รายการอ้างอิง

Chen, H.T., Hsieh, T.K., and Lee, H.C. 1977. Separation of Proteins Via Semicontinuous pH Parametric Pumping. AIChE Journal, vol. 23, No.5 .

----------., Pancharoen, U., Yang, W.T., and Kerobo, C.O. 1980. Separation of Proteins Via pH Parametric Pumping. Separation Scince and Technology.

----------., Pancharoen, U., Yang, W.T., and Kerobo, C.O. 1980. Separation of Proteins Via Multicolumn pH Parametric Pumping. AIChE Journal, vol. 26, No.5 .

James, M., and Neuhaus, W. 1982. Human Biochemistry : Specialized tissues and body fluid. 10th edition. New York : The C.V. Mosby Company.

Shaffer, A.G., and Hamrin, C.E. 1975. Enzyme Separation by Parametric Pumping. AIChE Journal, vol.21, No.4.

Thomas, M. 1992. Textbook of Biochemistry with Clinical Correlation : Protein Composition and Structure. 3rd edition. Willey-Liss.

ภาคผนวก

ภาคผนวก ก

โปรแกรมคอมพิวเตอร์

ก.1 โปรแกรมสำหรับการคำนวณหาความเข้มข้นในแต่ละรอบของการทดลองของรูปแบบที่ 1.

```
MODE-1: Program Two Column Parametric pumping System
Uses Crt, Printer;
Var
  a,n,ch,RE : integer;
  VL,VS,Yto,Yb1,Ymo,Kdp1,Kdp2,Kup1,Kup2 : real;
  Yt,Yb,Ym,Kp1s,Kp1m,Kp2s,Kp2m : Array[0..100,1..2] of real;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Initial data;
  begin
    Kp1s[0,1]      := Kdp1;        Kp1m[0,1]      := Kup1;
    Kp2s[0,1]      := Kup2;        Kp2m[0,1]      := Kdp2;
    Kp1s[0,2]      := Kdp1;        Kp1m[0,2]      := Kup1;
    Kp2s[0,2]      := Kup2;        Kp2m[0,2]      := Kdp2;
    Yt[0,1]        := Yto;         Yb[1,1]        := Yb1;
    Ym[0,1]        := Ymo;         Yt[0,2]        := Yto;
    Yb[1,2]        := Yb1;         Ym[0,2]        := Ymo;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Exp-Data;
  begin
    n:= 1;
  While n <= RE Do
  begin
  write('   Yt[',n,',1] = ');              read(Yt[n,1]);
  write('   Yb[',(n+1),',1] = ');          readln(Yb[(n+1),1]);
```

```
    n:= n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Trans-Ym;

```
  begin
          n :=  1;
  While n <= RE DO
  begin
       Ym[(2*n)-1,1]    := Ym[(2*n)-1,2];
       Ym[(2*n),1]      := Ym[(2*n),2];
          n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure ShowTrans-Ym;

```
  begin
          n :=  1;
  While n <= RE DO
  begin
    writeln('       Ym[',(2*n)-1,',1]       := ',Ym[(2*n)-1,2]:8:5);
    writeln('       Ym[',(2*n),',1]         := ',Ym[(2*n),2]:8:5);
          n := n+1;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Avg-Kp;

```
  begin
```

```
        n :=  1;
While n <= RE DO
begin
    Kp1s[n,1]       := (Kp1s[n,1]+Kp1s[n,2])/2;
    Kp2s[n,1]       := (Kp2s[n,1]+Kp2s[n,2])/2;
    Kp1m[n,1]       := (Kp1m[n,1]+Kp1m[n,2])/2;
    Kp2m[n,1]       := (Kp2m[n,1]+Kp2m[n,2])/2;
       n := n+1;
end;
end;
```

{----------}

Procedure Trans-Kp;

```
begin
        n :=  1;
While n <= RE DO
begin
    Kp1s[n,1]       := Kdp1;        Kp2s[n,1]       := Kup2;
    Kp1m[n,1]       := Kup1;        Kp2m[n,1]       := Kdp2;
       n := n+1;
end;
end;
```

{----------}

Procedure ShowTrans-Kp;

```
begin
        n :=  1;
While n <= RE DO
begin
  writeln('      Kp1s[',n,',1]       := ',Kp1s[n,1]:8:5);
  writeln('      Kp2s[',n,',1]       := ',Kp2s[n,1]:8:5);
```

```
    writeln('       Kp1m[',n,',1]       := ',Kp1m[n,1]:8:5);
    writeln('       Kp2m[',n,',1]       := ',Kp2m[n,1]:8:5);
          n := n+1;
  readln;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure ShowAvg-Kp;

```
  begin
          n :=  1;
  While n <= RE DO
  begin
      writeln('       Kp1s[',n,',1]       := ',Kp1s[n,1]:8:5);
      writeln('       Kp2s[',n,',1]       := ',Kp2s[n,1]:8:5);
      writeln('       Kp1m[',n,',1]       := ',Kp1m[n,1]:8:5);
      writeln('       Kp2m[',n,',1]       := ',Kp2m[n,1]:8:5);
         n := n+1;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Estimate;

```
  begin
          n :=  1;
  While n <= RE DO
  begin
      Yt[n,2]           := ((VL*Yt[(n-1),2])+(VS*Kup2*Ym[(2*n)-2,2]))/(VL+(VS*Kdp1));
      Ym[(2*n)-1,2]     := ((VL*Ym[(2*n)-2,2])+(VS*Kup1*Yb[n,2]))/(VL+(VS*Kdp2));
```

