```
%summer(HBC)
nEV=input('Enter the No of Vehicles: ');
Array_BEV_40PHEV_30PHEV = zeros(1,nEV); % elements 1=>BHEVs 2=>BEVs

%........ BEV=>20%=>'1'____PHEV-40=>30%=>'2'____PHEV-30=>50%=>'3'[HBC]
    for i=1:nEV
       if i<=round(nEV*50/100)
           Array_BEV_40PHEV_30PHEV(i)=1;
       elseif i>round(nEV*50/100) && i<=round(nEV*80/100)
           Array_BEV_40PHEV_30PHEV(i)=2;
       elseif i>round(nEV*80/100) && i<=nEV
           Array_BEV_40PHEV_30PHEV(i)=3;
       end
    end


type=Array_BEV_40PHEV_30PHEV(randperm(length(Array_BEV_40PHEV_30PHEV)));
%Shuffle-> To indicate each element of Array

DM = normrnd(55,10,[1,nEV]);  % Daily Mileage following Normal Distribution
                              % Mean=55, S.D.=10, No of Random Variables
supposed to be generated=10
FTD = normrnd(18,8.41,[1,nEV]);  % 1st Trip Distance following Normal
Distribution

Tin = normrnd(16,1.2,[1,nEV]);  % Arrival Time following Normal Distribution
Tout = normrnd(22,1.2,[1,nEV]); % Departure Time following Normal
Distribution

while max(Tout)>24    % Tout shouldn't go beyond 24hrs
    Tout = normrnd(22,1.2,[1,nEV]);  %if goes, again generates Random
Variable
end

rate_USEP_MWh1 = [139.16 129.13 117.72 110.30 105.87 104.99 105.09 104.45
100.18 100.08 104.97 105.01 105.78 113.14 128.95 139.27 151.80 200.78 209.43
169.81 164.36 151.76 146.39 138.87 128.92 139.29 139.36 92.30 101.59 85.31
133.54 142.81 142.81 145.61 151.70 158.79 175.52 208.34 306.47 306.48 306.43
306.50 236.37 184.81 151.65 142.72 139.49 139.40];
rate_USEP_MWh = zeros(nEV,48);
```

```
for i=1:nEV
    for j=1:48
         rate_USEP_MWh(i,j)=rate_USEP_MWh1(j);
    end
end

rate_USEP_kWh = rate_USEP_MWh/1000;
%type1 = rate_USEP_kWh(randperm(length(rate_USEP_kWh)));

for i=1:nEV
    % <--------------1st Objective: To Find ENERGY REQ.(of all individual
BHEVs & BEVs)---------------------->
    if type(i)==1  %BEV
        AER=117;
        Bc=24;
    elseif type(i)==2  %PHEV-40
        AER=40*1.6;
        Bc=18.4;
    elseif type(i)==3   %PHEV-30
        AER=30*1.6;
        Bc=13.8;
    end

    STD(i)=DM(i)-FTD(i);
    SOCa(i)=1-FTD(i)/AER;
    SOCd(i)= STD(i)/AER+0.2;
    if SOCd(i)>1
        SOCreq(i)=1-SOCa(i);
    elseif SOCa(i)<SOCd(i) && SOCd(i)<1
        SOCreq(i)=SOCd(i)-SOCa(i);
    elseif SOCd(i)==SOCa(i)
        SOCreq(i)=0;
    elseif SOCd(i)>0.2 && SOCd(i)<SOCa(i)
        SOCreq(i)=-(SOCa(i)-SOCd(i));
    end
    Ereq(i)=SOCreq(i)*Bc/0.9;    % Energy req while Charging----------------
--------

    % we can use 'disp(Ereq(i));' as well to print values
```

```
% <------------2nd Objective: To find TOTAL NO OF SLOTS.----------------->

    if Tin(i)>round(Tin(i))
        Nin(i)=floor(Tin(i))*2+1;
    elseif Tin(i)<round(Tin(i))
        Nin(i)=floor(Tin(i))*2+2;
    end

    if Tout(i)>round(Tout(i))
        Nout(i)=floor(Tout(i))*2+1;
    elseif Tout(i)<round(Tout(i))
        Nout(i)=floor(Tout(i))*2+2;
    end

    Nslot(i)=Nout(i)-Nin(i)+1;  % No of Slots----------------------->


end

cost = zeros(nEV,48);
for i=1:nEV
    fprintf('\n\n\n _____________________Vehicle-%d_____________________\n',i);
    if Ereq(i)<0
        Ereq(i)=Ereq(i)*0.9*0.9; % Energy req while Discharging------------
-----------
    end
    fprintf('Energy Req for Vehicle-%d = %.7f\n',i,Ereq(i));%Print Ereq
Values as Output
    fprintf('Entering Slot of V-%d = %d\n\n',i,Nin(i));
    fprintf('Outgoing Slot of V-%d = %d\n\n',i,Nout(i));
    fprintf('No of Slots for Vehicle-%d = %d\n\n',i,Nslot(i));

    % <------------3rd Objective: Last Slot Adjustment.----------------->
    m=Ereq(i); %----store the value of Energy req in variale 'm'
    condition=true;
     array = zeros(1, Nslot(i));  %.................. to indicate amount of
charging/discharging
     arrayCHorDIS = zeros(1, Nslot(i));  %......... to indicate whether
charging(1)/discharging(-1)/idle(0)
```

```
    while condition
        total_power = 0; % Initially charged power

    % Loop upto (n-1) slots
        j=0;
        for slot = Nin(i):Nout(i)-1
            j=j+1;
            action = randi([1, 3]);   % 1: charge, 2: discharge, 3: idle

         switch action
            case 1    %---------Charging
                c1=3;  %-------Charging range: 3 to 8kW
