USER MANUAL

1. Enter the number of species present in the system.
    a. Example: Binary – 2, Ternary – 3, Quaternary – 4
    b. NOTE: The code only works for multicomponent gas/vapor systems. It will not work for systems with only one species.
2. Enter the vapor compositions of each species in the mixture.
3. Enter the temperature of the system in KELVIN.
4. Enter the pressure of the system in BAR.
5. Enter the critical properties of each species. The program will ask for the following properties:
    a. Critical Temperature (T_c) in KELVIN
    b. Critical Pressure (P_c) in BAR
    c. Acentric Factor (ω)
6. After all of these are entered, the program automatically computes and shows the values of the following parameters:
    a. Mixture Compressibility Factor (Z)
    b. Fugacity Coefficients ($\widehat{\emptyset}_i^v$) of each individual species in the mixture

NOTE: Please input NUMBERS only. The code will error if there are non-numerical symbols entered.

WRITE-UP

The prepared programs compute for the mixture compressibility factor (Z) and the fugacity coefficients ($\widehat{\emptyset}_i^v$) of each individual species in a mixture system using Cubic Equations of State (CEoS). Four (4) separate programs were prepared for each type of CEoS namely, van der Waals, Redlich-Kwong, Soave-Redlich-Kwong, and Peng-Robinson.

Despite their differences, most of these programs' codes contain similarities. Throughout the paper, similar codes will be generally explained while distinct ones will have their own exhibits.

Step 1: Code to Input Required Property Values.

```
% LPA1 Specification 2: Cubic Equation of State
% Peng-Robinson Equation of State (PR EoS)
% Multi-component Gas/Vapor System
%
% Definition:
%   y = vapor composition
%   press = total pressure of the system
%   temp = total temperature of the system
%   pressc = critical pressure of each species in the sy
%   tempc = critical temperature of each species in the sy
%   w = acentric factor
%   z = mixture compressibility factor
%   phi = fugacity coefficients of each species in the sy
%   fhat = fugacity of each species in the system

clc
clear

% Input the following variables
  n = input('Enter no. of species in the mixture: ');
  for b = 1:n
    y(1,b)=input(['Enter vapor composition (y) of species ' num2str(b) ': ']);
  end
  temp = input('Enter Temperature (in K): ');
  press = input('Enter Pressure (in bar): ');
  fprintf('\n');
  tempc = zeros(1,n);
  pressc = zeros(1,n);
  w = zeros(1,n);
  for b = 1:n
      tempc(1,b) = input(['Enter critical temperature (Tc) of species ' num2str(b) ': ']);
      pressc(1,b) = input(['Enter critical pressure (Pc) of species ' num2str(b) ': ']);
      w(1,b) = input(['Enter acentric factor (w) of species ' num2str(b) ': ']);
  end
```

This part of the program is similar with that of the other CEoS. At this section, we define a code that would enable us to input and store the necessary property values to be used for the succeeding calculations.

This part of the code prompts the user to input the required property values. This includes the no. of components in the system, vapor compositions, system temperature and pressure, and critical properties for each species (n, y_i, T, P, $T_{c,i}$, $P_{c,i}$, and ω).

This part creates an array for each critical property where the input values can be stored.

This code is looped to ask for each critical property n no. of times, where n is the number of components in the mixture.

Sample Result for the Input.

```
Enter no. of species in the mixture:
3
Enter vapor composition (y) of species 1:
0.21
Enter vapor composition (y) of species 2:
0.43
Enter vapor composition (y) of species 3:
0.36
Enter Temperature (in K):
373.15
Enter Pressure (in bar):
35
Enter critical temperature (Tc) of species 1:
190.6
Enter critical pressure (Pc) of species 1:
45.99
Enter acentric factor (w) of species 1:
0.012
Enter critical temperature (Tc) of species 2:
305.3
Enter critical pressure (Pc) of species 2:
48.72
Enter acentric factor (w) of species 2:
0.100
Enter critical temperature (Tc) of species 3:
369.8
Enter critical pressure (Pc) of species 3:
42.48
Enter acentric factor (w) of species 3:
0.152
```

This section of the program details the parameter constants to be used for each of the CEoS. Values of these constants were referenced from p. 100, Smith, Van Ness, et al. Introduction to ChE Thermo. (8th ed.)

Step 2: Code to store Parameter Constants for each of the CEoS.

```
% Parameter constants for PR EoS
  psi = 0.45724;
  omega = 0.07780;
  sigma = 1 + sqrt(2);
  epsilon = 1 - sqrt(2);
```

```
% Parameter constants for SRK EoS
  psi = 0.42748;
  omega = 0.08664;
  sigma = 1;
  epsilon = 0;
```

```
% Parameter constants for RK EoS
  psi = 0.42748;
  omega = 0.08664;
  sigma = 1;
  epsilon = 0;
```

```
% Parameter constants for vdW EoS
  psi = (27/64);
  omega = (1/8);
  sigma = 0;
  epsilon = 0;
```

Step 3. Code to calculate the Dimensionless Parameters for pure species.

This code is used for the Peng-Robinson Equation of State. This solves the reduced temperature, alpha parameter, and the dimensionless parameters for pure species (a_i and b_i). The algorithm is similar with the other CEoS programs except for the formula used to calculate the alpha parameter. Each CEoS have defined formulas to compute alpha.

```
% Determine the reduced temperature (tempr) for each species in the mixture
  tempr = zeros(1,n);
  for b = 1:n
      tempr(1,b) = temp/tempc(1,b);
  end

% Determine the parameter alpha in the PR EoS for each species in the system
  alpha_pr = zeros(1,n);
  for b = 1:n
      alpha_pr(1,b) = (1 + (0.37464 + 1.54226*w(1,b) - 0.26992*w(1,b)^2)*(1 - tempr(1,b)^(1/2)))^2;
  end

% Determine the values of the dimensionless parameters for pure species (a and b)
  % Solve for ai
    ai = zeros(1,n);
    for b = 1:n
        ai(1,b) = (psi*alpha_pr(1,b)*(83.14472^2)*(tempc(1,b)^2))/(pressc(1,b));
    end
  % Solve for bi
    bi = zeros(1,n);
    for b = 1:b
        bi(1,b) = (omega*83.14472*tempc(1,b))/(pressc(1,b));
    end
```

These parts create arrays containing calculated reduced temperatures and alpha parameters. The formula used are indicated below the figure.