```
        Yb[(n+1),2]      := ((VL*Yb[n,2])+(VS*Kdp2*Ym[(2*n)-1,2]))/(VL+(VS*Kup1));
        Ym[(2*n),2]      := ((VL*Ym[(2*n)-1,2])+(VS*Kdp1*Yt[n,2]))/(VL+(VS*Kup2));
            n := n+1;
    end;
    readln;
    end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Show-Estimate;
    begin
    n:= 1;
    While n <= RE Do
      begin
        writeln('   Yt[',n,',2]              = ',Yt[n,2]:8:5);
        writeln('   Ym[',((2*n)-1),',2]      = ',Ym[(2*n)-1,2]:8:5);
        writeln('   Yb[',(n+1),',2]          = ',Yb[(n+1),2]:8:5);
        writeln('   Ym[',(2*n),',2]          = ',Ym[(2*n),2]:8:5);
      n := n+1;
      readln;
      end;
    readln;
    end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Kp-Calculation;
    begin
    n:= 1;
    While n <= RE Do
      begin
        Kp1s[n,2]  := (VL(Yt[n-1,1]-Yt[n,1]))+(VS*Kp2s[n-1,2]*Ym[2n-2,1]) / (VS*Yt[n,1]);
        Kp2m[n,2] := (VL(Ym[2n-2,1]-Ym[2n-1,1]))+(VS*Kp1m[n-1,2]*Yb[n,1]) / (VS*Ym[(2*n)-1,1]);
```

```
        Kp1m[n,2] := (VL(Yb[n,1]-Yb[n+1,1])+(VS*Kp2m[n,2]*Ym[2n-1,1])) / (VS*Yb[n+1,1]);
        Kp2s[n,2]  := (VL(Ym[2n-1,1]-Ym[2n,1]))+(VS*Kp1s[n,2]*Yt[n,1]) / (VS*Ym[(2*n),1]);
           n := n+1;
   end;
   end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure ShowKp-Cal;
   begin
      n :=  1;        While n <= RE DO
   begin
        writeln('       Kp1s[',n,',2]      := ',Kp1s[n,2]:8:5);
        writeln('       Kp2s[',n,',2]      := ',Kp2s[n,2]:8:5);
        writeln('       Kp1m[',n,',2]      := ',Kp1m[n,2]:8:5);
        writeln('       Kp2m[',n,',2]      := ',Kp2m[n,2]:8:5);
           n := n+1;
   end;
   end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Calculation;
   begin
   Trans-Ym;
   Kp-Calculation;
   ShowKp-Cal;
   readln;
   Avg-Kp;
   n := 1;
   While n<= RE Do
      begin
            Yt[n,2]  := ((VL*Yt[(n-1),2])+(VS*Kp2s[(n-1),1]*Ym[(2*n)-2,2])) / (VL+(VS*Kp1s[n,1]));
```

```
        Ym[2n-1,2] := ((VL*Ym[(2*n)-2,2])+(VS*Kp1m[(n-1),1]*Yb[n,2])) / (VL+(VS*Kp2m[n,1]));
        Yb[(n+1),2] := ((VL*Yb[n,2])+(VS*Kp2m[n,1]*Ym[(2*n)-1,2])) / (VL+(VS*Kp1m[n,1]));
        Ym[(2*n),2] := ((VL*Ym[(2*n)-1,2])+(VS*Kp1s[n,1]*Yt[n,2])) / (VL+(VS*Kp2s[n,1]));
      writeln('    Yt[',n,',2]          := ',Yt[n,2]:8:5);
      writeln('    Ym[',(2*n)-1,',2]    := ',Ym[(2*n)-1,2]:8:5);
      writeln('    Yb[',(n+1),',2]      := ',Yb[(n+1),2]:8:5);
      writeln('    Ym[',(2*n),',2]      := ',Ym[(2*n),2]:8:5);
  writeln;
        n := n+1;
  readln;
    end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Begin {Main}
  clrscr;
  Textcolor(Yellow);
      write ('          VL   = '); readln (VL);
      write ('          VS   = '); readln (VS);
      write ('          RE   = '); readln (RE);
      write ('         Kp1+  = '); readln (Kdp1);
      write ('         Kp2+  = '); readln (Kup2);
      write ('         Kp1-  = '); readln (Kup1);
      write ('         Kp2-  = '); readln (Kdp2);
      write ('         Yto   = '); readln (Yto);
            Yb1 := Ymo := Yto;
  writeln;
  Initial-data;
  writeln; Estimate;
  Show_Estimate;
```

```
Exp_data;
Trans_Kp;
writeln;
  ch := 1;
  While ch > 0 Do
  begin
  Calculation;
  write (' contitue (1/0) := ');
  readln(a);
  ch := a;
  end;
End.{Main}
{-----------------------------------------------------------------------------------------------------------------------------}
```

ก.2 โปรแกรมสำหรับการคำนวณหาความเข้มข้นในแต่ละรอบของการทดลองของรูปแบบที่ 2.

MODE-4 : Program Two Column Parametric pumping System.

```
Uses Crt, Printer;
Var
  a,n,ch,RE : integer;
  VL,YAto,YAb1,YAs1,YAmo,KAdp1,KAdp2,KAdp3,KAup1,KAup2 : real;
  VS,YBto,YBb1,YBs1,YBmo,KBdp1,KBdp2,KBdp3,KBup1,KBup2 : real;
  YAt,YAb,YAm,KAp1s,KAp1m,KAp2s,KAp3s,KAp2m : Array[0..100,1..2] of real;
  YBt,YBm,YBs,KBp1s,KBp1m,KBp2s,KBp3s,KBp2m : Array[0..100,1..2] of real;
{-----------------------------------------------------------------------------------------------------------------------------}
Procedure Initial-data;
  begin
    YAt[0,1]      := YAto;      YBt[0,1]      := YBto;
    YAb[1,1]      := YAb1;      YBs[1,1]      := YBs1;
    YAm[0,1]      := YAmo;      YBm[0,1]      := YBmo;
```

```
YAs[0,1]        := YAs1;        YBb[0,1]        := YBb1;
YAt[0,2]        := YAto;        YBt[0,2]        := YBto;
YAb[1,2]        := YAb1;        YBs[1,2]        := YBs1;
YAm[0,2]        := YAmo;        YBm[0,2]        := YBmo;
YAs[0,2]        := YAs1;        YBb[0,2]        := YBb1;

