

Chapter Two: Computer Modeling Applications for Water and Wastewater Properties

2.7 Homework Problems in Water and Wastewater Properties

2.7.1 Discussion Problems

1)

Solution:

See Section 2.2 of Chapter 2 of the book.

2)

Solution:

See Sections 2.3, 2.4, 2.5 and 2.6 of Chapter 2 of the book.

3)

Solution:

See Section 2.3 of Chapter 2 of the book.

4)

Solution:

See Section 2.3 of Chapter 2 of the book.

5)

Solution:

See Section 2.4 of Chapter 2 of the book.

6)

Solution:

See latest *Standard Methods for the Examination of Water and Wastewater*, American Water Works Association, APHA.

7)

Solution:

See latest *Standard Methods for the Examination of Water and Wastewater*, American Water Works Association, APHA.

8)

Solution:

Using equations 2.6, 2.7, 2.8, 2.9, and 2.10.

$$s_a = 0.03 + 1.805c_h \quad (2.6)$$

Where:

s_a = Salinity, g/kg.

c_h = Chlorinity, g/kg.

$$\rho = \frac{m}{V} \quad (2.7)$$

Where:

ρ = Density of the fluid, kg/m³.

m = Mass, kg.

V = Volume, m^3 .

$$v = \frac{1}{\rho} \quad (2.8)$$

Where:

v = Specific volume of the fluid, m^3/kg .

ρ = Density of the fluid, kg/m^3 .

$$\gamma = \frac{mg}{V} = \rho g \quad (2.9)$$

Where:

γ = Specific weight, N/m^3 .

m = Mass, kg .

g = Gravitational acceleration, m/s^2 .

V = Volume, m^3 .

ρ = Density of the fluid, kg/m^3 .

$$s.g. = \frac{\rho}{\rho_{\text{water at } 4^\circ C}} \quad (2.10)$$

9)

Solution:

Search relevant journals of fluid mechanics, for example <https://www.cambridge.org>.

10)

Table 2.1 Examples of Different Types of Fluids

Type	Example
Newtonian	Most fluids, Water, Gasoline, Oil, Air, Emulsions
Dilatants	Moving sand
Non-Newtonian	Flubber, Oobleck (suspension of starch in water)
Ideal liquid	Practically no ideal fluid exists. Real fluids are like kerosene, petrol, castor oil
Ideal solid	
Pseudoplastic	Clay, Cement, Milk, Colloidal solutions
Plastic	Sludge
Thixotropic	Raw sludge, Ketchup, Yoghurt, Paint
Rheoplectic	Gypsum paste, Printer ink

11)

Solution:

Absolute viscosity coefficient is equal to kinematic viscosity coefficient, multiplied by density,

i.e.:

$$\mu = \rho \nu \quad (2.12)$$

where:

μ = Dynamic (absolute) viscosity, $\text{N}\cdot\text{s}\cdot\text{m}^{-2}$.

ν = Kinematic viscosity, $\text{m}^2\cdot\text{s}^{-1}$ (usually defined as the ratio of dynamic viscosity to mass density).

ρ = Density, $\text{kg}\cdot\text{m}^{-3}$.

12)

Solution:

Surface tension is a property of a liquid that permits the attraction between molecules to form an imaginary film that is able to resist tensile forces at the interface between two immiscible liquids or at the interface between a liquid and a gas. As represented by equation 2.13.

$$h = \frac{4\sigma \cos\phi}{gD(\rho_l - \rho_g)} \quad (2.13)$$

Where:

h = Column height by which liquid rose along capillary tube, m.

σ = Surface tension, N/m.

ϕ = Angle of contact subtended by the heavier fluid and tube ($= 0^\circ$ for most organic liquids and water against glass {provided the glass is wet with a film of the liquid}, $= 130^\circ$ for mercury against glass).

ρ_l = Density of liquid, kg/m^3 .

ρ_g = Density of gas (or light liquid), kg/m^3 .

g = Local gravitational acceleration, m/s^2 .

D = Inside diameter of capillary tube, m

13)

Solution:

- If any formulation changes at the molecular level (e.g. by the addition of another

chemical or surfactant), or if anything contaminates the fluid, then its surface tension will change. Measuring surface tension level is a direct indicator of the quality of any chemical and any formulation.

- It is surface tension which keeps the billions of cells in the human body functional. Surface tension ensures the proper organization of the cellular bio-molecules, proteins, lipids, and nucleic acids, into membranes and various types of cellular organelles.
- Surface tension is a fundamental property of water, making it an ideal medium allowing for life to exist. As a general, easily comprehensible example, it is surface tension that makes water drops spherical.

14)

Solution:

Bulk modulus is a property that is used to evaluate the degree of compressibility. Equation 2.14 gives the bulk modulus.

$$E_b = -\frac{dP}{dV/V} = -\frac{dP}{\rho/d\rho} \quad (2.14)$$

where:

E_b = Bulk modulus, $N \cdot m^{-2}$.

dP = Differential change in pressure, Pa.

dV = Differential change in volume, m^3 .

V = Volume, m^3 .

ρ = Density of fluid, kg/m^3 .

15)

Solution:**Table 2.2 Differences between Alpha, Beta and Gamma Radiation**

	Alpha Particles	Beta Particles	Gamma Rays
Energy level	Most energetic	Much smaller than Alpha	Waves of electromagnetic energy
Travel distance	Few inches in air	Travel with high speed	Travel with speed of light
Stopping material	Sheet of paper	Thin sheet of aluminium or glass	Thick wall of concrete, lead , or steel.

Also See Section 6 of Chapter 2 of the book.

16)

Solution:

See Section 4 of Chapter 2 of the book.

17)

Solution:

See Section 4 of Chapter 2 of the book.

18)

Solution:

Bases and alkali substances are common, both in the natural environment and in our daily use.

Some examples of everyday use alkali substances are toothpastes and baking soda. Ammonia is an exception to the common pattern for alkali substances, as it is a water-soluble base without a hydroxyl group. Copper oxide is a relatively common base that is not water soluble, and thus it is not an alkali.

Any basic substance has a pH above 7.0 (the pH of pure water). Acidic substances have a pH of below 7.0. Basic substances are being basic because they are capable of accepting hydrogen ions, and hydrogen ions are what give acids their caustic properties. In the case of many alkali substances, the hydrogen ions bond with hydroxyl ions to create water. The non-hydroxyl portions of the bases often form salts with the remnants of the acids. In the case of ammonia, the hydrogen ion is accepted to form ammonium.

19)

Solution:

Hardness is generally associated with divalent metallic cations of calcium, Ca^{++} , or magnesium, Mg^{++} , or strontium, Sr^{++} , or ferrous ion, Fe^{++} , or manganous ions, Mn^{++} .

Advantages and disadvantages of hardness

- Hard water aids growth of teeth and bones.
- Hard water reduces degree of toxicity by poisoning with lead oxide (PbO) from pipelines made of lead.
- Soft waters are thought to be associated with cardiovascular diseases.
- Financial losses to hard water users via consumption of more soap.
- Development of residues, when using hard water, on hot water systems, boilers, domestic appliances, fittings, utensils, bath tubs, sinks, dishwashers and wash-hand basins.

