικόνα 7: Εξέλιξη μέσου αριθμού πελατών στο σύστημα για λ = 5*



*Εικόνα 8: Εργοδικές πιθανότητες καταστάσεων για λ = 5*

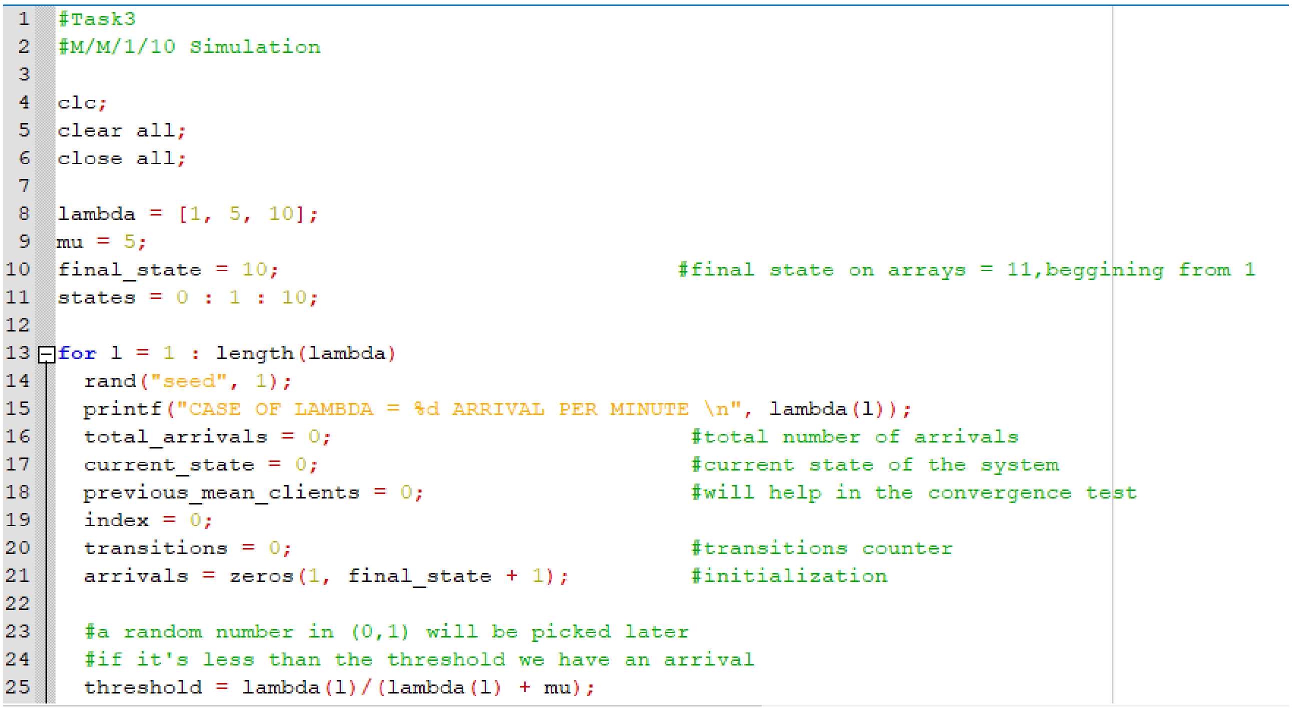


*Εικόνα 9: Εξέλιξη μέσου αριθμού πελατών στο σύστημα για λ = 10*