KAp1s[0,1]      := kAdp1;       KAp1s[0,2]      := kAdp1;
KBp1s[0,1]      := kBdp1;       KBp1s[0,2]      := kBdp1;
KAp2s[0,1]      := kAup2;       KAp2s[0,2]      := kAup2;
KBp2s[0,1]      := kBup2;       KBp2s[0,2]      := kBup2;
KAp3s[0,1]      := kAdp3;       KAp3s[0,2]      := kAdp3;
KBp3s[0,1]      := kBdp3;       KBp2s[0,2]      := kBdp3;
KAp1m[0,1]      := KAup1;       KAp1m[0,2]      := KAup1;
KBp1m[0,1]      := KBup1;       KBp1m[0,2]      := KBup1;
KAp2m[0,1]      := KAdp2;       KAp2m[0,2]      := KAdp2;
KBp2m[0,1]      := KBdp2;       KBp2m[0,2]      := KBdp2;
end;
```

{----------}

Procedure Estimate;

```
begin
        n := 1;
While n <= RE DO
begin
    YAt[n,2]        := ((VL*YAt[(n-1),2])+(VS*KAdp3*YAs[n,2])) / (VL+(VS*KAdp1));
    YAm[(2*n)-1,2]  := ((VL*YAm[(2*n)-2,2])+(VS*KAup1*YAb[n,2])) / (VL+(VS*KAdp2));
    YAb[(n+1),2]    := ((VL*YAb[n,2])+(VS*KAdp2*YAm[(2*n)-1,2])) / (VL+(VS*KAup1));
    YAm[(2*n),2]    := ((VL*YAm[(2*n)-1,2])+(VS*KAdp1*YAt[n,2])) / (VL+(VS*KAup2));
    YAs[(n+1),2]    := ((VL*YAs[n,2])+(VS*KAup2*YAm[(2*n),2])) / (VL+(VS*KAdp3));
```

```
        YBt[n,2]          := ((VL*YBt[(n-1),2])+(VS*KBdp3*YBs[n,2])) / (VL+(VS*KBdp1));
        YBm[(2*n)-1,2]    := ((VL*YBm[(2*n)-2,2])+(VS*KBup1*YBb[n,2])) / (VL+(VS*KBdp2));
        YBb[(n+1),2]      := ((VL*YBb[n,2])+(VS*KBdp2*YBm[(2*n)-1,2])) / (VL+(VS*KBup1));
        YBm[(2*n),2]      := ((VL*YBm[(2*n)-1,2])+(VS*KBdp1*YBt[n,2])) /   (VL+(VS*KBup2));
        YBs[(n+1),2]      := ((VL*YBs[n,2])+(VS*KBup2*YBm[(2*n),2])) / (VL+(VS*KBdp3));
            n := n+1;
    end;
    end;
{-------------------------------------------------------------------------------------------------------------------------------------------------------}
Procedure Show_Estimate;
    begin
    n:= 1;
    While n <= RE Do
      begin
        writeln('   YAt[',n,',2]          = ',YAt[n,2]:8:5);
        writeln('   YAm[',((2*n)-1),',2]  = ',YAm[(2*n)-1,2]:8:5);
        writeln('   YAb[',(n+1),',2]      = ',YAb[(n+1),2]:8:5);
        writeln('   YAm[',(2*n),',2]      = ',YAm[(2*n),2]:8:5);
        writeln('   YAs[',(n+1),',2]      = ',YAs[(n+1),2]:8:5);
        writeln;
        writeln('   YBt[',n,',2]          = ',YBt[n,2]:8:5);
        writeln('   YBm[',((2*n)-1),',2]  = ',YBm[(2*n)-1,2]:8:5);
        writeln('   YBb[',(n+1),',2]      = ',YBb[(n+1),2]:8:5);
        writeln('   YBm[',(2*n),',2]      = ',YBm[(2*n),2]:8:5);
        writeln('   YBs[',(n+1),',2]      = ',YBs[(n+1),2]:8:5);
      n := n+1;
      end;
    end;
{-------------------------------------------------------------------------------------------------------------------------------------------------------}
```

```
Procedure Exp_Data;
  begin
    n:= 1;
  While n <= RE Do
  begin
       write('   YAt[',n,',1]          = ');        read(YAt[n,1]);
       write('   YAb[',(n+1),',1]      = ');        readln(YAb[(n+1),1]);
       writeln;
       write('   YBt[',n,',1]          = ');        read(YBt[n,1]);
       write('   YBs[',(n+1),',1]      = ');        readln(YBs[(n+1),1]);
       writeln;
    n:= n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Trans_Kp;
  begin
         n :=  1;
  While n <= RE DO
  begin
       KAp1s[n,1]    := KAdp1;     KBp1s[n,1]    := KBdp1;
       KAp2s[n,1]    := KAup2;     KBp2s[n,1]    := KBup2;
       KAp1m[n,1]    := KAup1;     KBp1m[n,1]    := KBup1;
       KAp2m[n,1]    := KAdp2;     KBp2m[n,1]    := KBdp2;
       KAp3s[n,1]    := KAdp3;     KBp3s[n,1]    := KBdp3;
         n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

```
Procedure ShowTrans_Kp;
  begin
        n :=  1;
  While n <= RE DO
  begin
    writeln('       KAp1s[',n,',1]      := ',KAp1s[n,1]:8:5);
    writeln('       KAp2s[',n,',1]      := ',KAp2s[n,1]:8:5);
    writeln('       KAp1m[',n,',1]      := ',KAp1m[n,1]:8:5);
    writeln('       KAp2m[',n,',1]      := ',KAp2m[n,1]:8:5);
    writeln('       KAp3s[',n,',1]      := ',KAp3s[n,1]:8:5);
    writeln;
    writeln('       KBp1s[',n,',1]      := ',KBp1s[n,1]:8:5);
    writeln('       KBp2s[',n,',1]      := ',KBp2s[n,1]:8:5);
    writeln('       KBp1m[',n,',1]      := ',KBp1m[n,1]:8:5);
    writeln('       KBp2m[',n,',1]      := ',KBp2m[n,1]:8:5);
    writeln('       KBp3s[',n,',1]      := ',KBp3s[n,1]:8:5);
        n := n+1;
  readln;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Trans_Ym;
  begin
        n :=  1;
  While n <= RE DO
  begin
      YAm[(2*n)-1,1]          := YAm[(2*n)-1,2];
      YAm[(2*n),1]            := YAm[(2*n),2];
      YAs[(n+1),1]            := YAs[(n+1),2];
```

```
        YBm[(2*n)-1,1]          := YBm[(2*n)-1,2];
        YBm[(2*n),1]            := YBm[(2*n),2];
        YBb[(n+1),1]            := YBb[(n+1),2];
           n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------------}
```

