```
clear;close all; clc ; rng('default');
% Entities: Customer
% Resources:
%              1- Cashier (Server: can have a queue)
%              2- Salad Bar (Server: can have a queue)
%              3- Soup Bar (Server: can have a queue)
% Processes:
%              1- Paying
%              2- Getting Salad
%              3- Getting Soup
max_customers = 2000;
CashierAvailTime = [0];
% Start First Phase
for i = 1 : max_customers % cashier loop
    IAT = normrnd(1.0,0.2); % Inter-Arrival Time
    if i >1
            AT(i) = IAT + AT(i-1); % Arrival Time of i
    else
            AT(1) = IAT; % Arrival Time of 1st Customer
    end % end if
    [Start_Service_C(i), Srv_N] = Sieze(AT(i),CashierAvailTime);
    Service_Time_C = normrnd(4/6,1/6);
    Completion_C(i) = Start_Service_C(i)+Service_Time_C;
    CashierAvailTime = ReleaseRes(CashierAvailTime,1,Completion_C(i));
    T_C(i) = Completion_C(i) - AT(i); % Time Spent in Casier Sub-System
end % end cashier loop
Avg_T_C = mean(T_C);
Completion_C = sort(Completion_C);% sorts the elements
% End First Phase

% Start Second Phase
Arr_SALAD = Completion_C;
SaladAvailTimeArray = [0,0,0,0] ;
for i = 1 : max_customers % salad loop
   [Start_Service_Salad(i),Srv_N] =...
       Sieze(Arr_SALAD(i),SaladAvailTimeArray);
    Service_Time_Salad = normrnd(2,1/3);
    Completion_Salad(i) = Start_Service_Salad(i)+Service_Time_Salad;
    SaladAvailTimeArray = ...
        ReleaseRes(SaladAvailTimeArray,Srv_N,Completion_Salad(i));
    T_Salad(i) = Completion_Salad(i) - Arr_SALAD(i);
```

```
 end % end salad loop
 Avg_T_Salad = mean(T_Salad);
 Completion_Salad = sort(Completion_Salad);% sorts the elements
 % End Second Phase

 % Branching Before third Phase
 n_Salad_Soup = 0; % number of customers having both
 for i = 1 : max_customers
     if rand <0.6
         n_Salad_Soup = n_Salad_Soup + 1;
         Arr_SOUP(n_Salad_Soup) = Completion_Salad(i);
     end % end if
 end % end for

 % Start Third Phase
SoupAvailTimeArray = [0,0] ;
for i = 1 : n_Salad_Soup % soup loop
    [Start_Service_Soup(i),Srv_N] = ...
        Sieze(Arr_SOUP(i),SoupAvailTimeArray);
    Service_Time_Salad = normrnd(1,1/4);
    Completion_Soup(i) = Start_Service_Soup(i)+Service_Time_Salad;
    SoupAvailTimeArray = ...
        ReleaseRes(SoupAvailTimeArray,Srv_N,Completion_Soup(i));
    T_Soup(i) = Completion_Soup(i) - Arr_SOUP(i);
 end % end soup loop
 Avg_T_Soup = mean(T_Soup);
 % End Third Phase

  fprintf('Average time a customer spends paying & getting food');
  fprintf('\n if getting salad only is:%4.2f',Avg_T_C+Avg_T_Salad);
  fprintf('\n if getting both is:%4.2f',Avg_T_C+Avg_T_Salad+Avg_T_Soup);
```