To solve for a_i and b_i, several variables are needed (parameter constants for each CEoS, tempr, alpha, $T_{c,\,i}$, $P_{c,i}$, and R = 83.14 cm^3-bar/g mole-K). Arrays were created to store the calculated a_i and b_i values.

```
% Determine the parameter alpha in the SRK EoS for each species in the system
  alpha_srk = zeros(1,n);
  for b = 1:n
      alpha_srk(1,b) = (1 + (0.480 + 1.574*w(1,b) - 0.176*w(1,b)^2)*(1 - tempr(1,b)^(1/2)))^2;
  end
```

Equivalent code to calculate alpha in the SRK EoS.

```
% Determine the parameter alpha in the RK EoS for each species in the system
  alpha_rk = zeros(1,n);
  for b = 1:n
      alpha_rk(1,b) = (tempr(1,b)^(1/2));
  end
```

Equivalent code to calculate alpha in the RK EoS.

```
% Determine the parameter alpha in the vdW EoS for each species in the system
  alpha_vdW = ones(1,n);
```

Equivalent code to calculate alpha in the vdW EoS.

Eqn. of State	$\alpha(T_r)$
vdW (1873)	1
RK (1949)	$T_r^{-1/2}$
SRK (1972)	$\alpha_{\mathrm{SRK}}(T_r;\omega)^{\dagger} = \left[1+(0.480+1.574\,\omega-0.176\,\omega^2)(1-T_r^{1/2})\right]^2$
PR (1976)	$\alpha_{\mathrm{PR}}(T_r;\omega)^{\ddagger} = \left[1+\left(0.37464+1.54226\,\omega-0.26992\,\omega^2\right)\left(1-T_r^{1/2}\right)\right]^2$

$$T_R = \frac{T}{T_c} \qquad a_i = \psi\frac{\alpha(T_{ri},\omega)R^2T_{Ci}^2}{P_{Ci}} \qquad b_i = \Omega\frac{RT_{Ci}}{P_{Ci}}$$

Formulas used for this code.

Step 4: Code to calculate the Unlike Interaction Parameter (a_{ij})

This code is the same for all of the CEoS programs.

```
% Solve for the unlike interaction parameter (aij)
  % Initialize
    count = 0;
    for b = 1:n
        count = count + b;
    end
  % Solve for aij
    d = 1;
    aij = zeros(1, count);
    for b = 1:n-1
        for c = b+1:n
            aij(1,d) = (ai(1,b)*ai(1,c))^(1/2);
            d = d + 1;
        end
    end
    for e = 1:n
        aij(1,d) = (ai(1,e)*ai(1,e))^(1/2);
        d = d + 1;
    end
  % Transform aij from an array to a matrix
    aijmat = zeros(n);
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(b,c) = aij(1,d);
            d = d + 1;
        end
    end
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(c,b) = aij(1,d);
            d = d + 1;
        end
    end
    for e = 1:n
        aijmatrix(e,e) = aij(1,d);
        d = d + 1;
    end
```

This part adds element 0 to the array defined from 1 to n, where n is the number of species in the system. This modification is important in solving for the a_{ij} values.

Since values for each combination of b, c, and d are required, calculations for a_{ij} are done on a nested loop that run until specified limits. These limits were adjusted so that $a_{ij} = a_{ji}$, and to ensure that a_{ii} and a_{jj} will not be included in the resulting array.

To generalize the computed a_{ij} values, another calculation is performed in a for-loop that run until n. The formula for this calculation is given below the figure.

In this part, we transform array a_{ij} into a matrix, <aijmat>. The algorithm is the same as when setting up the array. Transforming the array into a matrix makes the succeeding calculations easier particularly in solving for the dimensionless parameters for mixtures.

$$a_{ij} = \left(a_i a_j\right)^{0.5}$$

This is the formula used to calculate for the unlike interaction parameter, a_{ij}.

Step 5: Code to calculate the Dimensionless Parameters for mixtures.

This code is the same for all of the CEoS programs.

```
% Determine the values of the dimensionless parameters for mixtures (amix, bmix, beta, and q)
  % Solve for amix
    amix = 0;
    for i = 1:n
        for j = 1:n
            amix = amix + (y(1,i)*y(1,j)*aijmatrix(i,j));
        end
    end
  % Solve for bmix
    bmix = 0;
    for i = 1:n
        bmix = bmix + (y(1,i)*bi(1,i));
    end
  % Solve for beta
    beta = 0;
    beta = (bmix*press)/(83.14472*temp);
  % Solve for q
    q = 0;
    q = (amix)/(bmix*83.14472*temp);
```

Since values for each combination of i and j are required, it would be more efficient to calculate for a_{mix} if it is performed inside a nested loop that runs until n, where n is the number of components. The formula used to calculate a_{mix} is listed below the figure.

In this part, we calculate the value of b_{mix}. The calculation is looped n number times. The formula used is listed below the figure. a_{mix} and b_{mix} are expected to produce scalar values.

Aside from calculating a_{mix} and b_{mix}, we also need to determine β and q. These parameters are used to calculate the mixture compressibility factors (Z) and the fugacity coefficients ($\hat{\emptyset}_i^v$). This part of the code describes the calculation procedures to obtain beta and q.

$$a = \sum_i \sum_j x_i x_j a_{ij} \qquad b = \sum_i x_i b_i$$

$$\beta = \frac{bP}{RT} \qquad q = \frac{a}{bRT}$$

These are the formulas used to calculate for the dimensionless parameters for mixtures, a_{mix}, b_{mix}, beta, and q.

$$Z = 1 + \beta - q\beta \frac{Z - \beta}{(Z + \varepsilon\beta)(Z + \sigma\beta)}$$

This is the formula used to calculate the value of the mixture compressibility factor (Z).

Step 6: Code to calculate for the Mixture Compressibility Factor (Z).

```
% Determine the mixture compressibility factor (z)
  % Use fzero function to find z
    z = fzero(@(z) (1 + beta - (q*beta)*((z - beta)/((z + epsilon*beta)*(z + sigma*beta))))-z, 0.5);

    fprintf('\n');
    fprintf('The value of the mixture compressibility factor (Z) is ');
    disp(z);
    fprintf('\n');
```

We use the MATLAB function <fzero> to calculate the value of Z. This function solves a root finding problem specified by a formula. In this case, the formula uses calculated values of q and β and as well as parameter constants for each CEoS.

The function displays the calculated Z value.

Step 7: Code to calculate the I factor and the Partial Molar EoS Properties.

This code is used for the Peng-Robinson Equation of State. This solves the I factor and the partial molar EoS properties ($\bar{a}_i$ and $\bar{q}_i$). The algorithm is similar with the other CEoS programs except for the formula used to calculate the I factor. The PR EoS, SRK EoS, and the RK EoS share the same formula, while the vdW EoS have a different one. The formulas used for this code is listed below.

```
% Determine I in the PR EoS
  I = 0;
  I = (1/(sigma - epsilon))*log((z + sigma*beta)/(z + epsilon*beta));

% Determine the partial molar EoS properties of the system (aibar, qibar)
  % Solve for aibar
  aibar = zeros(1,n);
  d = 1;
  for k = 1:n
      sum = 0;
      for i = 1:n
          for j = 1:n
              if i==j
                  sum = sum + 0;
              else
                  sum = sum + y(1,j)*aijmatrix(i,j);
              end
          end
      end
      sum = sum + (2*y(1,k)*aijmatrix(k,k)) - amix;
      aibar(1,k)= sum;
  end
  % Solve for qibar
  qibar = zeros(1,n);
  d = 1;
  for b = 1:n
      qibar(1,b) = q*(1 + (aibar(1,b)/amix) - (bi(1,b)/bmix));
  end
```

This part calculates the I factor. The I factor is an important character in solving for the fugacity coefficients ($\widehat{\emptyset}_i^v$) of each species in the system.