Procedure ShowTrans_Ym;

```
  begin
          n :=  1;
  While n <= RE DO
  begin
    writeln('       YAm[',(2*n)-1,',1]    := ',YAm[(2*n)-1,2]:8:5);
    writeln('       YAm[',(2*n),',1]      := ',YAm[(2*n),2]:8:5);
    writeln('       YAs[',(n+1),',1]      := ',YAs[(2*n),2]:8:5);
    writeln;
    writeln('       YBm[',(2*n)-1,',1]    := ',YBm[(2*n)-1,2]:8:5);
    writeln('       YBm[',(2*n),',1]      := ',YBm[(2*n),2]:8:5);
    writeln('       YBb[',(n+1),',1]      := ',YBb[(2*n),2]:8:5);
          n := n+1;
    readln;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Kp_Calculation;

```
  begin
  n:= 1;    While n <= RE Do
   begin
```

```
KAp1s[n,2]  := ((VL*(YAt[(n-1),1]-YAt[n,1]))+(VS*KAp3s[(n-1),2]*YAs[n,1]))/(VS*YAt[n,1]);
KAp2m[n,2] := (VL*(YAm[2n-2,1]-YAm[2n-1,1]))+(VS*KAp1m[n-1,2]*YAb[n,1])/(VS*YAm[(2*n)-1,1]);
KAp1m[n,2] := ((VL*(YAb[n,1]-YAb[n+1,1]))+(VS*KAp2m[n,2]*YAm[(2*n)-1,1]))/(VS*YAb[(n+1),1]);
KAp2s[n,2]  := ((VL*(YAm[2n-1,1]-YAm[2n,1]))+(VS*KAp1s[n,2]*YAt[n,1]))/(VS*YAm[(2*n),1]);
KAp3s[n,2]  := ((VL*(YAs[n,1]-YAs[(n+1),1]))+(VS*KAp2s[n,2]*YAm[(2*n),1]))/(VS*YAs[(n+1),1]);

KBp1s[n,2]  := ((VL*(YBt[(n-1),1]-YBt[n,1]))+(VS*KBp3s[(n-1),2]*YBs[n,1]))/(VS*YBt[n,1]);
KBp2m[n,2]:=(VL*(YBm[2n-2,1]-YBm[2n-1,1]))+(VS*KBp1m[(n-1),2]*YBb[n,1])/(VS*YBm[(2*n)-1,1]);
KBp1m[n,2] := ((VL*(YBb[n,1]-YBb[(n+1),1]))+(VS*KBp2m[n,2]*YBm[2n-1,1]))/(VS*YBb[(n+1),1]);
KBp2s[n,2] := ((VL*(YBm[(2*n)-1,1]-YBm[(2*n),1]))+(VS*KBp1s[n,2]*YBt[n,1]))/(VS*YBm[(2*n),1]);
KBp3s[n,2] := ((VL*(YBs[n,1]-YBs[(n+1),1]))+(VS*KBp2s[n,2]*YBm[(2*n),1]))/(VS*YBs[(n+1),1]);
          n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure ShowKp_Cal;
  begin
          n :=  1;
  While n <= RE DO
  begin
    writeln('       KAp1s[',n,',2]       := ',KAp1s[n,2]:8:5);
    writeln('       KAp2s[',n,',2]       := ',KAp2s[n,2]:8:5);
    writeln('       KAp3s[',n,',2]       := ',KAp3s[n,2]:8:5);
    writeln('       KAp1m[',n,',2]       := ',KAp1m[n,2]:8:5);
    writeln('       KAp2m[',n,',2]       := ',KAp2m[n,2]:8:5);
    writeln;
    writeln('       KBp1s[',n,',2]       := ',KBp1s[n,2]:8:5);
    writeln('       KBp2s[',n,',2]       := ',KBp2s[n,2]:8:5);
    writeln('       KBp3s[',n,',2]       := ',KBp3s[n,2]:8:5);
```