- Staining of clothes, and some household utensils, upon use of hard water.
- Hard water precipitates hardness-soap which may remain in pores of skins making it feeling rough and uncomfortable.
- Increase of a laxative effect on new hard water consumers primarily with existence of magnesium sulfates

20)

Solution:

See latest *Standard Methods for the Examination of Water and Wastewater*, American Water Works Association, APHA.

21)

Solution:

See Section 4 of Chapter 2 of the book.

22)

Solution:

The Biochemical Oxygen Demand (BOD) test measures the relative amount of oxygen that is needed to biologically stabilize organic matter present in a sample.

Equation 2.24 gives the value of BOD exerted at any time t and temperature T (See Figure 2.6).

$$BOD_t^T = L_o - L_t = L_o(1 - 10^{-k^{\circ}t}) \quad (2.24)$$

23)

Solution:

The advantages of the test include: estimation of the size of treatment units, evaluation of treatment efficiency, and estimation of the relative amount of oxygen required for oxidation of organic pollutants

24)

Table 2.3 Chemicals not in source (raw) water

Chemical	First chemical (Ans)	Second chemical (Ans)	Third chemical (Ans)
Example	Chlorine	Aluminum sulfate	Soda ash (Sodium carbonate) and Sodium hydroxide
How chemical entered water	Disinfection process	Coagulation/sedimentation treatment	Adjusting pH
Potential health impacts and effects	Carcinogenic trihalomethanes if formed	Aluminum linked with Alzheimer's disease, discoloration	Eye irritation or abdominal pain in high concentration
Suggested methods for getting rid of harmful chemical	Proper dosing, getting rid of residual chlorine	Use of optimum dose	Proper dosing

25)

Solution:

See extra references, for example World Water Works Association

(<http://www.worldwaterworks.com>), or Global Water Partnership (GWP – <http://www.gwp.org>).

26)

Solution:

See extra references, for example World Health Organization (<http://www.who.int>), or The World Bank (<http://www.worldbank.org>).

27)

Table 2.4 Group Matching

Group (I)	Rearranged group (II) (Ans)
Pollution	Improper storage
True solutions removal	Gas transfer
Pathogen removal	Chlorination
Colloidal suspensions removal	Chemical coagulation
Physical treatment	Sedimentation
Preliminary treatment	Grit removal
Leakage	Water loss
Sanitation barrier	Hand washing
House fly	<i>Muscadomestica</i>
Scistosomiasis	<i>Scistosoma</i>
Malaria	Anopheles mosquitoes
Eye disease	Trachoma
Water contact diseases	Water-based
Water washed diseases	Quantity related
Sanitation - related diseases	Fecal related
Water borne diseases	Quality related
Water insect vector	Water site related
Trypanosomiasis	African sleeping sickness
Methaemoglobinaemia	Blue babies syndrome
Fluoride	Tooth decay

28)

- a) T
- b) T
- c) T
- d) T
- e) T
- f) T

- g) T
h) T

29)

- a) **disease causing**
b) **Health**
c) **poverty**
d) **Lead**
e) **lethal dose**
f) **Guidelines**
g) **corrosive**

2.7.2 Specific Mathematical Problems

1)

Solution

- Given: Angle of contact, Column height, Temperature, Inside diameter of capillary tube.
- Use equation of surface tension to calculate its value given the above.

$$\sigma = \frac{h * g D (\rho_l - \rho_g)}{4 \cos \phi}$$

Where:

σ = Surface tension, N/m.

h = Column height by which liquid rose along capillary tube, m.

ϕ = Angle of contact subtended by the heavier fluid and tube (= 0° for most organic liquids and water against glass {provided the glass is wet with a film of the liquid = 130° for mercury against glass}).

ρ_l = Density of liquid, kg/m³.

ρ_g = Density of gas (or light liquid), kg/m³.

$g =$ Local gravitational acceleration, m/s^2 .

$D =$ Inside diameter of capillary tube, m

Program 2.1: Surface Tension

```
Imports System.Math

Public Class Form1

    Const g = 9.81           'acceleration due to gravity

    '*****

    ' Density column of Table Appendix A1:

    ' Physical Properties of Water

    '*****

    Dim density() As Double =

        {

            0, 0, 0, 0, 0,

            0.999965, 0.999941, 0.999902, 0.999849, 0.999781,

            0.9997, 0.999605, 0.999498, 0.999377, 0.999244,

            0.999099, 0.998943, 0.998774, 0.998595, 0.998405,

            0.998203, 0.997992, 0.99777, 0.997538, 0.997296,

            0.997044, 0.996783, 0.996512, 0.996232, 0.995944,

            0.995646, 0.99534, 0.995025, 0.994702, 0.994371,

            0.99403, 0.99368, 0.99333, 0.99296

        }

    Dim tempStr() As String =
```

```

{
    "5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15",
    "16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26",
    "27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37",
    "38"
}

```

```

Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

```

```

    Me.FormBorderStyle = FormBorderStyle.FixedSingle

```

```

    Me.MaximizeBox = False

```

```

    Me.Text = "Program 2.1"

```

```

    Label1.Text = "Temperature (C):"

```

```

    Label2.Text = "The diameter of the capillary tube (m):"

```

```

    Label3.Text = "The angle of contact in degrees:"

```

```

    Label4.Text = "The extent of the liquid rise in the tube (m):"

```

```

    Label5.Text = "Surface tension (N/m):"

```

```

    Label6.Text = "Decimal Places:"

```

```

    ComboBox1.Items.Clear()

```

```

    ComboBox1.Items.AddRange(tempStr)

```

```

    ComboBox1.SelectedIndex = 0

```

```

    NumericUpDown1.Maximum = 15

```

```

    NumericUpDown1.Minimum = 0

```

```

    NumericUpDown1.Value = 8

```

```
End Sub
```

```
Private Sub TextBox1_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox1.TextChanged
```

```
    calculateResults()
```

```
End Sub
```

```
Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox2.TextChanged
```

```
    calculateResults()
```

```
End Sub
```

```
Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles TextBox3.TextChanged
```

```
    calculateResults()
```

```
End Sub
```

```
Private Sub NumericUpDown1_ValueChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles NumericUpDown1.ValueChanged
```

```
    calculateResults()
```

```
End Sub
```

```
Sub calculateResults()
```

```
    Dim s, h, rho, D, phi As Double
```

```

    Dim T As Integer = ComboBox1.SelectedIndex

    If T = -1 Then

        MsgBox("Please select temperature from the list.", vbOKOnly Or
vbCritical)

        Exit Sub

    End If

    T += 5

    rho = density(T) * 1000

    D = Val(TextBox1.Text)

    phi = Val(TextBox2.Text)

    h = Val(TextBox3.Text)

    phi = phi * PI / 180    'in radians

    s = h * g * D * rho / (4 * Cos(phi))

    TextBox4.Text = FormatNumber(s, NumericUpDown1.Value)

End Sub

Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles ComboBox1.SelectedIndexChanged

    calculateResults()

End Sub

End Class

```


2)

Solution

1. Relate the Centigrade temperature to the Fahrenheit temperature using equation:

$$^{\circ}\text{C} = [5/9](^{\circ}\text{F} - 32)$$

2. For temperatures to be equal: assume temperature = x and substitute in step 1 above:

$$x = [5/9](x-32)$$

This yields $x = -40^{\circ}$.