Since values for each combination of d, k, i and j are required, calculations for $\bar{a}_i$ are done on a nested loop that run until n, where n is the number of component species in the system.

$\bar{q}_i$ is determined after calculating $\bar{a}_i$, b_i, and b_{mix}. It is later used in finding the fugacity coefficient of species present in the system. The values of $\bar{a}_i$ and $\bar{q}_i$ are stored in arrays.

```
% Determine I in the SRK EoS
  I = 0;
  I = (1/(sigma - epsilon))*log((z + sigma*beta)/(z + epsilon*beta));
```

Equivalent code to calculate I in the SRK EoS.

```
% Determine I in the RK EoS
  I = 0;
  I = (1/(sigma - epsilon))*log((z + sigma*beta)/(z + epsilon*beta));
```

Equivalent code to calculate I in the RK EoS.

```
% Determine I in the vdW EoS
  I = 0;
  I = ((beta)/(z + epsilon*beta));
```

Equivalent code to calculate I in the vdW EoS.

These are the formulas used to calculate for the I factor and the partial molar EoS properties ($\bar{a}_i$ and $\bar{q}_i$).

$$I_{RK,SRK,PR} = \frac{1}{\sigma - \varepsilon} \ln\left(\frac{Z + \sigma\beta}{Z + \varepsilon\beta}\right) \qquad I_{vdW} = \frac{\beta}{Z + \varepsilon\beta}$$

$$\bar{a}_i = \left[\frac{\partial(na)}{\partial n_i}\right]_{T,n_j} = 2y_i a_{ii} + 2\sum_{i \neq j}^{N} y_j a_{ij} - a$$

$$\bar{q}_i = q\left(1 + \frac{\bar{a}_i}{a} - \frac{b_i}{b}\right)$$

Step 8: Code to calculate the Fugacity Coefficients and Component Fugacities.

This code solves for the fugacity coefficients ($\widehat{\emptyset}_i^v$) and the fugacity of each species in the mixture ($\hat{f}_i^v$). The algorithm is the same for all of the CEoS programs.

```
% Determine the fugacity coefficients and fugacity of each individual species in the mixture (phi, fhat)
  % Solve for lnphi
  lnphi = zeros(1,n);
  d = 1;
  for b = 1:n
      lnphi(1,b)=((bi(1,b)/bmix)*(z - 1) - log(z - beta) - (qibar(1,b)*I));
  end
  % Solve for phi
  phi = zeros(1,n);
  for a = 1:n
      phi(1,a) = exp(lnphi(1,a));
  end

  fprintf('The fugacity coefficients of each individual species in the mixture (phi values) are: ');
  fprintf('\n');
  for a = 1:n
      fprintf('phi%d:   %.4f\n', a, phi(1,a));
  end
  fprintf('\n');

  % Solve for fhat
  fhat = zeros(1,n);
  for a = 1:n
    fhat(1,a) = phi(1,a)*press*y(1,a);
  end

  fprintf('The fugacity of each individual species in the mixture (fhat values) are: ');
  fprintf('\n');
  for a = 1:n
      fprintf('fhat%d:   %.4f bar \n', a, fhat(1,a));
  end
```

These parts of the code solve for the values of $\ln\widehat{\emptyset}_i^v$ and $\widehat{\emptyset}_i^v$. To solve them, calculated parameters, such as b_i, b_{mix}, Z, β, $\bar{q}_i$, and I were used. Calculations were looped until n, where n is the number of species present in the system. Arrays were used to store these calculated values.

This function displays the calculated $\widehat{\emptyset}_i^v$ values.

In this part, the values of $\hat{f}_i^v$ were solved. Calculations were also looped until n and the obtained values are stored in an array. The formulas used in the calculation of $\widehat{\emptyset}_i^v$ and $\hat{f}_i^v$ are listed below.

This function displays the calculated $\hat{f}_i^v$ values.

$$ln\widehat{\phi_i} = \frac{b_i}{b}(Z-1) - \ln(Z-\beta) - \bar{q}_i I \qquad \hat{f}_i = \emptyset_i y_i P$$

These are the formulas used to calculate for the fugacity coefficients and the fugacity of each species in the mixture ($\widehat{\emptyset}_i^v$ and $\hat{f}_i^v$).

Sample Result for the Output

Output for the Peng-Robinson EoS

```
The value of the mixture compressibility factor (Z) is     0.8320


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8494
phi2:   0.7404
phi3:   0.6820

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   6.2431 bar
fhat2:   11.1434 bar
fhat3:   8.5930 bar
```

Output for the Soave-Redlich-Kwong EoS

```
The value of the mixture compressibility factor (Z) is     0.8454

The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8613
phi2:   0.7563
phi3:   0.7028

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   6.3308 bar
fhat2:   11.3820 bar
fhat3:   8.8557 bar
```

Output for the Redlich-Kwong EoS

```
The value of the mixture compressibility factor (Z) is     0.7902


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.7830
phi2:   0.6728
phi3:   0.6487

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   5.7552 bar
fhat2:   10.1258 bar
fhat3:   8.1731 bar
```

Output for the van der Waals EoS

```
The value of the mixture compressibility factor (Z) is     0.8472


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8413
phi2:   0.7372
phi3:   0.7034

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   6.1834 bar
fhat2:   11.0950 bar
fhat3:   8.8625 bar
```