```
    writeln('       KBp1m[',n,',2]       := ',KBp1m[n,2]:8:5);
    writeln('       KBp2m[',n,',2]       := ',KBp2m[n,2]:8:5);
          n := n+1;
  readln;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Avg_Kp;

```
  begin
          n :=  1;
  While n <= RE DO
  begin
      KAp1s[n,1]      := (KAp1s[n,1]+KAp1s[n,2])/2;
      KAp2s[n,1]      := (KAp2s[n,1]+KAp2s[n,2])/2;
      KAp3s[n,1]      := (KAp3s[n,1]+KAp3s[n,2])/2;
      KAp1m[n,1]      := (KAp1m[n,1]+KAp1m[n,2])/2;
      KAp2m[n,1]      := (KAp2m[n,1]+KAp2m[n,2])/2;

      KBp1s[n,1]      := (KBp1s[n,1]+KBp1s[n,2])/2;
      KBp2s[n,1]      := (KBp2s[n,1]+KBp2s[n,2])/2;
      KBp3s[n,1]      := (KBp3s[n,1]+KBp3s[n,2])/2;
      KBp1m[n,1]      := (KBp1m[n,1]+KBp1m[n,2])/2;
      KBp2m[n,1]      := (KBp2m[n,1]+KBp2m[n,2])/2;
        n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------------}
```

```
Procedure ShowAvg_Kp;
  begin
        n :=  1;
  While n <= RE DO
  begin
    writeln('       KAp1s[',n,',1]       := ',KAp1s[n,1]:8:5);
    writeln('       KAp2s[',n,',1]       := ',KAp2s[n,1]:8:5);
    writeln('       KAp3s[',n,',1]       := ',KAp3s[n,1]:8:5);
    writeln('       KAp1m[',n,',1]       := ',KAp1m[n,1]:8:5);
    writeln('       KAp2m[',n,',1]       := ',KAp2m[n,1]:8:5);
    writeln;
    writeln('       KBp1s[',n,',1]       := ',KBp1s[n,1]:8:5);
    writeln('       KBp2s[',n,',1]       := ',KBp2s[n,1]:8:5);
    writeln('       KBp3s[',n,',1]       := ',KBp3s[n,1]:8:5);
    writeln('       KBp1m[',n,',1]       := ',KBp1m[n,1]:8:5);
    writeln('       KBp2m[',n,',1]       := ',KBp2m[n,1]:8:5);
        n := n+1;
  readln;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Calculation;
  begin
      Trans_Ym;
      Kp_Calculation;
      ShowKp_Cal;
      Avg_Kp;
      n := 1;
```

```
While n<= RE Do
    begin
YAt[n,2]          := ((VL*YAt[(n-1),2])+(VS*KAp3s[(n-1),1]*YAs[n,2])) / (VL+(VS*KAp1s[n,1]));
YAm[(2*n)-1,2]    := ((VL*YAm[(2*n)-2,2])+(VS*KAp1m[(n-1),1]*YAb[n,2]))/(VL+(VS*KAp2m[n,1]));
YAb[(n+1),2]      := ((VL*YAb[n,2])+(VS*KAp2m[n,1]*YAm[(2*n)-1,2]))/(VL+(VS*KAp1m[n,1]));
YAm[(2*n),2]      := ((VL*YAm[(2*n)-1,2])+(VS*KAp1s[n,1]*YAt[n,2]))/(VL+(VS*KAp2s[n,1]));
YAs[(n+1),2]      := ((VL*YAs[n,2])+(VS*KAp2s[n,1]*YAm[(2*n),2]))/(VL+(VS*KAp3s[n,1]));
YBt[n,2]          := ((VL*YBt[(n-1),2])+(VS*KBp3s[(n-1),1]*YBs[n,2]))/(VL+(VS*KBp1s[n,1]));
YBm[(2*n)-1,2]    := ((VL*YBm[(2*n)-2,2])+(VS*KBp1m[(n-1),1]*YBb[n,2]))/(VL+(VS*KBp2m[n,1]));
YBb[(n+1),2]      := ((VL*YBb[n,2])+(VS*KBp2m[n,1]*YBm[(2*n)-1,2]))/(VL+(VS*KBp1m[n,1]));
YBm[(2*n),2]      := ((VL*YBm[(2*n)-1,2])+(VS*KBp1s[n,1]*YBt[n,2]))/(VL+(VS*KBp2s[n,1]));
YBs[(n+1),2]      := ((VL*YBs[n,2])+(VS*KBp2s[n,1]*YBm[(2*n),2]))/(VL+(VS*KBp3s[n,1]));

        writeln('    YAt[',n,',2]          := ',YAt[n,2]:8:5);
        writeln('    YAm[',(2*n)-1,',2]    := ',YAm[(2*n)-1,2]:8:5);
        writeln('    YAb[',(n+1),',2]      := ',YAb[(n+1),2]:8:5);
        writeln('    YAm[',(2*n),',2]      := ',YAm[(2*n),2]:8:5);
        writeln('    YAs[',(n+1),',2]      := ',YAs[(n+1),2]:8:5);
        writeln;
        writeln('    YBt[',n,',2]          := ',YBt[n,2]:8:5);
        writeln('    YBm[',(2*n)-1,',2]    := ',YBm[(2*n)-1,2]:8:5);
        writeln('    YBb[',(n+1),',2]      := ',YBb[(n+1),2]:8:5);
        writeln('    YBm[',(2*n),',2]      := ',YBm[(2*n),2]:8:5);
        writeln('    YBs[',(n+1),',2]      := ',YBs[(n+1),2]:8:5);
        writeln;
            n := n+1;
    end;
end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

```
Begin {Main}
  clrscr;
  Textcolor(Yellow);
  write ('          VL = ');          readln (VL);
  write ('          VS = ');          readln (VS);
  write ('          RE = ');          readln (RE);
  write ('          KAp1+ = ');       readln (KAdp1);
  write ('          KAp2+ = ');       readln (KAup2);
  write ('          KAp3+ = ');       readln (KAdp3);
  write ('          KAp1- = ');       readln (KAup1);
  write ('          KAp2- = ');       readln (KAdp2);
  writeln;
  write ('          KBp1+ = ');       readln (KBdp1);
  write ('          KBp2+ = ');       readln (KBup2);
  write ('          KBp3+ = ');       readln (KBdp3);
  write ('          KBp1- = ');       readln (KBup1);
  write ('          KBp2- = ');       readln (KBdp2);
  write ('          YAto = ');        readln (YAto);
  write ('          YBto = ');        readln (YBto);
        YAb1    := YAto;   YBb1    := YBto;
        YAmo    := YAto;   YBmo    := YBto;
        YAs1    := YAto;   YBs1    := YBto;
  writeln;
      Initial_data;
      Estimate;
      Show_Estimate;
      Exp_data;
  writeln;
      Trans_Kp;
```

```
  writeln;
    ch := 1;
    While ch > 0 Do
    begin
      Calculation;
    write (' contitue (1/0) := ');
    readln(a);
    ch := a;
    end;
  readln
End.
{-----------------------------------------------------------------------------------------------------------------------------------------------------------}
```

ก.3 โปรแกรมสำหรับการคำนวณหาความเข้มข้นในแต่ละรอบของการทดลองของรูปแบบที่ 3.