Program 2.2: Equal Temperatures

```
Public Class Form1

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

        Dim msg As String

        Me.Text = "Program 2.2"

        Me.FormBorderStyle = FormBorderStyle.FixedSingle

        Me.MaximizeBox = False

        Dim x As Double

        Const fact = 5 / 9

        x = (-fact * 32) / (1 - fact)

        msg = "Equation:" + vbCrLf + "C = [5/9](F - 32)" + vbCrLf + vbCrLf
        msg += "For temperatures to be equal:" + vbCrLf
        msg += "assume temperature = x," + vbCrLf
        msg += "and substitute in step 1 above, giving:" + vbCrLf + vbCrLf
    End Sub
End Class
```

```

        msg += "x = [5/9](x-32)" + vbCrLf
        msg += "Which yields: x = " + FormatNumber(x, 1)
        Label11.Text = msg
    End Sub
End Class

```

3)

Solution

1. Given: volume and specific gravity.
2. Find weight, W in $N = \text{Volume (V)} * \text{specific gravity } (\rho) * \text{gravitational acceleration (g in } m/s^2)$.

Program 2.3: Weight of Liquid

```

Public Class Form1
    'gravitational acceleration
    Const g = 9.81
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Me.Text = "Program 2.3"
        Me.FormBorderStyle = FormBorderStyle.FixedSingle
        Me.MaximizeBox = False
        Label11.Text = "Volume:"
    End Sub
End Class

```

```

        Label2.Text = "Specific Gravity:"

        Label3.Text = "Weight:"

        Button1.Text = "&Calculate"

    End Sub

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

        Dim V, sg, W As Double

        V = Val(TextBox1.Text)

        sg = Val(TextBox2.Text)

        W = V * sg * g

        TextBox3.Text = FormatNumber(W, 2)

    End Sub

End Class

```

4)

Solution

1. Given: viscosity (dynamic and kinematic).
2. Absolute viscosity (Pa*s (Pascal*second)) = viscosity (centipoise) *0.1 Ns/m².
3. Kinematic viscosity ν (m²/s) = absolute viscosity (μ)/ ρ .

Program 2.4: Viscosity

```

Public Class Form1

```

```

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

    Dim msg As String

    msg = "Equations:" + vbCrLf + vbCrLf

    msg += "1. Absolute viscosity (Pa*s) = Viscosity (centipoise)*0.1
Ns/m2"

    msg += vbCrLf

    msg += "2. Kinematic viscosity (m2/s) = Absolute Viscosity/Density"

    Label1.Text = msg

    Label2.Text = "Viscosity (centipoise):"

    Label3.Text = "Density (kg/m3):"

    Label4.Text = "Absolute Viscosity (Pa*s):"

    Label5.Text = "Kinematic Viscosity (m2/s):"

    Button1.Text = "&Calculate"

    Me.Text = "Program 2.4"

    Me.FormBorderStyle = FormBorderStyle.FixedSingle

    Me.MaximizeBox = False

End Sub

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

    Dim Vis, AbsVis, KinVis, Density As Double

    Vis = Val(TextBox1.Text)

    Density = Val(TextBox2.Text)

```

```

    AbsVis = Vis * 0.1

    KinVis = AbsVis / Density

    TextBox3.Text = AbsVis

    TextBox4.Text = KinVis

End Sub

End Class

```

5)

Solution

- 1) Given: Materials found in the periodic table.
- 2) Determine molecular weight of compound as:
 - a. MWbarium acetate ($\text{Ba}(\text{C}_2\text{H}_3\text{O}_2)_2$) = $137 + 2 \cdot (12 \cdot 2 + 1 \cdot 3 + 16 \cdot 2) = 255$
 - b. MWnitrous oxide (N_2O) = $14 \cdot 2 + 16 = 44$
 - c. MWammonium aluminum sulfate ($(\text{NH}_4)_2\text{Al}_2(\text{SO}_4)_4$) = $2 \cdot (14 + 1 \cdot 4) + 27 \cdot 2 + 4 \cdot (32 + 16 \cdot 4)$
= 474
 - d. MWstrontium sulfate (SrSO_4) = $87 + 32 + 16 \cdot 4 = 183$
 - e. MWcopper sulfate, (Blue Vitriol) ($\text{CuSO}_4 \cdot 5\text{H}_2\text{O}$) = $63.5 + 32 + 16 \cdot 4 + 5 \cdot (1 \cdot 2 + 16) =$
249.5
- 3) Determine the equivalent weight as: $\text{EW} = \text{MW}/a$ (valency)
 - a. EWbarium acetate ($\text{Ba}(\text{C}_2\text{H}_3\text{O}_2)_2$) = $(137 + 2 \cdot (12 \cdot 2 + 1 \cdot 3 + 16 \cdot 2))/2 = 127.5$
 - b. EWnitrous oxide (N_2O) = $(14 \cdot 2 + 16)/2 = 22$
 - c. EWammonium aluminum sulfate
 $(\text{NH}_4)_2\text{Al}_2(\text{SO}_4)_4 = 2 \cdot (14 + 1 \cdot 4) + 27 \cdot 2 + 4 \cdot (32 + 16 \cdot 4)/2 = 237$

- d. EWstrontium sulfate (SrSO_4) = $44 + 32 + 16 \times 4/2 = 70$
- e. EWcopper sulfate, (Blue Vitriol) ($\text{CuSO}_4 \cdot 5\text{H}_2\text{O}$) = $63.5 + 32 + 16 \times 4 + 5 \times (1 \times 2 + 16)/2 = 124.75$

Program 2.5: Molecular Weights

```
Public Class Form1
```

```
    Dim PerTab(118) As Double
```

```
    Dim Element(118) As String
```

```
    Const TotalElements = 118
```

```
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
```

```
MyBase.Load
```

```
        'Molecular weights from the periodic table
```

```
        PerTab(0) = 1.008 : PerTab(1) = 4.002602 : PerTab(2) = 6.94
```

```
        PerTab(3) = 9.012182 : PerTab(4) = 10.81 : PerTab(5) = 12.011
```

```
        PerTab(6) = 14.007 : PerTab(7) = 15.999 : PerTab(8) = 18.998
```

```
        PerTab(9) = 20.1797 : PerTab(10) = 22.989 : PerTab(11) = 24.305
```

```
        PerTab(12) = 26.981 : PerTab(13) = 28.085 : PerTab(14) = 30.973
```

```
        PerTab(15) = 32.06 : PerTab(16) = 35.45 : PerTab(17) = 39.948
```

```
        PerTab(18) = 39.0983 : PerTab(19) = 40.078 : PerTab(20) = 44.955
```

```
        PerTab(21) = 47.867 : PerTab(22) = 50.9415 : PerTab(23) = 51.9961
```

```
        PerTab(24) = 54.938 : PerTab(25) = 55.845 : PerTab(26) = 58.933
```

```
        PerTab(27) = 58.6934 : PerTab(28) = 63.546 : PerTab(29) = 65.38
```