Program code for the PR EoS

```
% LPA1 Specification 2: Cubic Equation of State
% Peng-Robinson Equation of State (PR EoS)
% Multi-component Gas/Vapor System
%
% Definition:
%   y = vapor composition
%   press = total pressure of the system
%   temp = total temperature of the system
%   pressc = critical pressure of each species in the system
%   tempc = critical temperature of each species in the system
%   w = acentric factor
%   z = mixture compressibility factor
%   phi = fugacity coefficients of each species in the system
%   fhat = fugacity of each species in the system

clc
clear

% Input the following variables
  n = input('Enter no. of species in the mixture: ');
  for b = 1:n
    y(1,b)=input(['Enter vapor composition (y) of species ' num2str(b) ': ']);
  end
  temp = input('Enter Temperature (in K): ');
  press = input('Enter Pressure (in bar): ');
  fprintf('\n');
  tempc = zeros(1,n);
  pressc = zeros(1,n);
  w = zeros(1,n);
  for b = 1:n
      tempc(1,b) = input(['Enter critical temperature (Tc) of species ' num2str(b) ':
']);
      pressc(1,b) = input(['Enter critical pressure (Pc) of species ' num2str(b) ':
']);
      w(1,b) = input(['Enter acentric factor (w) of species ' num2str(b) ': ']);
  end

% Parameter constants for PR EoS
  psi = 0.45724;
  omega = 0.07780;
  sigma = 1 + sqrt(2);
  epsilon = 1 - sqrt(2);

% Determine the reduced temperature (tempr) for each species in the mixture
  tempr = zeros(1,n);
  for b = 1:n
      tempr(1,b) = temp/tempc(1,b);
  end

% Determine the parameter alpha in the PR EoS for each species in the system
  alpha_pr = zeros(1,n);
  for b = 1:n
      alpha_pr(1,b) = (1 + (0.37464 + 1.54226*w(1,b) - 0.26992*w(1,b)^2)*(1 -
tempr(1,b)^(1/2)))^2;
  end

% Determine the values of the dimensionless parameters for pure species (a and b)
  % Solve for ai
    ai = zeros(1,n);
    for b = 1:n
        ai(1,b) = (psi*alpha_pr(1,b)*(83.14472^2)*(tempc(1,b)^2))/(pressc(1,b));
    end
  % Solve for bi
```

```
    bi = zeros(1,n);
    for b = 1:b
        bi(1,b) = (omega*83.14472*tempc(1,b))/(pressc(1,b));
    end

% Solve for the unlike interaction parameter (aij)
  % Initialize
    count = 0;
    for b = 1:n
        count = count + b;
    end
  % Solve for aij
    d = 1;
    aij = zeros(1, count);
    for b = 1:n-1
        for c = b+1:n
            aij(1,d) = (ai(1,b)*ai(1,c))^(1/2);
            d = d + 1;
        end
    end
    for e = 1:n
        aij(1,d) = (ai(1,e)*ai(1,e))^(1/2);
        d = d + 1;
    end
  % Transform aij from an array to a matrix
    aijmat = zeros(n);
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(b,c) = aij(1,d);
            d = d + 1;
        end
    end
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(c,b) = aij(1,d);
            d = d + 1;
        end
    end
    for e = 1:n
        aijmatrix(e,e) = aij(1,d);
        d = d + 1;
    end

% Determine the values of the dimensionless parameters for mixtures (amix, bmix, beta,
and q)
  % Solve for amix
    amix = 0;
    for i = 1:n
        for j = 1:n
            amix = amix + (y(1,i)*y(1,j)*aijmatrix(i,j));
        end
    end
  % Solve for bmix
    bmix = 0;
    for i = 1:n
        bmix = bmix + (y(1,i)*bi(1,i));
    end
  % Solve for beta
    beta = 0;
    beta = (bmix*press)/(83.14472*temp);
```

```
  % Solve for q
    q = 0;
    q = (amix)/(bmix*83.14472*temp);

% Determine the mixture compressibility factor (z)
  % Use fzero function to find z
    z = fzero(@(z) (1 + beta - (q*beta)*((z - beta)/((z + epsilon*beta)*(z +
sigma*beta))))-z, 0.5);

    fprintf('\n');
    fprintf('The value of the mixture compressibility factor (Z) is ');
    disp(z);
    fprintf('\n');

% Determine I in the PR EoS
  I = 0;
  I = (1/(sigma - epsilon))*log((z + sigma*beta)/(z + epsilon*beta));

% Determine the partial molar EoS properties of the system (aibar, qibar)
  % Solve for aibar
  aibar = zeros(1,n);
  d = 1;
  for k = 1:n
      sum = 0;
      for i = 1:n
          for j = 1:n
              if i==j
                  sum = sum + 0;
              else
                  sum = sum + y(1,j)*aijmatrix(i,j);
              end
          end
      end
      sum = sum + (2*y(1,k)*aijmatrix(k,k)) - amix;
      aibar(1,k)= sum;
  end
  % Solve for qibar
  qibar = zeros(1,n);
  d = 1;
  for b = 1:n
      qibar(1,b) = q*(1 + (aibar(1,b)/amix) - (bi(1,b)/bmix));
  end

% Determine the fugacity coefficients and fugacity of each individual species in the
mixture (phi, fhat)
  % Solve for lnphi
  lnphi = zeros(1,n);
  d = 1;
  for b = 1:n
      lnphi(1,b)=((bi(1,b)/bmix)*(z - 1) - log(z - beta) - (qibar(1,b)*I));
  end
  % Solve for phi
  phi = zeros(1,n);
  for a = 1:n
      phi(1,a) = exp(lnphi(1,a));
  end

  fprintf('The fugacity coefficients of each individual species in the mixture (phi
values) are: ');
  fprintf('\n');
  for a = 1:n
      fprintf('phi%d:   %.4f\n', a, phi(1,a));
```

```
  end
  fprintf('\n');

  % Solve for fhat
  fhat = zeros(1,n);
  for a = 1:n
    fhat(1,a) = phi(1,a)*press*y(1,a);
  end

  fprintf('The fugacity of each individual species in the mixture (fhat values) are:
');
  fprintf('\n');
  for a = 1:n
      fprintf('fhat%d:   %.4f bar \n', a, fhat(1,a));
  end
```