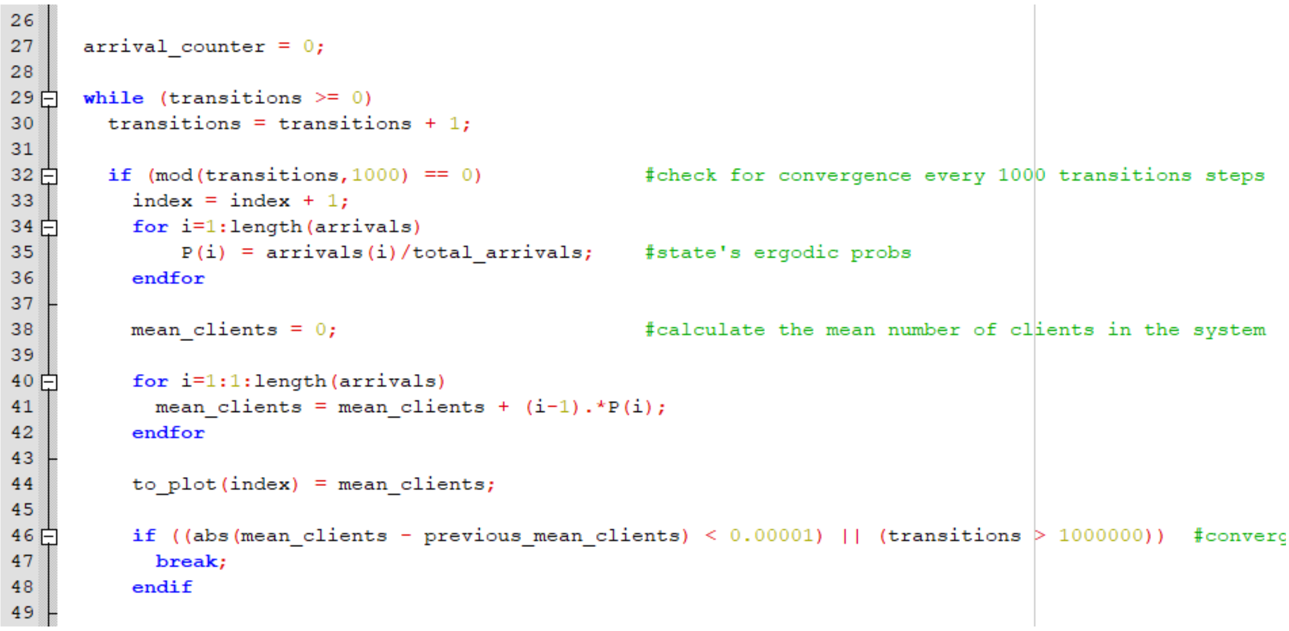


*Εικόνα 10: Εργοδικές πιθανότητες καταστάσεων για λ = 10*

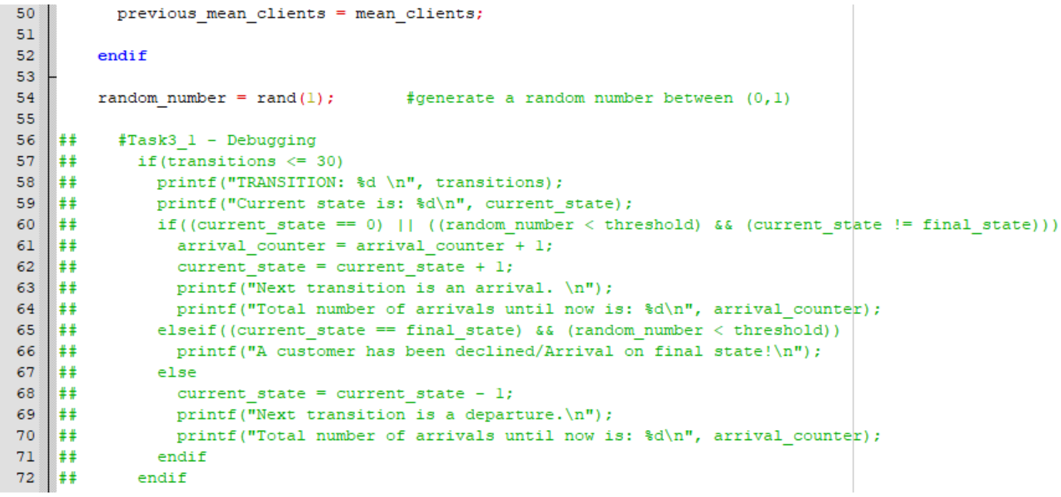
Ο κώδικας που χρησιμοποιήθηκε για να παραχθούν τα παραπάνω αποτελέσματα:



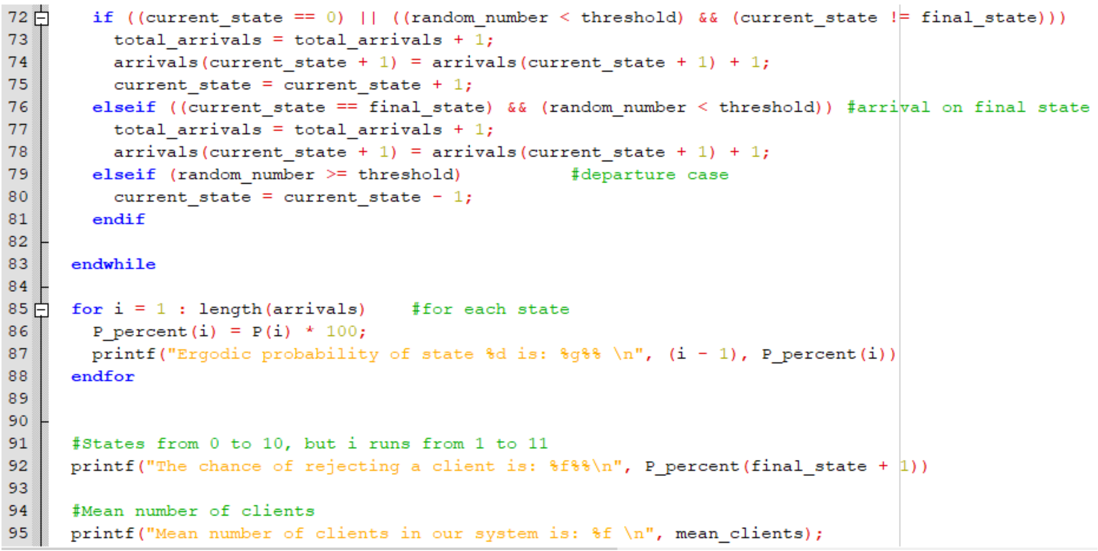
```
#Task3
#M/M/1/10 Simulation

clc;
clear all;
close all;

lambda = [1, 5, 10];
mu = 5;
final_state = 10;                               #final state on arrays = 11,beggining from 1
states = 0 : 1 : 10;

for l = 1 : length(lambda)
  rand("seed", 1);
  printf("CASE OF LAMBDA = %d ARRIVAL PER MINUTE \n", lambda(l));
  total_arrivals = 0;                             #total number of arrivals
  current_state = 0;                              #current state of the system
  previous_mean_clients = 0;                      #will help in the convergence test
  index = 0;
  transitions = 0;                                #transitions counter
  arrivals = zeros(1, final_state + 1);           #initialization

  #a random number in (0,1) will be picked later
  #if it's less than the threshold we have an arrival
  threshold = lambda(l)/(lambda(l) + mu);
```



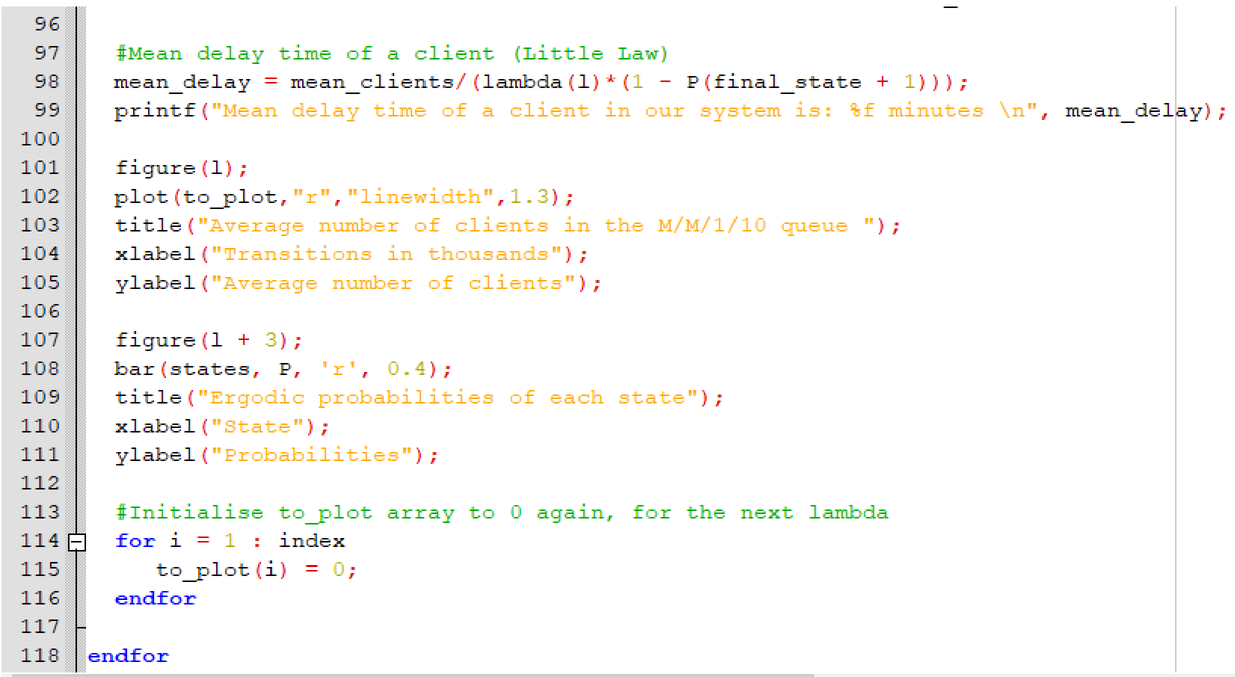
```

  arrival_counter = 0;

  while (transitions >= 0)
    transitions = transitions + 1;

    if (mod(transitions,1000) == 0)             #check for convergence every 1000 transitions steps
      index = index + 1;
      for i=1:length(arrivals)
          P(i) = arrivals(i)/total_arrivals;      #state's ergodic probs
      endfor

      mean_clients = 0;                           #calculate the mean number of clients in the system

      for i=1:1:length(arrivals)
        mean_clients = mean_clients + (i-1).*P(i);
      endfor

      to_plot(index) = mean_clients;

      if ((abs(mean_clients - previous_mean_clients) < 0.00001) || (transitions > 1000000))  #converg
        break;
      endif