PerTab(30) = 69.723 : PerTab(31) = 72.63 : PerTab(32) = 74.9216
 PerTab(33) = 78.96 : PerTab(34) = 79.904 : PerTab(35) = 83.798
 PerTab(36) = 85.4678 : PerTab(37) = 87.62 : PerTab(38) = 88.90585
 PerTab(39) = 91.224 : PerTab(40) = 92.90638 : PerTab(41) = 95.96
 PerTab(42) = 98 : PerTab(43) = 101.07 : PerTab(44) = 102.9
 PerTab(45) = 106.42 : PerTab(46) = 107.8682 : PerTab(47) = 112.411
 PerTab(48) = 114.818 : PerTab(49) = 118.71 : PerTab(50) = 121.76
 PerTab(51) = 127.6 : PerTab(52) = 126.9 : PerTab(53) = 131.293
 PerTab(54) = 132.9 : PerTab(55) = 137.327 : PerTab(56) = 138.9
 PerTab(57) = 140.116 : PerTab(58) = 140.9 : PerTab(59) = 144.242
 PerTab(60) = 145 : PerTab(61) = 150.36 : PerTab(62) = 151.964
 PerTab(63) = 157.25 : PerTab(64) = 158.92 : PerTab(65) = 162.5
 PerTab(66) = 164.93 : PerTab(67) = 167.259 : PerTab(68) = 168.93
 PerTab(69) = 173.054 : PerTab(70) = 174.9668 : PerTab(71) = 178.49
 PerTab(72) = 180.94 : PerTab(73) = 183.84 : PerTab(74) = 186.207
 PerTab(75) = 190.23 : PerTab(76) = 192.217 : PerTab(77) = 195.084
 PerTab(78) = 196.96 : PerTab(79) = 200.59 : PerTab(80) = 204.38
 PerTab(81) = 207.2 : PerTab(82) = 208.98 : PerTab(83) = 209
 PerTab(84) = 210 : PerTab(85) = 222 : PerTab(86) = 223
 PerTab(87) = 226 : PerTab(88) = 227 : PerTab(89) = 232.03
 PerTab(90) = 231.03 : PerTab(91) = 238.02 : PerTab(92) = 237
 PerTab(93) = 244 : PerTab(94) = 243 : PerTab(95) = 247
 PerTab(96) = 247 : PerTab(97) = 251 : PerTab(98) = 252
 PerTab(99) = 257 : PerTab(100) = 258 : PerTab(101) = 259

```

PerTab(102) = 262 : PerTab(103) = 267 : PerTab(104) = 268
PerTab(105) = 271 : PerTab(106) = 272 : PerTab(107) = 270
PerTab(108) = 276 : PerTab(109) = 281 : PerTab(110) = 280
PerTab(111) = 285 : PerTab(112) = 284 : PerTab(113) = 289
PerTab(114) = 288 : PerTab(115) = 293 : PerTab(116) = 294
PerTab(117) = 294

```

'element names

```

Element(0) = "H" : Element(1) = "He" : Element(2) = "Li"
Element(3) = "Be" : Element(4) = "B" : Element(5) = "C"
Element(6) = "N" : Element(7) = "O" : Element(8) = "F"
Element(9) = "Ne" : Element(10) = "Na" : Element(11) = "Mg"
Element(12) = "Al" : Element(13) = "Si" : Element(14) = "P"
Element(15) = "S" : Element(16) = "Cl" : Element(17) = "Ar"
Element(18) = "K" : Element(19) = "Ca" : Element(20) = "Sc"
Element(21) = "Ti" : Element(22) = "V" : Element(23) = "Cr"
Element(24) = "Mn" : Element(25) = "Fe" : Element(26) = "Co"
Element(27) = "Ni" : Element(28) = "Cu" : Element(29) = "Zn"
Element(30) = "Ga" : Element(31) = "Ge" : Element(32) = "As"
Element(33) = "Se" : Element(34) = "Br" : Element(35) = "Kr"
Element(36) = "Rb" : Element(37) = "Sr" : Element(38) = "Y"
Element(39) = "Zr" : Element(40) = "Nb" : Element(41) = "Mo"
Element(42) = "Tc" : Element(43) = "Ru" : Element(44) = "Rh"
Element(45) = "Pd" : Element(46) = "Ag" : Element(47) = "Cd"

```



```
Element(48) = "In" : Element(49) = "Sn" : Element(50) = "Sb"  
Element(51) = "Te" : Element(52) = "I" : Element(53) = "Xe"  
Element(54) = "Cs" : Element(55) = "Ba" : Element(56) = "La"  
Element(57) = "Ce" : Element(58) = "Pr" : Element(59) = "Nd"  
Element(60) = "Pm" : Element(61) = "Sm" : Element(62) = "Eu"  
Element(63) = "Gd" : Element(64) = "Tb" : Element(65) = "Dy"  
Element(66) = "Ho" : Element(67) = "Er" : Element(68) = "Tm"  
Element(69) = "Yb" : Element(70) = "Lu" : Element(71) = "Hf"  
Element(72) = "Ta" : Element(73) = "W" : Element(74) = "Re"  
Element(75) = "Os" : Element(76) = "Ir" : Element(77) = "Pt"  
Element(78) = "Au" : Element(79) = "Hg" : Element(80) = "Tl"  
Element(81) = "Pb" : Element(82) = "Bi" : Element(83) = "Po"  
Element(84) = "At" : Element(85) = "Rn" : Element(86) = "Fr"  
Element(87) = "Ra" : Element(88) = "Ac" : Element(89) = "Th"  
Element(90) = "Pa" : Element(91) = "U" : Element(92) = "Np"  
Element(93) = "Pu" : Element(94) = "Am" : Element(95) = "Cm"  
Element(96) = "Bk" : Element(97) = "Cf" : Element(98) = "Es"  
Element(99) = "Fm" : Element(100) = "Md" : Element(101) = "No"  
Element(102) = "Lr" : Element(103) = "Rf" : Element(104) = "Db"  
Element(105) = "Sg" : Element(106) = "Bh" : Element(107) = "Hs"  
Element(108) = "Mt" : Element(109) = "Ds" : Element(110) = "Rg"  
Element(111) = "Cn" : Element(112) = "Nh" : Element(113) = "Fl"  
Element(114) = "Mc" : Element(115) = "Lv" : Element(116) = "Ts"  
Element(117) = "Og"
```

```

Me.Text = "Program 2.5"

Me.FormBorderStyle = FormBorderStyle.FixedSingle

Me.MaximizeBox = False

Label1.Text = "Enter formula:"

Label2.Text = "Molecular Weight:"

Label3.Text = "Equivalent Weight:"

Button1.Text = "&calculate weights"

End Sub