Program code for the SRK EoS

```
% LPA1 Specification 2: Cubic Equation of State
% Soave-Redlich-Kwong Equation of State (SRK EoS)
% Multi-component Gas/Vapor System
%
% Definition:
%   y = vapor composition
%   press = total pressure of the system
%   temp = total temperature of the system
%   pressc = critical pressure of each species in the system
%   tempc = critical temperature of each species in the system
%   w = acentric factor
%   z = mixture compressibility factor
%   phi = fugacity coefficients of each species in the system
%   fhat = fugacity of each species in the system

clc
clear

% Input the following variables
  n = input('Enter no. of species in the mixture: ');
  for b = 1:n
    y(1,b)=input(['Enter vapor composition (y) of species ' num2str(b) ': ']);
  end
  temp = input('Enter Temperature (in K): ');
  press = input('Enter Pressure (in bar): ');
  fprintf('\n');
  tempc = zeros(1,n);
  pressc = zeros(1,n);
  w = zeros(1,n);
  for b = 1:n
      tempc(1,b) = input(['Enter critical temperature (Tc) of species ' num2str(b) ':
']);
      pressc(1,b) = input(['Enter critical pressure (Pc) of species ' num2str(b) ':
']);
      w(1,b) = input(['Enter acentric factor (w) of species ' num2str(b) ': ']);
  end

% Parameter constants for SRK EoS
  psi = 0.42748;
  omega = 0.08664;
  sigma = 1;
  epsilon = 0;

% Determine the reduced temperature (tempr) for each species in the mixture
  tempr = zeros(1,n);
  for b = 1:n
      tempr(1,b) = temp/tempc(1,b);
  end

% Determine the parameter alpha in the SRK EoS for each species in the system
  alpha_srk = zeros(1,n);
  for b = 1:n
      alpha_srk(1,b) = (1 + (0.480 + 1.574*w(1,b) - 0.176*w(1,b)^2)*(1 -
tempr(1,b)^(1/2)))^2;
  end

% Determine the values of the dimensionless parameters for pure species (a and b)
  % Solve for ai
    ai = zeros(1,n);
    for b = 1:n
        ai(1,b) = (psi*alpha_srk(1,b)*(83.14472^2)*(tempc(1,b)^2))/(pressc(1,b));
    end
```

```
  % Solve for bi
    bi = zeros(1,n);
    for b = 1:b
        bi(1,b) = (omega*83.14472*tempc(1,b))/(pressc(1,b));
    end

% Solve for the unlike interaction parameter (aij)
  % Initialize
    count = 0;
    for b = 1:n
        count = count + b;
    end
  % Solve for aij
    d = 1;
    aij = zeros(1, count);
    for b = 1:n-1
        for c = b+1:n
            aij(1,d) = (ai(1,b)*ai(1,c))^(1/2);
            d = d + 1;
        end
    end
    for e = 1:n
        aij(1,d) = (ai(1,e)*ai(1,e))^(1/2);
        d = d + 1;
    end
  % Transform aij from an array to a matrix
    aijmat = zeros(n);
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(b,c) = aij(1,d);
            d = d + 1;
        end
    end
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(c,b) = aij(1,d);
            d = d + 1;
        end
    end
    for e = 1:n
        aijmatrix(e,e) = aij(1,d);
        d = d + 1;
    end

% Determine the values of the dimensionless parameters for mixtures (amix, bmix, beta,
and q)
  % Solve for amix
    amix = 0;
    for i = 1:n
        for j = 1:n
            amix = amix + (y(1,i)*y(1,j)*aijmatrix(i,j));
        end
    end
  % Solve for bmix
    bmix = 0;
    for i = 1:n
        bmix = bmix + (y(1,i)*bi(1,i));
    end
  % Solve for beta
    beta = 0;
```

```
    beta = (bmix*press)/(83.14472*temp);
  % Solve for q
    q = 0;
    q = (amix)/(bmix*83.14472*temp);

% Determine the mixture compressibility factor (z)
  % Use fzero function to find z
    z = fzero(@(z) (1 + beta - (q*beta)*((z - beta)/((z + epsilon*beta)*(z +
sigma*beta))))-z, 0.5);

    fprintf('\n');
    fprintf('The value of the mixture compressibility factor (Z) is ');
    disp(z);
    fprintf('\n');

% Determine I in the SRK EoS
  I = 0;
  I = (1/(sigma - epsilon))*log((z + sigma*beta)/(z + epsilon*beta));

% Determine the partial molar EoS properties of the system (aibar, qibar)
  % Solve for aibar
  aibar = zeros(1,n);
  d = 1;
  for k = 1:n
      sum = 0;
      for i = 1:n
          for j = 1:n
              if i==j
                  sum = sum + 0;
              else
                  sum = sum + y(1,j)*aijmatrix(i,j);
              end
          end
      end
      sum = sum + (2*y(1,k)*aijmatrix(k,k)) - amix;
      aibar(1,k)= sum;
  end
  % Solve for qibar
  qibar = zeros(1,n);
  d = 1;
  for b = 1:n
      qibar(1,b) = q*(1 + (aibar(1,b)/amix) - (bi(1,b)/bmix));
  end

% Determine the fugacity coefficients and fugacity of each individual species in the
mixture (phi, fhat)
  % Solve for lnphi
  lnphi = zeros(1,n);
  d = 1;
  for b = 1:n
      lnphi(1,b)=((bi(1,b)/bmix)*(z - 1) - log(z - beta) - (qibar(1,b)*I));
  end
  % Solve for phi
  phi = zeros(1,n);
  for a = 1:n
      phi(1,a) = exp(lnphi(1,a));
  end

  fprintf('The fugacity coefficients of each individual species in the mixture (phi
values) are: ');
  fprintf('\n');
  for a = 1:n
```

```
      fprintf('phi%d:   %.4f\n', a, phi(1,a));
  end
  fprintf('\n');

  % Solve for fhat
  fhat = zeros(1,n);
  for a = 1:n
    fhat(1,a) = phi(1,a)*press*y(1,a);
  end

  fprintf('The fugacity of each individual species in the mixture (fhat values) are:
');
  fprintf('\n');
  for a = 1:n
      fprintf('fhat%d:   %.4f bar \n', a, fhat(1,a));
  end
```