```
MODE-2 : Program Two Column Parametric pumping System.
Uses Crt, Printer;
Var
  a,n,ch,RE : integer;
  VL,YAto,YAb1,YAmo,KAdp1,KAdp2,KAdp3,KAup1,KAup2,KAup3 : real;
  VS,YBto,YBb1,YBmo,KBdp1,KBdp2,KBdp3,KBup1,KBup2,KBup3 : real;
  YAt,YAb,YAm,KAp1s,KAp1m,KAp2s,KAp2m,KAp3s,KAp3m,KAp2ss,KAp2mm :
  Array[-1..100,1..2] of real;
  YBt,YBb,YBm,KBp1s,KBp1m,KBp2s,KBp2m,KBp3s,KBp3m,KBp2ss,KBp2mm :
  Array[-1..100,1..2] of real;
{-----------------------------------------------------------------------------------------------------------------------------------------------------------}
Procedure Initial-data;
  begin
      YAt[-1,1]      := YAto;      YBt[-1,1]      := YBto;
      YAm[0,1]       := YAmo;      YBm[0,1]       := YBmo;
```

```
    YAb[0,1]      := YAb1;      YBb[0,1]      := YBb1;
    YAt[0,1]      := YAto;      YBt[0,1]      := YBto;
    YAb[1,1]      := YAb1;      YBb[1,1]      := YBb1;
    YAt[-1,2]     := YAto;      YBt[-1,2]     := YBto;
    YAm[0,2]      := YAmo;      YBm[0,2]      := YBmo;
    YAb[0,2]      := YAb1;      YBb[0,2]      := YBb1;
    YAt[0,2]      := YAto;      YBt[0,2]      := YBto;
    YAb[1,2]      := YAb1;      YBb[1,2]      := YBb1;

    KAp1s[0,1]    := KAdp1;     KAp1s[0,2]    := KAdp1;
    KBp1s[0,1]    := KBdp1;     KBp1s[0,2]    := KBdp1;
    KAp2s[0,1]    := KAup2;     KAp2s[0,2]    := KAup2;
    KBp2s[0,1]    := KBup2;     KBp2s[0,2]    := KBup2;
    KAp3s[0,1]    := KAdp3;     KAp3s[0,2]    := KAdp3;
    KBp3s[0,1]    := KBdp3;     KBp2s[0,2]    := KBdp3;
    KAp1m[0,1]    := KAup1;     KAp1m[0,2]    := KAup1;
    KBp1m[0,1]    := KBup1;     KBp1m[0,2]    := KBup1;
    KAp2m[0,1]    := KAdp2;     KAp2m[0,2]    := KAdp2;
    KBp2m[0,1]    := KBdp2;     KBp2m[0,2]    := KBdp2;
    KAp3m[0,1]    := KAup3;     KAp3m[0,2]    := KAup3;
    KBp3m[0,1]    := KBup3;     KBp3m[0,2]    := KBup3;
    KAp2ss[0,1]   := KAup2;     KAp2ss[0,2]   := KAup2;
    KBp2ss[0,1]   := KBup2;     KBp2ss[0,2]   := KBup2;
    KAp2mm[0,1]   := KAdp2;     KBp2mm[0,2]   := KAdp2;
    KBp2mm[0,1]   := KBdp2;     KBp2mm[0,2]   := KBdp2;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Estimate;

```
  begin
```

```
        n :=  1;      While n <= RE DO
  begin
YAt[(2*n)-1,2]   := ((VL*YAt[(2*n)-3,2])+(VS*KAup2*YAm[(4*n)-4,2]))/(VL+(VS*KAdp1));
YAm[(4*n)-3,2]   := ((VL*YAm[(4*n)-4,2])+(VS*KAup1*YAb[(2*n)-1,2]))/(VL+(VS*KAdp2));
YAb[(2*n),2]     := ((VL*YAb[(2*n)-2,2])+(VS*KAdp2*YAm[(4*n)-3,2]))/(VL+(VS*KAup3));
YAm[(4*n)-2,2]   := ((VL*YAm[(4*n)-3,2])+(VS*KAdp1*YAt[(2*n)-1,2]))/(VL+(VS*KAup2));
YAt[(2*n),2]     := ((VL*YAt[(2*n)-2,2])+(VS*KAup2*YAm[(4*n)-2,2]))/(VL+(VS*KAdp3));
YAm[(4*n)-1,2]   := ((VL*YAm[(4*n)-2,2])+(VS*KAup3*YAb[(2*n),2]))/(VL+(VS*KAdp2));
YAb[(2*n)+1,2]   := ((VL*YAb[(2*n)-1,2])+(VS*KAdp2*YAm[(4*n)-1,2]))/(VL+(VS*KAup1));
YAm[(4*n),2]     := ((VL*YAm[(4*n)-1,2])+(VS*KAdp3*YAt[(2*n),2]))/(VL+(VS*KAup2));

YBt[(2*n)-1,2]   := ((VL*YBt[(2*n)-3,2])+(VS*KBup2*YBm[(4*n)-4,2]))/(VL+(VS*KBdp1));
YBm[(4*n)-3,2]   := ((VL*YBm[(4*n)-4,2])+(VS*KBup1*YBb[(2*n)-1,2]))/(VL+(VS*KBdp2));
YBb[(2*n),2]     := ((VL*YBb[(2*n)-2,2])+(VS*KBdp2*YBm[(4*n)-3,2]))/(VL+(VS*KBup3));
YBm[(4*n)-2,2]   := ((VL*YBm[(4*n)-3,2])+(VS*KBdp1*YBt[(2*n)-1,2]))/(VL+(VS*KBup2));
YBt[(2*n),2]     := ((VL*YBt[(2*n)-2,2])+(VS*KBup2*YBm[(4*n)-2,2]))/(VL+(VS*KBdp3));
YBm[(4*n)-1,2]   := ((VL*YBm[(4*n)-2,2])+(VS*KBup3*YBb[(2*n),2]))/(VL+(VS*KBdp2));
YBb[(2*n)+1,2]   := ((VL*YBb[(2*n)-1,2])+(VS*KBdp2*YBm[(4*n)-1,2]))/(VL+(VS*KBup1));
YBm[(4*n),2]     := ((VL*YBm[(4*n)-1,2])+(VS*KBdp3*YBt[(2*n),2]))/(VL+(VS*KBup2));
  n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Show-Estimate;
  begin
  n:= 1;
  While n <= RE Do
   begin
      writeln('   YAt[',n,',2]              = ',YAt[(2*n)-1,2]:8:5);
```

```
        writeln('   YAm[',(4*n)-3,',2]     = ',YAm[(4*n)-3,2]:8:5);
        writeln('   YAb*[',(n+1),',2]      = ',YAb[(2*n),2]:8:5);
        writeln('   YAm[',(4*n)-2,',2]     = ',YAm[(4*n)-2,2]:8:5);
        writeln('   YAt*[',n,',2]          = ',YAt[(2*n),2]:8:5);
        writeln('   YAm[',(4*n)-1,',2]     = ',YAm[(4*n)-1,2]:8:5);
        writeln('   YAb[',(n+1),',2]       = ',YAb[(2*n)+1,2]:8:5);
        writeln('   YAm[',(4*n),',2]       = ',YAm[(4*n),2]:8:5);
        writeln;
        writeln('   YBt[',n,',2]           = ',YBt[(2*n)-1,2]:8:5);
        writeln('   YBm[',(4*n)-3,',2]     = ',YBm[(4*n)-3,2]:8:5);
        writeln('   YBb*[',(n+1),',2]      = ',YBb[(2*n),2]:8:5);
        writeln('   YBm[',(4*n)-2,',2]     = ',YBm[(4*n)-2,2]:8:5);
        writeln('   YBt*[',n,',2]          = ',YBt[(2*n),2]:8:5);
        writeln('   YBm[',(4*n)-1,',2]     = ',YBm[(4*n)-1,2]:8:5);
        writeln('   YBb[',(n+1),',2]       = ',YBb[(2*n)+1,2]:8:5);
        writeln('   YBm[',(4*n),',2]       = ',YBm[(4*n),2]:8:5);
    n := n+1;
    readln;
    end;
{-------------------------------------------------------------------------------------------------------------------------------------------------------}
Procedure Exp-Data;
    begin
      n:= 1;
    While n <= RE Do
    begin
        write('   YAt[',n,',1]    = ');          read(YAt[(2*n)-1,1]);
        write('   YAb[',(n+1),',1] = ');         readln(YAb[(2*n)+1,1]);
    writeln;
        write('   YBt[',n,',1]    = ');          read(YBt[(2*n),1]);
```

```
        write('    YBb[',(n+1),',1] = ');          readln(YBb[(2*n),1]);
        writeln;
    n:= n+1;
  readln;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Trans-Kp;