```

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

```

```

    Dim compoundStr As String = TextBox1.Text

    Dim i As Integer = 0

    Dim len As Integer = compoundStr.Length

    Dim MolWt As Double = 0.0

    Dim tmp As Double = 0.0

    Dim Valency As Double

    While i < len

        'skip any leading white space

        While Char.IsWhiteSpace(compoundStr.Chars(i))

            i += 1

        End While

        'check for braced expressions

```

```

    If compoundStr.Chars(i) = "(" Then
        'error occured during parsing
        If getSegment(compoundStr, i, len, tmp) = 0 Then
            Exit Sub
        End If
        MolWt += tmp
        Continue While
    End If
    'error occured during parsing
    If getElem(compoundStr, i, len, tmp) = 0 Then
        Exit Sub
    End If
    MolWt += tmp
End While

TextBox3.Text = FormatNumber(MolWt, 2)

Valency = Val(TextBox2.Text)

If Valency = 0.0 Then
    Valency = 1.0
End If

TextBox4.Text = FormatNumber(MolWt / Valency, 2)

End Sub

```

```

Public Function getSegment(ByRef compoundStr As String, ByRef i As
Integer, len As Integer, ByRef Result As Double) As Integer

```

```

If compoundStr.Chars(i) <> "(" Then
    Return 0
End If

'skip '('

i += 1

Dim MolWt As Double = 0.0
Dim tmp As Double = 0.0
Dim fact As Integer = 0
While i < len
    'skip any leading white space
    While Char.IsWhiteSpace(compoundStr.Chars(i))
        i += 1
    End While
    'end of braced expression
    If compoundStr.Chars(i) = ")" Then
        Exit While
    End If
    tmp = 0
    'check for braced expressions
    If compoundStr.Chars(i) = "(" Then
        'error occurred during parsing
        If getSegment(compoundStr, i, len, tmp) = 0 Then
            Return 0
        End If
    End If

```

```

Else
    'error occurred during parsing'
    If getElem(compoundStr, i, len, tmp) = 0 Then
        Return 0
    End If
End If

MolWt += tmp

End While

'check for a missing '''
If compoundStr.Chars(i) <> ")" Then
    MsgBox("Error: Missing ''' in formula.", vbCritical Or vbOKOnly)
    Return 0
End If

i += 1

fact = 0

While i < len AndAlso Char.IsDigit(compoundStr.Chars(i))
    fact = (fact * 10) + Integer.Parse(compoundStr.Chars(i))
    i += 1
End While

If fact <> 0 Then
    MolWt *= fact
End If

Result = MolWt

Return 1

```

End Function

```

Public Function getElem(ByRef compoundStr As String, ByRef i As Integer,
len As Integer, ByRef Result As Double) As Integer

    Dim MolWt As Double = 0.0

    Dim arrIndex As Integer

    Dim fact As Integer = 0

    If Char.IsUpper(compoundStr.Chars(i)) Then

        Dim name As String

        If i < len - 1 AndAlso Char.IsLower(compoundStr.Chars(i + 1))
Then
            name = compoundStr.Chars(i) + compoundStr.Chars(i + 1)

            i += 2

        Else
            name = compoundStr.Chars(i)

            i += 1

        End If

        arrIndex = getIndex(name)

        If arrIndex = -1 Then

            MsgBox("Error: Invalid element name: " + name, vbCritical Or
vbOKOnly)

            Return 0

        End If

        MolWt = PerTab(arrIndex)

```

```

        While i < len AndAlso Char.IsDigit(compoundStr.Chars(i))
            fact = (fact * 10) + Integer.Parse(compoundStr.Chars(i))
            i += 1
        End While

        If fact <> 0 Then
            MolWt *= fact
        End If

        Result = MolWt

        Return 1
    End If

    Return 0
End Function

Public Function getIndex(name As String) As Integer
    Dim i As Integer
    For i = 0 To TotalElements - 1
        If Element(i).Equals(name) Then
            Return i
        End If
    Next

    Return -1
End Function
End Class

```

6)

Solution

- 1) Given: wt. $\text{H}_2\text{SO}_4 = 2.5 \text{ g}$, $V = 4 \text{ L}$.
- 2) Find the equivalent weight of the compound H_2SO_4 as: $\text{EW} = \text{MW}/\text{av} = [2*1 + 32 + 4*16]/2 = 98/2 = 49$.
- 3) Use the equation of normality as: $N = (\text{wt}/\text{EW})/V$
- 4) Substitute the values given in step 1 into step 3 above to find the normality of the acid: $N = (2.5\text{g}/49)/4 \text{ L}$, which yields $N = 0.013\text{N}$
- 5) Find the molecular weight of the compound H_2SO_4 as: $\text{MW} = \text{Summation of atomic weights of acid} = [2*1 + 32 + 4*16] = 98$
- 6) Use the equation of molarity as: $M = (\text{wt}/\text{MW})/V$
- 7) Substitute the values given in step 1 into step 3 above to find the molarity of the acid:
- 8) $M = (2.5\text{g}/98)/4 \text{ L}$, which yields $M = 0.0063\text{M}$

Program 2.6: Normality and Molarity

To save space, this program listing is truncated. The parts that are similar to Program 2.5 are marked as such.

```
Public Class Form1
```

```
    Dim PerTab(118) As Double
```

```
    Dim Element(118) As String
```

```
    Const TotalElements = 118
```



```
Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
```

```
    '*****'
```

```
    ' The code that loads the arrays with values from the periodic table
```

```
    ' has been truncated. It can be found from Program 2.5 Listing.
```

```
    '*****'
```

```
    .....
```

```
    Me.Text = "Program 2.6"
```

```
    Me.FormBorderStyle = FormBorderStyle.FixedSingle
```

```
    Me.MaximizeBox = False
```

```
    Label1.Text = "Enter formula:"
```

```
    Label2.Text = "Valency:"
```

```
    Label3.Text = "Weight (g):"
```

```
    Label4.Text = "Volume (L):"
```

```
    Label5.Text = "Normality (N):"
```

```
    Label6.Text = "Molarity (M):"
```

```
    Button1.Text = "&Calculate"
```

```
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click
```

```
    Dim compoundStr As String = TextBox1.Text
```

```
    Dim i As Integer = 0
```

```

Dim len As Integer = compoundStr.Length

Dim MolWt As Double = 0.0

Dim tmp As Double = 0.0

Dim Valency As Double

Dim Wt, Vol As Double

Dim N, M As Double

While i < len

    'skip any leading white space

    While Char.IsWhiteSpace(compoundStr.Chars(i))

        i += 1

    End While

    'check for braced expressions

    If compoundStr.Chars(i) = "(" Then

        'error occured during parsing

        If getSegment(compoundStr, i, len, tmp) = 0 Then

            Exit Sub

        End If

        MolWt += tmp

        Continue While

    End If

    'error occured during parsing

    If getElem(compoundStr, i, len, tmp) = 0 Then

        Exit Sub

    End If

```

```

        MolWt += tmp

    End While

    Valency = Val(TextBox2.Text)

    If Valency = 0.0 Then

        Valency = 1.0

    End If

    Wt = Val(TextBox3.Text)

    Vol = Val(TextBox4.Text)

    N = (Wt / (MolWt / Valency)) / Vol

    M = (Wt / MolWt) / Vol

    TextBox5.Text = FormatNumber(N, 4)

    TextBox6.Text = FormatNumber(M, 4)

End Sub

' ***** '

' The code for getSegment(...), getElem(...) and getIndex(...) functions
' has been truncated. It can be found from Program 2.5 Listing.