Program code for the RK EoS

```
% LPA1 Specification 2: Cubic Equation of State
% Redlich-Kwong Equation of State (RK EoS)
% Multi-component Gas/Vapor System
%
% Definition:
%   y = vapor composition
%   press = total pressure of the system
%   temp = total temperature of the system
%   pressc = critical pressure of each species in the system
%   tempc = critical temperature of each species in the system
%   w = acentric factor
%   z = mixture compressibility factor
%   phi = fugacity coefficients of each species in the system
%   fhat = fugacity of each species in the system

clc
clear

% Input the following variables
  n = input('Enter no. of species in the mixture: ');
  for b = 1:n
    y(1,b)=input(['Enter vapor composition (y) of species ' num2str(b) ': ']);
  end
  temp = input('Enter Temperature (in K): ');
  press = input('Enter Pressure (in bar): ');
  fprintf('\n');
  tempc = zeros(1,n);
  pressc = zeros(1,n);
  w = zeros(1,n);
  for b = 1:n
      tempc(1,b) = input(['Enter critical temperature (Tc) of species ' num2str(b) ':
']);
      pressc(1,b) = input(['Enter critical pressure (Pc) of species ' num2str(b) ':
']);
      w(1,b) = input(['Enter acentric factor (w) of species ' num2str(b) ': ']);
  end

% Parameter constants for RK EoS
  psi = 0.42748;
  omega = 0.08664;
  sigma = 1;
  epsilon = 0;

% Determine the reduced temperature (tempr) for each species in the mixture
  tempr = zeros(1,n);
  for b = 1:n
      tempr(1,b) = temp/tempc(1,b);
  end

% Determine the parameter alpha in the RK EoS for each species in the system
  alpha_rk = zeros(1,n);
  for b = 1:n
      alpha_rk(1,b) = (tempr(1,b)^(-1/2));
  end

% Determine the values of the dimensionless parameters for pure species (a and b)
  % Solve for ai
    ai = zeros(1,n);
    for b = 1:n
        ai(1,b) = (psi*alpha_rk(1,b)*(83.14472^2)*(tempc(1,b)^2))/(pressc(1,b));
    end
  % Solve for bi
```

```
    bi = zeros(1,n);
    for b = 1:b
        bi(1,b) = (omega*83.14472*tempc(1,b))/(pressc(1,b));
    end

% Solve for the unlike interaction parameter (aij)
  % Initialize
    count = 0;
    for b = 1:n
        count = count + b;
    end
  % Solve for aij
    d = 1;
    aij = zeros(1, count);
    for b = 1:n-1
        for c = b+1:n
            aij(1,d) = (ai(1,b)*ai(1,c))^(1/2);
            d = d + 1;
        end
    end
    for e = 1:n
        aij(1,d) = (ai(1,e)*ai(1,e))^(1/2);
        d = d + 1;
    end
  % Transform aij from an array to a matrix
    aijmat = zeros(n);
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(b,c) = aij(1,d);
            d = d + 1;
        end
    end
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(c,b) = aij(1,d);
            d = d + 1;
        end
    end
    for e = 1:n
        aijmatrix(e,e) = aij(1,d);
        d = d + 1;
    end

% Determine the values of the dimensionless parameters for mixtures (amix, bmix, beta,
and q)
  % Solve for amix
    amix = 0;
    for i = 1:n
        for j = 1:n
            amix = amix + (y(1,i)*y(1,j)*aijmatrix(i,j));
        end
    end
  % Solve for bmix
    bmix = 0;
    for i = 1:n
        bmix = bmix + (y(1,i)*bi(1,i));
    end
  % Solve for beta
    beta = 0;
    beta = (bmix*press)/(83.14472*temp);
```

```
  % Solve for q
    q = 0;
    q = (amix)/(bmix*83.14472*temp);

% Determine the mixture compressibility factor (z)
  % Use fzero function to find z
    z = fzero(@(z) (1 + beta - (q*beta)*((z - beta)/((z + epsilon*beta)*(z +
sigma*beta))))-z, 0.5);

    fprintf('\n');
    fprintf('The value of the mixture compressibility factor (Z) is ');
    disp(z);
    fprintf('\n');

% Determine I in the RK EoS
  I = 0;
  I = (1/(sigma - epsilon))*log((z + sigma*beta)/(z + epsilon*beta));

% Determine the partial molar EoS properties of the system (aibar, qibar)
  % Solve for aibar
  aibar = zeros(1,n);
  d = 1;
  for k = 1:n
      sum = 0;
      for i = 1:n
          for j = 1:n
              if i==j
                  sum = sum + 0;
              else
                  sum = sum + y(1,j)*aijmatrix(i,j);
              end
          end
      end
      sum = sum + (2*y(1,k)*aijmatrix(k,k)) - amix;
      aibar(1,k)= sum;
  end
  % Solve for qibar
  qibar = zeros(1,n);
  d = 1;
  for b = 1:n
      qibar(1,b) = q*(1 + (aibar(1,b)/amix) - (bi(1,b)/bmix));
  end

% Determine the fugacity coefficients and fugacity of each individual species in the
mixture (phi, fhat)
  % Solve for lnphi
  lnphi = zeros(1,n);
  d = 1;
  for b = 1:n
      lnphi(1,b)=((bi(1,b)/bmix)*(z - 1) - log(z - beta) - (qibar(1,b)*I));
  end
  % Solve for phi
  phi = zeros(1,n);
  for a = 1:n
      phi(1,a) = exp(lnphi(1,a));
  end

  fprintf('The fugacity coefficients of each individual species in the mixture (phi
values) are: ');
  fprintf('\n');
  for a = 1:n
      fprintf('phi%d:   %.4f\n', a, phi(1,a));
```

```
  end
  fprintf('\n');

  % Solve for fhat
  fhat = zeros(1,n);
  for a = 1:n
    fhat(1,a) = phi(1,a)*press*y(1,a);
  end

  fprintf('The fugacity of each individual species in the mixture (fhat values) are:
');
  fprintf('\n');
  for a = 1:n
      fprintf('fhat%d:   %.4f bar \n', a, fhat(1,a));
  end
```