                c2=8;
                charge_power = c1+(c2-c1)*rand;  %---------- random
generation
                total_power = total_power + charge_power;
                cost(i,slot)= charge_power;
                array(j)= charge_power;
                arrayCHorDIS(j)=1;   %........charging(1)
                %fprintf('Slot-%d -----> Charging: %f kW.  Total Charged
Power: %.2f kW\n', slot, charge_power, total_power);

            case 2    %.........Discharging
                d1=0;  %-------Discharging range: 0 to 2kW
                d2=2;
                discharge_power = d1+(d2-d1)*rand;  %---------- random
generation
                total_power = total_power - discharge_power;
                cost(i,slot)= -discharge_power;
                array(j)= -discharge_power;
                arrayCHorDIS(j)=-1;   %........Discharging(-1)
                %fprintf('Slot %d -----> Discharging %f kW.  Total Charged
Power: %.2f kW\n', slot, discharge_power, total_power);

             case 3    % Idle
                 array(j)= 0;
                 arrayCHorDIS(j)=0;
                %fprintf('Slot %d -----> Idle.  Total Power: %.2f kW\n',
slot, total_power);
```

```
        end
       end       % End operation of (n-1)th slot

 % nth(last slot) calculation & Range
checking_____________________________________________

    if total_power > m    % DISCHARGING NEEDED
        final_discharged_power = total_power-m;
        if (final_discharged_power>2)
            %fprintf('\n.....NEW SCHEDULE.....\n');
        elseif (final_discharged_power>=1 && final_discharged_power<=2)
             k=0;
             tp=0;
            for slot=Nin(i):Nout(i)-1
                k=k+1;
                if  arrayCHorDIS(k)==1
                     fprintf('Slot-%d -----> Charging: %f kW.  Total Charged
Power: %.2f kW\n', slot, array(k), tp+array(k));
                     tp=tp+array(k);
                elseif arrayCHorDIS(k)==-1
                    fprintf('Slot %d -----> Discharging %f kW.  Total
Charged Power: %.2f kW\n', slot, array(k), tp+array(k));
                    tp=tp+array(k);
                elseif arrayCHorDIS(k)==0
                    fprintf('Slot %d -----> Idle.  Total Power: %.2f kW\n',
slot, tp);
                end
            end
            fprintf('Slot %d(last slot) ----->Need of Discharging,
Discharged Power = %f',Nout(i),final_discharged_power);
            fprintf('\n   So, finally Total Energy Charged = %f kWh\n\n',
(total_power-final_discharged_power));
            cost(i,Nout(i))= -final_discharged_power;
            condition = false;
        end

    elseif total_power<m  % CHARGING NEEDED
        final_charged_power = m-total_power;
        if (final_discharged_power>8 && final_discharged_power<3)
        %fprintf('\n.....NEW SCHEDULE.....\n');
```

```
        elseif (final_charged_power>=3 && final_charged_power<=8)
 % for printing 1st to n-1 slot detals..............
             k=0;
             tp=0;
            for slot=Nin(i):Nout(i)-1
                k=k+1;
                if  arrayCHorDIS(k)==1
                     fprintf('Slot-%d -----> Charging: %f kWh.  Total
Charged Energy: %.2f kWh\n', slot, array(k), tp+array(k));
                     tp=tp+array(k);
                elseif arrayCHorDIS(k)==-1
                    fprintf('Slot %d -----> Discharging %f kWh.  Total
Charged Energy: %.2f kWh\n', slot, array(k), tp+array(k));
                    tp=tp+array(k);
                elseif arrayCHorDIS(k)==0
                    fprintf('Slot %d -----> Idle.  Total Energy: %.2f
kWh\n', slot, tp);
                end
            end

% for printing last slot details..................
            fprintf('Slot %d(last slot) ----->Need of Charging, Charged
Energy = %f kWh',Nout(i),final_charged_power);
            fprintf('\n   So, finally Total Energy charged = %f kWh\n\n',
(total_power+final_charged_power));
            cost(i,Nout(i))= final_charged_power;
            condition = false;
        end
    end
end



end

cost=cost.*rate_USEP_kWh;
disp(cost);

for i=1:nEV
    sum=0;
    for j=1:48
```

```
        sum=sum+cost(i,j);
    end
    fprintf('\n\n Total Cost for Vehicle-%d = %f',i,sum);
end
```