' ***** '

.....

End Class

```

7)

Solution:

- 1) Given: fructose. $\text{CH}_2\text{OH}(\text{CHOH})_3\text{COCH}_2\text{OH}$, and compound zincdimethyl-

dithiocarbamate $\text{Zn}[\text{S}_2\text{CN}(\text{CH}_3)_2]_2$.

2) Determine the molecular weight of fructose $\text{CH}_2\text{OH}(\text{CHOH})_3\text{COCH}_2\text{OH}$ as: $\text{MW}_{\text{fructose}}$
 $= 12 \cdot 6 + 1 \cdot 12 = 84$

3) Compute the percentage of elements in fructose as:

4) % C in fructose $= 6 \cdot 12 / 84 = 85.7\%$.

5) Determine the molecular weight of zincdimethyl-dithiocarbamate $\text{Zn}[\text{S}_2\text{CN}(\text{CH}_3)_2]_2$ as:

$\text{MW}_{\text{compound}} = 65 + 2 \cdot (32 \cdot 2 + 12 + 14 + 2 \cdot (12 + 1 \cdot 3)) = 305$

6) Compute the percentage of zinc in zincdimethyl-dithiocarbamate as:

7) % Zn in compound $= (1 \cdot 65 / 305) \cdot 100 = 21.3\%$.

Program 2.7: Elemental Percentages

To save space, this program listing is truncated. The parts that are similar to Program 2.5 are marked as such.

```
Public Class Form1
```

```
    Dim PerTab(118) As Double
```

```
    Dim Element(118) As String
```

```
    Dim reqElementIndex As Integer
```

```
    Dim reqElementCount As Integer
```

```
    Const TotalElements = 118
```

```
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
```

```
MyBase.Load
```

```
'*****'
```

```
' The code that loads the arrays with values from the periodic table  
' has been truncated. It can be found from Program 2.5 Listing.
```

```
'*****'
```

```
.....
```

```
Me.Text = "Program 2.7"
```

```
Me.FormBorderStyle = FormBorderStyle.FixedSingle
```

```
Me.MaximizeBox = False
```

```
Label1.Text = "Enter formula:"
```

```
Label2.Text = "Element Symbol to count its percentage:"
```

```
Label3.Text = "Molecular Weight:"
```

```
Label4.Text = "Percentage of requested element:"
```

```
Button1.Text = "&calculate weights"
```

```
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles  
Button1.Click
```

```
Dim compoundStr As String = TextBox1.Text
```

```
Dim i As Integer = 0
```

```
Dim len As Integer = compoundStr.Length
```

```
Dim MolWt As Double = 0.0
```

```
Dim tmp As Double = 0.0
```

```
Dim Valency As Double
```

```
Dim tmp2 As Integer = 0
```

```

Dim perc As Double

Dim reqElementName As String
reqElementName = TextBox2.Text
reqElementCount = 0
reqElementIndex = getIndex(reqElementName)
If reqElementIndex = -1 Then
    MsgBox("Invalid element symbol: " + reqElementName)
    Exit Sub
End If

While i < len
    'skip any leading white space
    While Char.IsWhiteSpace(compoundStr.Chars(i))
        i += 1
    End While
    'check for braced expressions
    If compoundStr.Chars(i) = "(" Then
        'error occurred during parsing
        If getSegment(compoundStr, i, len, tmp, tmp2) = 0 Then
            Exit Sub
        End If
        MolWt += tmp
        reqElementCount += tmp2
        Continue While
    End If
End While

```

```

        End If

        'error occurred during parsing

        If getElem(compoundStr, i, len, tmp, tmp2) = 0 Then

            Exit Sub

        End If

        MolWt += tmp

        reqElementCount += tmp2

    End While

    TextBox3.Text = FormatNumber(MolWt, 2)

    perc = reqElementCount * PerTab(reqElementIndex) / MolWt

    TextBox4.Text = FormatNumber(perc * 100, 2) + "%"

End Sub

```

```

Public Function getSegment(ByRef compoundStr As String, ByRef i As
Integer, len As Integer, ByRef Result As Double, ByRef Result2 As Integer) As
Integer

```

```

    If compoundStr.Chars(i) <> "(" Then

```

```

        Return 0

```

```

    End If

```

```

    'skip '('

```

```

    i += 1

```

```

    Dim MolWt As Double = 0.0

```

```

    Dim tmp As Double = 0.0

```

```

    Dim fact As Integer = 0

```

```

Dim count As Integer = 0
Dim tmp2 As Integer = 0
While i < len
    'skip any leading white space
    While Char.IsWhiteSpace(compoundStr.Chars(i))
        i += 1
    End While
    'end of braced expression
    If compoundStr.Chars(i) = ")" Then
        Exit While
    End If
    tmp = 0
    tmp2 = 0
    'check for braced expressions
    If compoundStr.Chars(i) = "(" Then
        'error occurred during parsing
        If getSegment(compoundStr, i, len, tmp, tmp2) = 0 Then
            Return 0
        End If
    Else
        'error occurred during parsing
        If getElem(compoundStr, i, len, tmp, tmp2) = 0 Then
            Return 0
        End If
    End If
End While

```



```

        End If

        MolWt += tmp

        count += tmp2

    End While

    'check for a missing ')'

    If compoundStr.Chars(i) <> ")" Then

        MsgBox("Error: Missing ')' in formula.", vbCritical Or vbOKOnly)

        Return 0

    End If

    i += 1

    fact = 0

    While i < len AndAlso Char.IsDigit(compoundStr.Chars(i))

        fact = (fact * 10) + Integer.Parse(compoundStr.Chars(i))

        i += 1

    End While

    If fact <> 0 Then

        MolWt *= fact

        count *= fact

    End If

    Result = MolWt

    Result2 = count

    Return 1

End Function

```

```

Public Function getElem(ByRef compoundStr As String, ByRef i As Integer,
len As Integer, ByRef Result As Double, ByRef Result2 As Integer) As Integer