Program code for the vdW EoS

```
% LPA1 Specification 2: Cubic Equation of State
% Van der Waals Equation of State (vdW EoS)
% Multi-component Gas/Vapor System
%
% Definition:
%   y = vapor composition
%   press = total pressure of the system
%   temp = total temperature of the system
%   pressc = critical pressure of each species in the system
%   tempc = critical temperature of each species in the system
%   w = acentric factor
%   z = mixture compressibility factor
%   phi = fugacity coefficients of each species in the system
%   fhat = fugacity of each species in the system

clc
clear

% Input the following variables
  n = input('Enter no. of species in the mixture: ');
  for b = 1:n
    y(1,b)=input(['Enter vapor composition (y) of species ' num2str(b) ': ']);
  end
  temp = input('Enter Temperature (in K): ');
  press = input('Enter Pressure (in bar): ');
  fprintf('\n');
  tempc = zeros(1,n);
  pressc = zeros(1,n);
  w = zeros(1,n);
  for b = 1:n
      tempc(1,b) = input(['Enter critical temperature (Tc) of species ' num2str(b) ':
']);
      pressc(1,b) = input(['Enter critical pressure (Pc) of species ' num2str(b) ':
']);
      w(1,b) = input(['Enter acentric factor (w) of species ' num2str(b) ': ']);
  end

% Parameter constants for vdW EoS
  psi = (27/64);
  omega = (1/8);
  sigma = 0;
  epsilon = 0;

% Determine the reduced temperature (tempr) for each species in the mixture
  tempr = zeros(1,n);
  for b = 1:n
      tempr(1,b) = temp/tempc(1,b);
  end

% Determine the parameter alpha in the vdW EoS for each species in the system
  alpha_vdW = ones(1,n);

% Determine the values of the dimensionless parameters for pure species (a and b)
  % Solve for ai
    ai = zeros(1,n);
    for b = 1:n
        ai(1,b) = (psi*alpha_vdW(1,b)*(83.14472^2)*(tempc(1,b)^2))/(pressc(1,b));
    end
  % Solve for bi
    bi = zeros(1,n);
    for b = 1:b
        bi(1,b) = (omega*83.14472*tempc(1,b))/(pressc(1,b));
```

```
    end

% Solve for the unlike interaction parameter (aij)
  % Initialize
    count = 0;
    for b = 1:n
        count = count + b;
    end
  % Solve for aij
    d = 1;
    aij = zeros(1, count);
    for b = 1:n-1
        for c = b+1:n
            aij(1,d) = (ai(1,b)*ai(1,c))^(1/2);
            d = d + 1;
        end
    end
    for e = 1:n
        aij(1,d) = (ai(1,e)*ai(1,e))^(1/2);
        d = d + 1;
    end
  % Transform aij from an array to a matrix
    aijmat = zeros(n);
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(b,c) = aij(1,d);
            d = d + 1;
        end
    end
    d = 1;
    for b = 1:n-1
        for c = b+1:n
            aijmatrix(c,b) = aij(1,d);
            d = d + 1;
        end
    end
    for e = 1:n
        aijmatrix(e,e) = aij(1,d);
        d = d + 1;
    end

% Determine the values of the dimensionless parameters for mixtures (amix, bmix, beta,
and q)
  % Solve for amix
    amix = 0;
    for i = 1:n
        for j = 1:n
            amix = amix + (y(1,i)*y(1,j)*aijmatrix(i,j));
        end
    end
  % Solve for bmix
    bmix = 0;
    for i = 1:n
        bmix = bmix + (y(1,i)*bi(1,i));
    end
  % Solve for beta
    beta = 0;
    beta = (bmix*press)/(83.14472*temp);
  % Solve for q
    q = 0;
    q = (amix)/(bmix*83.14472*temp);
```

```
% Determine the mixture compressibility factor (z)
  % Use fzero function to find z
    z = fzero(@(z) (1 + beta - (q*beta)*((z - beta)/((z + epsilon*beta)*(z +
sigma*beta))))-z, 0.5);

    fprintf('\n');
    fprintf('The value of the mixture compressibility factor (Z) is ');
    disp(z);
    fprintf('\n');

% Determine I in the vdW EoS
  I = 0;
  I = ((beta)/(z + epsilon*beta));

% Determine the partial molar EoS properties of the system (aibar, qibar)
  % Solve for aibar
  aibar = zeros(1,n);
  d = 1;
  for k = 1:n
      sum = 0;
      for i = 1:n
          for j = 1:n
              if i==j
                  sum = sum + 0;
              else
                  sum = sum + y(1,j)*aijmatrix(i,j);
              end
          end
      end
      sum = sum + (2*y(1,k)*aijmatrix(k,k)) - amix;
      aibar(1,k)= sum;
  end
  % Solve for qibar
  qibar = zeros(1,n);
  d = 1;
  for b = 1:n
      qibar(1,b) = q*(1 + (aibar(1,b)/amix) - (bi(1,b)/bmix));
  end

% Determine the fugacity coefficients and fugacity of each individual species in the
mixture (phi, fhat)
  % Solve for lnphi
  lnphi = zeros(1,n);
  d = 1;
  for b = 1:n
      lnphi(1,b)=((bi(1,b)/bmix)*(z - 1) - log(z - beta) - (qibar(1,b)*I));
  end
  % Solve for phi
  phi = zeros(1,n);
  for a = 1:n
      phi(1,a) = exp(lnphi(1,a));
  end

  fprintf('The fugacity coefficients of each individual species in the mixture (phi
values) are: ');
  fprintf('\n');
  for a = 1:n
      fprintf('phi%d:   %.4f\n', a, phi(1,a));
  end
  fprintf('\n');
```

```
  % Solve for fhat
  fhat = zeros(1,n);
  for a = 1:n
    fhat(1,a) = phi(1,a)*press*y(1,a);
  end

  fprintf('The fugacity of each individual species in the mixture (fhat values) are:
');
  fprintf('\n');
  for a = 1:n
      fprintf('fhat%d:   %.4f bar \n', a, fhat(1,a));
  end
```

TEST:
System A: methane(1), ethane(2), and propane (3) at 373.15K and 35 bar
(y_1 = 0.21, y_2 = 0.43, y_3 = 0.36)

Output for Program 1: Peng-Robinson Equation of State

```
Enter no. of species in the mixture:
3
Enter vapor composition (y) of species 1:
0.21
Enter vapor composition (y) of species 2:
0.43
Enter vapor composition (y) of species 3:
0.36
Enter Temperature (in K):
373.15
Enter Pressure (in bar):
35

Enter critical temperature (Tc) of species 1:
190.6
Enter critical pressure (Pc) of species 1:
45.99
Enter acentric factor (w) of species 1:
0.012
Enter critical temperature (Tc) of species 2:
305.3
Enter critical pressure (Pc) of species 2:
48.72
Enter acentric factor (w) of species 2:
0.100
Enter critical temperature (Tc) of species 3:
369.8
Enter critical pressure (Pc) of species 3:
42.48
Enter acentric factor (w) of species 3:
0.152

The value of the mixture compressibility factor (Z) is     0.8320


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8494
phi2:   0.7404
phi3:   0.6820

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   6.2431 bar
fhat2:   11.1434 bar
fhat3:   8.5930 bar
>>
```

Output for Program 2: Soave-Redlich-Kwong Equation of State

```
Enter no. of species in the mixture:
3
Enter vapor composition (y) of species 1:
0.21
Enter vapor composition (y) of species 2:
0.43
Enter vapor composition (y) of species 3:
0.36
Enter Temperature (in K):
373.15
Enter Pressure (in bar):
35

Enter critical temperature (Tc) of species 1:
190.6
Enter critical pressure (Pc) of species 1:
45.99
Enter acentric factor (w) of species 1:
0.012
Enter critical temperature (Tc) of species 2:
305.3
Enter critical pressure (Pc) of species 2:
48.72
Enter acentric factor (w) of species 2:
0.100
Enter critical temperature (Tc) of species 3:
369.8
Enter critical pressure (Pc) of species 3:
42.48
Enter acentric factor (w) of species 3:
0.152

The value of the mixture compressibility factor (Z) is     0.8529


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8714
phi2:   0.7689
phi3:   0.7104

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   6.4047 bar
fhat2:   11.5716 bar
fhat3:   8.9512 bar
>>
```

Output for Program 3: Redlich-Kwong Equation of State

```
Enter no. of species in the mixture: 3
Enter vapor composition (y) of species 1: .21
Enter vapor composition (y) of species 2: .43
Enter vapor composition (y) of species 3: .36
Enter Temperature (in K): 373.15
Enter Pressure (in bar): 35

Enter critical temperature (Tc) of species 1: 190.6
Enter critical pressure (Pc) of species 1: 45.99
Enter acentric factor (w) of species 1: .012
Enter critical temperature (Tc) of species 2: 305.3
Enter critical pressure (Pc) of species 2: 48.72
Enter acentric factor (w) of species 2: .1
Enter critical temperature (Tc) of species 3: 369.8
Enter critical pressure (Pc) of species 3: 42.48
Enter acentric factor (w) of species 3: .152

The value of the mixture compressibility factor (Z) is     0.8454

The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8613
phi2:   0.7563
phi3:   0.7028

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   6.3308 bar
fhat2:   11.3820 bar
fhat3:   8.8557 bar
```