```



```
      previous_mean_clients = mean_clients;

    endif

    random_number = rand(1);       #generate a random number between (0,1)

##    #Task3_1 - Debugging
##      if(transitions <= 30)
##        printf("TRANSITION: %d \n", transitions);
##        printf("Current state is: %d\n", current_state);
##        if((current_state == 0) || ((random_number < threshold) && (current_state != final_state)))
##          arrival_counter = arrival_counter + 1;
##          current_state = current_state + 1;
##          printf("Next transition is an arrival. \n");
##          printf("Total number of arrivals until now is: %d\n", arrival_counter);
##        elseif((current_state == final_state) && (random_number < threshold))
##          printf("A customer has been declined/Arrival on final state!\n");
##        else
##          current_state = current_state - 1;
##          printf("Next transition is a departure.\n");
##          printf("Total number of arrivals until now is: %d\n", arrival_counter);
##        endif
##      endif
```



```
    if ((current_state == 0) || ((random_number < threshold) && (current_state != final_state)))
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
      current_state = current_state + 1;
    elseif ((current_state == final_state) && (random_number < threshold)) #arrival on final state
      total_arrivals = total_arrivals + 1;
      arrivals(current_state + 1) = arrivals(current_state + 1) + 1;
    elseif (random_number >= threshold)          #departure case
      current_state = current_state - 1;
    endif

  endwhile

  for i = 1 : length(arrivals)     #for each state
    P_percent(i) = P(i) * 100;
    printf("Ergodic probability of state %d is: %g%% \n", (i - 1), P_percent(i))
  endfor


  #States from 0 to 10, but i runs from 1 to 11
  printf("The chance of rejecting a client is: %f%%\n", P_percent(final_state + 1))

  #Mean number of clients
  printf("Mean number of clients in our system is: %f \n", mean_clients);
```



```

  #Mean delay time of a client (Little Law)
  mean_delay = mean_clients/(lambda(l)*(1 - P(final_state + 1)));
  printf("Mean delay time of a client in our system is: %f minutes \n", mean_delay);

  figure(l);
  plot(to_plot,"r","linewidth",1.3);
  title("Average number of clients in the M/M/1/10 queue ");
  xlabel("Transitions in thousands");
  ylabel("Average number of clients");

  figure(l + 3);
  bar(states, P, 'r', 0.4);
  title("Ergodic probabilities of each state");
  xlabel("State");
  ylabel("Probabilities");

  #Initialise to_plot array to 0 again, for the next lambda
  for i = 1 : index
     to_plot(i) = 0;
  endfor

endfor
```

*Εικόνα 11: Ο κώδικας που υλοποιεί τα ζητούμενα*

**(3)** Παρατηρούμε, ποιοτικά, πως **αύξηση του λ συνεπάγεται παράλληλα αύξηση του απαιτούμενου αριθμού μεταβάσεων για σύγκλιση** στην εργοδική κατάσταση. Βασιζόμενοι στις ανωτέρω γραφικές παραστάσεις, μπορούμε να πούμε πως τα μεταβατικά φαινόμενα εξασθενούν μετά τα πρώτα 10/60/100 χιλιάδες transitions, επομένως θα μπορούσαμε να αγνοήσουμε τις μεταβάσεις αυτές προκειμένου να επιτύχουμε γρηγορότερη σύγκλιση της προσομοίωσης.

**(4)** Αν το μ ήταν μεταβλητό, αρχικά θα το ορίζαμε ως πίνακα: μ = [2μ, 3μ, 4μ, 5μ, 6μ, 7μ, 8μ, 9μ, 10μ, 11μ]. Απόρροια αυτού, θα ήταν να οριστεί στη συνέχεια το threshold ως πίνακας για κάθε κατάσταση (από 1 έως 10). Οι 2 αυτές αλλαγές, θα κάλυπταν το ενδεχόμενο μεταβλητών ρυθμών εξυπηρετήσεως.