    Dim MolWt As Double = 0.0

    Dim arrIndex As Integer

    Dim fact As Integer = 0

    Dim count As Integer = 0

    If Char.IsUpper(compoundStr.Chars(i)) Then

        Dim name As String

        If i < len - 1 AndAlso Char.IsLower(compoundStr.Chars(i + 1))
Then
            name = compoundStr.Chars(i) + compoundStr.Chars(i + 1)

            i += 2

        Else

            name = compoundStr.Chars(i)

            i += 1

        End If

        arrIndex = getIndex(name)

        If arrIndex = -1 Then

            MsgBox("Error: Invalid element name: " + name, vbCritical Or
vbOKOnly)

            Return 0

        End If

        MolWt = PerTab(arrIndex)

        If arrIndex = reqElementIndex Then

```

```

        count = 1

    End If

    While i < len AndAlso Char.IsDigit(compoundStr.Chars(i))

        fact = (fact * 10) + Integer.Parse(compoundStr.Chars(i))

        i += 1

    End While

    If fact <> 0 Then

        MolWt *= fact

        count *= fact

    End If

    Result = MolWt

    Result2 = count

    Return 1

End If

Return 0

End Function

```

```

'*****'

```

```

' The code for getIndex(...) function

```

```

' has been truncated. It can be found from Program 2.5 Listing.

```

```

'*****'

```

```

.....

```

```

End Class

```

8)

Solution:

1) Given: $[H^+]$ or $[OH^-]$ ion concentration.

2) Find the hydromium ion concentration using equation:

$$pH = -\log\{H^+\} = \log 1/[H^+]$$

3) Substitute the values given in step 1 into equation to determine the pH of the solution:

4) Otherwise use equation

$$pOH = 14 - pH \text{ to compute pH value.}$$

5) Use scale 1 to 14 to differentiate between the acidity of different solutions (1 to 7 acidic and 7 to 14 alkaline).

Program 2.8: pH Calculation

```
Public Class Form1
```

```
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
```

```
MyBase.Load
```

```
        Me.Text = "Program 2.8"
```

```
        Me.FormBorderStyle = FormBorderStyle.FixedSingle
```

```
        Me.MaximizeBox = False
```

```
        RadioButton1.Text = "[H+]:"
```

```
        RadioButton2.Text = "[OH-]:"
```

```
        RadioButton1.Checked = True
```

```
        RadioButton2.Checked = False
```

```
        TextBox2.Enabled = False
```

```

Label1.Text = "pH:"

Button1.Text = "&Calculate pH"

End Sub

```

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

```

```

    Dim conc As Double

    Dim pH As Double

    Dim msg As String

    If RadioButton1.Checked Then

        conc = Val(TextBox1.Text)

        pH = -Math.Log10(conc)

    Else

        conc = Val(TextBox1.Text)

        Dim pOH As Double

        pOH = -Math.Log10(conc)

        pH = 14.0 - pOH

    End If

    msg = FormatNumber(pH, 1)

    If pH >= 1.0 And pH < 7.0 Then

        msg += " (Acidic)"

    ElseIf pH >= 7.0 And pH <= 14 Then

        msg += " (Alkaline)"

    Else msg += " (Out of range)"

```

```

        End If

        TextBox3.Text = msg
    End Sub

    Private Sub RadioButton1_CheckedChanged(sender As Object, e As EventArgs)
        Handles RadioButton1.CheckedChanged

        If RadioButton1.Checked Then

            TextBox1.Enabled = True
            TextBox2.Enabled = False

        End If
    End Sub

    Private Sub RadioButton2_CheckedChanged(sender As Object, e As EventArgs)
        Handles RadioButton2.CheckedChanged

        If RadioButton2.Checked Then

            TextBox2.Enabled = True
            TextBox1.Enabled = False

        End If
    End Sub
End Class

```

9)

Solution:

- 1) Given: concentration of cations involving calcium, magnesium, sodium, strontium and

iron, and anions containing: chloride, sulfate, nitrate, carbonate and bicarbonate, the total hardness and percent experimental error.

- 2) Find concentration of different substances in terms of milliequivalent/L by dividing given concentrations in mg/L, by the equivalent weight of each substance.
- 3) Convert the concentrations expressed as milliequivalent/L calculated in 2 above to mg/L as CaCO_3 . This is to be done by multiplying the concentrations of equivalent/L by the equivalent weight of CaCO_3 . $\text{EW of CaCO}_3 = (40+12+16*3)/2 = 50$. Tabulate your results.
- 4) Use value of experimental error to identify missing concentration of anion or cation as deemed necessary.
- 5) Use value of total hardness = $\text{Ca}^{++} + \text{Mg}^{++} + \text{Sr}^{++} = \text{mg/L CaCO}_3$ in your computations where appropriate.

Program 2.9: Ion Concentration

```
Public Class Form1
```

```
    Dim ions() As String =
```

```
    {
```

```
        "Ca++", "Mg++", "Sr++", "Fe++",
```

```
        "HCO3-", "SO4-", "Cl-", "NO3-", "SiO3-"
```

```
    }
```

```
    Dim EW() As Double =
```

```
    {
```

```

        20, 12.15, 44, 28,
        61, 48, 35.5, 62, 76
    }

    Const totalIons = 9

    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles
MyBase.Load

        Me.Text = "Program 2.9"

        Me.FormBorderStyle = FormBorderStyle.FixedSingle

        Me.MaximizeBox = False

        Label1.Text = "Fill in the cation and anion concentration in the
table," + vbCrLf

        Label1.Text += "enter a value for percent experimental error," + vbCrLf
        Label1.Text += "then click Calculate." + vbCrLf

        Label1.Text += "The first empty ion will be considered the"
        Label1.Text += " value to be calculated"

        Label2.Text = "Percent experimental error:"

        Button1.Text = "&Calculate"

        DataGridView1.Rows.Clear()

        DataGridView1.Columns.Clear()

        DataGridView1.Columns.Add("ion", "Ions")

        DataGridView1.Columns.Add("concMgL", "Conc. mg/L")

        DataGridView1.Rows.Add(totalIons)

        Dim i As Integer

```



```

For i = 0 To totalIons - 1
    DataGridView1.Rows(i).Cells("ion").Value = ions(i)
Next

DataGridView1.AllowUserToAddRows = False
DataGridView1.AllowUserToDeleteRows = False
DataGridView1.AllowUserToOrderColumns = False

End Sub

```

```

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click

```

```

    If DataGridView1.ColumnCount <> 5 Then
        DataGridView1.Columns.Add("EW", "EW mg/meq")
        DataGridView1.Columns.Add("concMeqL", "Conc. meq/L")
        DataGridView1.Columns.Add("concCaCO3", "Conc. mg/L CaCO3")
    End If

    Dim i As Integer
    Dim missingValIndex As Integer = -1
    Dim mgL, meqL, CaCO3 As Double
    Dim totMeqL As Double = 0.0
    Dim totCations As Double = 0
    Dim totAnions As Double = 0

    For i = 0 To totalIons - 1
        DataGridView1.Rows(i).Cells("EW").Value = EW(i).ToString
        mgL = Val(DataGridView1.Rows(i).Cells("concMgL").Value)