```
  begin
        n :=  1;
  While n <= RE DO
  begin
      KAp1s[n,1]    := KAdp1;     KBp1s[n,1]    := KBdp1;
      KAp2s[n,1]    := KAup2;     KBp2s[n,1]    := KBup2;
      KAp2ss[n,1]   := KAup2;     KBp2ss[n,1]   := KBup2;
      KAp3s[n,1]    := KAdp3;     KBp3s[n,1]    := KBdp3;
      KAp1m[n,1]    := KAup1;     KBp1m[n,1]    := KBup1;
      KAp2m[n,1]    := KAdp2;     KBp2m[n,1]    := KBdp2;
      KAp2mm[n,1]   := KAdp2;     KBp2mm[n,1]   := KBdp2;
      KAp3m[n,1]    := KAup3;     KBp3m[n,1]    := KBup3;
    n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure ShowTrans-Kp;

```
  begin
        n :=  1;
  While n <= RE DO
```

```
begin
        writeln('       KAp1s[',n,',1]     := ',KAp1s[n,1]:8:5);
        writeln('       KAp2s[',n,',1]     := ',KAp2s[n,1]:8:5);
        writeln('       KAp2s*[',n,',1]    := ',KAp2ss[n,1]:8:5);
        writeln('       KAp3s[',n,',1]     := ',KAp3s[n,1]:8:5);
        writeln('       KAp1m[',n,',1]     := ',KAp1m[n,1]:8:5);
        writeln('       KAp2m[',n,',1]     := ',KAp2m[n,1]:8:5);
        writeln('       KAp2m*[',n,',1]    := ',KAp2mm[n,1]:8:5);
        writeln('       KAp3m[',n,',1]     := ',KAp3m[n,1]:8:5);
        writeln;
        writeln('       KBp1s[',n,',1]     := ',KBp1s[n,1]:8:5);
        writeln('       KBp2s[',n,',1]     := ',KBp2s[n,1]:8:5);
        writeln('       KBp2s*[',n,',1]    := ',KBp2ss[n,1]:8:5);
        writeln('       KBp3s[',n,',1]     := ',KBp3s[n,1]:8:5);
        writeln('       KBp1m[',n,',1]     := ',KBp1m[n,1]:8:5);
        writeln('       KBp2m[',n,',1]     := ',KBp2m[n,1]:8:5);
        writeln('       KBp2m*[',n,',1]    := ',KBp2mm[n,1]:8:5);
        writeln('       KBp3m[',n,',1]     := ',KBp3m[n,1]:8:5);
          n := n+1;
end;
readln;
end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Trans-Ym;

```
begin
        n :=  1;
While n <= RE DO
begin
            YAm[(4*n)-3,1]          := YAm[(4*n)-3,2];
```

```
            YAb[(2*n),1]          := YAb[(2*n),2];
            YAm[(4*n)-2,1]        := YAm[(4*n)-2,2];
            YAt[(2*n),1]          := YAt[(2*n),2];
            YAm[(4*n)-1,1]        := YAm[(4*n)-1,2];
            YAm[(4*n),1]          := YAm[(4*n),2];

            YBt[(2*n)-1,1]        := YBt[(2*n)-1,2];
            YBm[(4*n)-3,1]        := YBm[(4*n)-3,2];
            YBm[(4*n)-2,1]        := YBm[(4*n)-2,2];
            YBm[(4*n)-1,1]        := YBm[(4*n)-1,2];
            YBb[(2*n)+1,1]        := YBb[(2*n)+1,2];
            YBm[(4*n),1]          := YBm[(4*n),2];
        n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure ShowTrans-Ym;
  begin
        n :=  1;
  While n <= RE DO
  begin
      writeln('     YAm[',(4*n)-3,',1]      := ',YAm[(4*n)-3,2]:8:5);
      writeln('     YAb*[',(n+1),',1]       := ',YAb[(2*n),2]:8:5);
      writeln('     YAm[',(4*n)-2,',1]      := ',YAm[(4*n)-2,2]:8:5);
      writeln('     YAt*[',(n),',1]         := ',YAt[(2*n),2]:8:5);
      writeln('     YAm[',(4*n)-1,',1]      := ',YAm[(4*n)-1,2]:8:5);
      writeln('     YAm[',(4*n),',1]        := ',YAm[(4*n),2]:8:5);
      writeln;
      writeln('     YBt[',n,',1]            := ',YBt[(2*n)-1,2]:8:5);
```