Output for Program 4: van der Waals Equation of State

```
Enter no. of species in the mixture:
3
Enter vapor composition (y) of species 1:
0.21
Enter vapor composition (y) of species 2:
0.43
Enter vapor composition (y) of species 3:
0.36
Enter Temperature (in K):
373.15
Enter Pressure (in bar):
35

Enter critical temperature (Tc) of species 1:
190.6
Enter critical pressure (Pc) of species 1:
45.99
Enter acentric factor (w) of species 1:
0.012
Enter critical temperature (Tc) of species 2:
305.3
Enter critical pressure (Pc) of species 2:
48.72
Enter acentric factor (w) of species 2:
0.100
Enter critical temperature (Tc) of species 3:
369.8
Enter critical pressure (Pc) of species 3:
42.48
Enter acentric factor (w) of species 3:
0.152

The value of the mixture compressibility factor (Z) is     0.8472


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8413
phi2:   0.7372
phi3:   0.7034

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   6.1834 bar
fhat2:   11.0950 bar
fhat3:   8.8625 bar
>>
```

TEST:

System B: n-pentane(1), n-hexane(2), cyclohexane(3), and cyclopentane(4) at 500K and 7 bar ($y_1 = 0.35$, $y_2 = 0.35$, $y_3 = 0.15$, $y_4 = 0.15$)

Output for Program 1: Peng-Robinson Equation of State

```
Enter no. of species in the mixture:
4
Enter vapor composition (y) of species 1:
0.35
Enter vapor composition (y) of species 2:
0.35
Enter vapor composition (y) of species 3:
0.15
Enter vapor composition (y) of species 4:
0.15
Enter Temperature (in K):
500
Enter Pressure (in bar):
7

Enter critical temperature (Tc) of species 1:
469.7
Enter critical pressure (Pc) of species 1:
33.70
Enter acentric factor (w) of species 1:
0.252
Enter critical temperature (Tc) of species 2:
507.6
Enter critical pressure (Pc) of species 2:
30.25
Enter acentric factor (w) of species 2:
0.301
Enter critical temperature (Tc) of species 3:
553.6
Enter critical pressure (Pc) of species 3:
40.73
Enter acentric factor (w) of species 3:
0.210
Enter critical temperature (Tc) of species 4:
511.8
Enter critical pressure (Pc) of species 4:
45.02
Enter acentric factor (w) of species 4:
0.196

The value of the mixture compressibility factor (Z) is     0.9199


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.7905
phi2:   0.7756
phi3:   0.8116
phi4:   0.8163

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   1.9368 bar
fhat2:   1.9002 bar
fhat3:   0.8522 bar
fhat4:   0.8572 bar
>>
```

Output for Program 2: Soave-Redlich-Kwong Equation of State

```
Enter no. of species in the mixture:
4
Enter vapor composition (y) of species 1:
0.35
Enter vapor composition (y) of species 2:
0.35
Enter vapor composition (y) of species 3:
0.15
Enter vapor composition (y) of species 4:
0.15
Enter Temperature (in K):
500
Enter Pressure (in bar):
7

Enter critical temperature (Tc) of species 1:
469.7
Enter critical pressure (Pc) of species 1:
33.70
Enter acentric factor (w) of species 1:
0.252
Enter critical temperature (Tc) of species 2:
507.6
Enter critical pressure (Pc) of species 2:
30.25
Enter acentric factor (w) of species 2:
0.301
Enter critical temperature (Tc) of species 3:
553.6
Enter critical pressure (Pc) of species 3:
40.73
Enter acentric factor (w) of species 3:
0.210
Enter critical temperature (Tc) of species 4:
511.8
Enter critical pressure (Pc) of species 4:
45.02
Enter acentric factor (w) of species 4:
0.196

The value of the mixture compressibility factor (Z) is     0.9275


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8049
phi2:   0.7906
phi3:   0.8242
phi4:   0.8286

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   1.9719 bar
fhat2:   1.9369 bar
fhat3:   0.8654 bar
fhat4:   0.8701 bar
>>
```

Output for Program 3: Redlich-Kwong Equation of State

```
Enter no. of species in the mixture: 4
Enter vapor composition (y) of species 1: .35
Enter vapor composition (y) of species 2: .35
Enter vapor composition (y) of species 3: .15
Enter vapor composition (y) of species 4: .15
Enter Temperature (in K): 500
Enter Pressure (in bar): 7

Enter critical temperature (Tc) of species 1: 469.7
Enter critical pressure (Pc) of species 1: 33.7
Enter acentric factor (w) of species 1: .252
Enter critical temperature (Tc) of species 2: 507.6
Enter critical pressure (Pc) of species 2: 30.25
Enter acentric factor (w) of species 2: .301
Enter critical temperature (Tc) of species 3: 553.6
Enter critical pressure (Pc) of species 3: 40.73
Enter acentric factor (w) of species 3: .21
Enter critical temperature (Tc) of species 4: 511.8
Enter critical pressure (Pc) of species 4: 45.02
Enter acentric factor (w) of species 4: .196

The value of the mixture compressibility factor (Z) is     0.9275

The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8044
phi2:   0.7916
phi3:   0.8255
phi4:   0.8294

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   1.9708 bar
fhat2:   1.9394 bar
fhat3:   0.8668 bar
fhat4:   0.8709 bar
```

Output for Program 4: van der Waals Equation of State

```
Enter no. of species in the mixture:
4
Enter vapor composition (y) of species 1:
0.35
Enter vapor composition (y) of species 2:
0.35
Enter vapor composition (y) of species 3:
0.15
Enter vapor composition (y) of species 4:
0.15
Enter Temperature (in K):
500
Enter Pressure (in bar):
7

Enter critical temperature (Tc) of species 1:
469.7
Enter critical pressure (Pc) of species 1:
33.70
Enter acentric factor (w) of species 1:
0.252
Enter critical temperature (Tc) of species 2:
507.6
Enter critical pressure (Pc) of species 2:
30.25
Enter acentric factor (w) of species 2:
0.301
Enter critical temperature (Tc) of species 3:
553.6
Enter critical pressure (Pc) of species 3:
40.73
Enter acentric factor (w) of species 3:
0.210
Enter critical temperature (Tc) of species 4:
511.8
Enter critical pressure (Pc) of species 4:
45.02
Enter acentric factor (w) of species 4:
0.196

The value of the mixture compressibility factor (Z) is     0.9366


The fugacity coefficients of each individual species in the mixture (phi values) are:
phi1:   0.8130
phi2:   0.8033
phi3:   0.8366
phi4:   0.8384

The fugacity of each individual species in the mixture (fhat values) are:
fhat1:   1.9918 bar
fhat2:   1.9681 bar
fhat3:   0.8784 bar
fhat4:   0.8804 bar
>>
```