```

```

        If mgL = 0 Then
            If missingValIndex <> -1 Then
                MsgBox("Error: More than one missing value!.", vbCritical
Or vbOKOnly)

                Exit Sub
            End If
            missingValIndex = i
        Else
            meqL = mgL / EW(i)
            DataGridView1.Rows(i).Cells("concMeqL").Value =
FormatNumber(meqL, 2)
            CaCO3 = meqL * 50
            DataGridView1.Rows(i).Cells("concCaCO3").Value =
FormatNumber(CaCO3, 2)
            totMeqL += meqL
            If i < 4 Then
                totCations += meqL
            Else
                totAnions += meqL
            End If
        End If
    Next
    If missingValIndex = -1 Then
        Exit Sub
    End If

```

```

End If

'*****

'calculate value of x

'*****

Dim x, y As Double

Dim err As Double = Val(TextBox1.Text)

If err > 1 Then

    err /= 100

End If

If missingValIndex < 4 Then

    'missing is cation

    Dim a As Double = (err * totCations) - totCations + totAnions

    Dim b As Double = 1 - err

    x = a / b

Else

    'missing is anion

    x = totCations - totAnions - (err * totCations)

End If

y = x * EW(missingValIndex)

DataGridView1.Rows(missingValIndex).Cells("concMgL").Value =
FormatNumber(y, 2)

DataGridView1.Rows(missingValIndex).Cells("concMeqL").Value =
FormatNumber(x, 2)

CaCO3 = x * 50

```

```

        DataGridView1.Rows(missingValIndex).Cells("concCaCO3").Value =
FormatNumber(CaCO3, 2)

    End Sub

End Class

```

10)

Solution:

- Given: concentration of ions and experimental error of 3% .
- Determine concentration of given species in units of milliequivalent/L by dividing given concentration (mg/L), by the equivalent weight of each substance, i.e. $C_{\text{meq}} = C_{\text{mg/L}}/\text{EW}$.
- Convert concentrations (milliequivalent/L) to units of mg/L as CaCO_3 . This is achieved by multiplying concentrations of milliequivalent/L by equivalent weight (EW) of CaCO_3 .

Equivalent weight of calcium carbonate = molecular weight of calcium carbonate
(MW)/valency (Z), i.e.,

$$\text{EW of CaCO}_3 = \text{MW}/Z = (40+12+16*3)/2 = 50.$$

Tabulate results as in the table shown below:

Constituent	EW	Concentration		
	mg/meq	C, mg/L	c = C/EW, meq/L	c*50, (mg/L CaCO ₃)
Cations				
Ca ⁺⁺	20	35	1.75	87.5

Mg ⁺⁺	12.15	40	3.29	165
Sr ⁺⁺	44	9	0.2	10
Fe ⁺⁺	28	23	0.82	41
		Sum	6.06	

Anions

HCO ₃ ⁻	61	122	2	100
SO ₄ ⁻⁻	48	34	0.71	35
Cl ⁻	35.5	y	y/35.5	
NO ₃ ⁻	62	22	0.35	18
SiO ₃ ⁻	76	14	0.18	9
		Sum	3.24+y/35.5	

- For a percent error of 3% = Sum of cations – sum of anions/sum of cations

$$\text{Error} = 0.03 = (6.06 - (3.24 + x))/6.06$$

This yields, $x = 2.638 \text{ meq/l}$

Thus chloride concentration $y = 2.638 * 35.5 = 52.85 \text{ mg Cl}^-/\text{l}$.

This gives $2.638 * 50 = 131.9 \text{ mg/L}$ as CaCO_3 .

(Check error = $(6.06 - 3.24 - 2.638)/6.06 = 3\%$ O.K.)

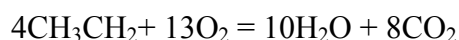
11)

Solution:

- Given: either $\text{CH}_3\text{CH}_2 = 0.1\text{M}$, glycerol $\text{CH}_2\text{OH}.\text{CHOH}.\text{CH}_2\text{OH} = 5\text{g}$ and uric acid

$\text{C}_5\text{H}_4\text{O}_3\text{N}_4 = 20 \text{ mg/L}$.

- 2) Determine the number of oxygen moles that react with ether as shown by the reaction equation:



Find the molecular weight of ether = $(12 \cdot 2 + 1 \cdot 5) = 29$.

The equation reveals that $4 \cdot 29$ of ether reacts with $13 \cdot 32$ of Oxygen.

Therefore, 0.1M ether reacts with x of O_2 .

Thus, the amount of Oxygen needed to oxidize ether is, $x = 0.1 \cdot (13 \cdot 32 \cdot 1000) / 116 = 358.62 \text{ mg/L}$.

- 3) The aerobic oxidation of glycerol may be represented by the following reaction equation:



Determine the molecular weight of glycerol

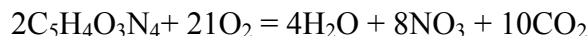
MW glycerol = $3 \cdot 12 + 8 \cdot 1 + 3 \cdot 16 = 84$.

The equation reveals that $2 \cdot 84$ moles glycerol reacts with $7 \cdot 32$ moles of O_2

Therefore, 5 g of glycerol reacts with y moles O_2 ,

Thus, the amount of Oxygen needed to oxidize the glycerol may be computed as: $y = 7 \cdot 32 \cdot 1000 / 168 = 1333.3 \text{ mg/L}$.

- 4) The aerobic oxidation of uric acid may be represented by the following reaction equation:



Find the molecular weight of uric acid = $12 \cdot 5 + 4 \cdot 1 + 3 \cdot 16 + 4 \cdot 14 = 168$.

The equation reveals that $2 \cdot 168$ moles Hexane reacts with $21 \cdot 32$ moles of O_2

Therefore, 20 mg of uric acid reacts with z moles O_2 ,

Thus, the amount of Oxygen needed to oxidize the uric acid may be computed as: $z = (21 \cdot 32) / 336 = 2 \text{ mg/L}$.

5) Determine the total BOD as:

Total BOD = amount of oxygen needed to oxidize either + amount of oxygen needed to oxidize glycerol + amount of Oxygen needed to oxidize uric acid = $x + y + z = 358.62 + 1333.3 + 2 = 1693.92$ mg/L.

Chapter Three: Computer Modeling Applications for Water Resources, Usage, Groundwater and Water Storage and Distribution

3.9 Homework Problems in Water Resources, Water Storage and Distribution

3.9.1 Discussion Problems

1)

Solution:

See Section 3.2 of Chapter 3 of the book.

2)

Solution:

See Section 3.2 of Chapter 3 of the book.

3)

Solution:

Transmissibility: rate at which water of prevailing kinematic viscosity is transmitted through a unit width of an aquifer under a unit hydraulic gradient. Equation 3.14 may be used for determination of transmissibility of an aquifer.

$$TR = kh \quad (3.14)$$

Where:

TR = Transmissibility of aquifer, $m^3/d \cdot m$.

k = Permeability coefficient, m/d.

h = Saturated thickness of aquifer, m.