```
        writeln('      YBm[',(4*n)-3,',1]      := ',YBm[(4*n)-3,2]:8:5);
        writeln('      YBm[',(4*n)-2,',1]      := ',YBm[(4*n)-2,2]:8:5);
        writeln('      YBm[',(4*n)-1,',1]      := ',YBm[(4*n)-1,2]:8:5);
        writeln('      YBb[',(n+1),',1]        := ',YBb[(2*n)+1,2]:8:5);
        writeln('      YBm[',(4*n),',1]        := ',YBm[(4*n),2]:8:5);
          n := n+1;
    readln;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------}
```

Procedure Kp-Calculation;

```
  begin
  n:= 1;
  While n <= RE Do
    begin
      KAp1s[n,2]    :=((VL*(YAt[2n-3,1]-YAt[2n-1,1]))+(VS*KAp2ss[n-1,2]*YAm[4n-4,1]))
                    /(VS*YAt[(2*n)-1,1]);
      KAp2m[n,2]    := ((VL*(YAm[4n-4,1]-YAm[4n-3,1]))+(VS*KAp1m[n-1,2]*YAb[2n-1,1]))
                    /(VS*YAm[(4*n)-3,1]);
      KAp3m[n,2]    := ((VL*(YAb[2n-2,1]-YAb[2n,1]))+(VS*KAp2m[n,2]*YAm[4n-3,1]))
                    /(VS*YAb[2n,1]);
      KAp2s[n,2]    := ((VL*(YAm[4n-3,1]-YAm[4n-2,1]))+(VS*KAp1s[n,2]*YAt[2n-1,1]))
                    /(VS*YAm[4n-2,1]);
      KAp3s[n,2]    := ((VL*(YAt[2n-2,1]-YAt[2n,1]))+(VS*KAp2s[n,2]*YAm[4n-2,1]))
                    /(VS*YAt[2n,1]);
      KAp2mm[n,2]   := ((VL*(YAm[4n-2,1]-YAm[4n-1,1]))+(VS*KAp3m[n,2]*YAb[2n,1]))
                    /(VS*YAm[4n-1,1]);
      KAp1m[n,2]    := ((VL*(YAb[2n-1,1]-YAb[2n+1,1]))+(VS*KAp2mm[n,2]*YAm[4n-1,1]))
```

```
                    /(VS*YAb[2n+1,1]);
        KAp2ss[n,2]  := ((VL*(YAm[4n-1,1]-YAm[4n,1]))+(VS*KAp3s[n,2]*YAt[2n,1]))
                    /(VS*YAm[4n,1]);

        KBp1s[n,2]   := ((VL*(YBt[2n-3,1]-YBt[2n-1,1]))+(VS*KBp2ss[n-1,2]*YBm[4n-4,1]))
                    /(VS*YBt[2n-1,1]);
        KBp2m[n,2]   := ((VL*(YBm[4n-4,1]-YBm[4n-3,1]))+(VS*KBp1m[n-1,2]*YBb[2n-1,1]))
                    /(VS*YBm[4n-3,1]);
        KBp3m[n,2]   := ((VL*(YBb[2n-2,1]-YBb[2n,1]))+(VS*KBp2m[n,2]*YBm[4n-3,1]))
                    /(VS*YBb[2n,1]);
        KBp2s[n,2]   := ((VL*(YBm[4n-3,1]-YBm[4n-2,1]))+(VS*KBp1s[n,2]*YBt[2n-1,1]))
                    /(VS*YBm[4n-2,1]);
        KBp3s[n,2]   := ((VL*(YBt[2n-2,1]-YBt[2n,1]))+(VS*KBp2s[n,2]*YBm[4n-2,1]))
                    /(VS*YBt[2n,1]);
        KBp2mm[n,2]  := ((VL*(YBm[4n-2,1]-YBm[4n-1,1]))+(VS*KBp3m[n,2]*YBb[2n,1]))
                    /(VS*YBm[4n-1,1]);
        KBp1m[n,2]   := ((VL*(YBb[2n-1,1]-YBb[2n+1,1]))+(VS*KBp2mm[n,2]*YBm[4n-1,1]))
                    /(VS*YBb[2n+1,1]);
        KBp2ss[n,2]  := ((VL*(YBm[4n-1,1]-YBm[4n,1]))+(VS*KBp3s[n,2]*YBt[2n,1]))
                    /(VS*YBm[4n,1]);
    n := n+1;
  end;
  readln;
  end;
{-------------------------------------------------------------------------------------------------------------------------}
Procedure ShowKp-Cal;
  begin
        n :=  1;
  While n <= RE DO
```

```
begin
        writeln('       KAp1s[',n,',2]     := ',KAp1s[n,2]:8:5);
        writeln('       KAp2s[',n,',2]     := ',KAp2s[n,2]:8:5);
        writeln('       KAp2s*[',n,',2]    := ',KAp2ss[n,2]:8:5);
        writeln('       KAp3s[',n,',2]     := ',KAp3s[n,2]:8:5);
        writeln('       KAp1m[',n,',2]     := ',KAp1m[n,2]:8:5);
        writeln('       KAp2m[',n,',2]     := ',KAp2m[n,2]:8:5);
        writeln('       KAp2m*[',n,',2]    := ',KAp2mm[n,2]:8:5);
        writeln('       KAp3m[',n,',2]     := ',KAp3m[n,2]:8:5);
        writeln;
        writeln('       KBp1s[',n,',2]     := ',KBp1s[n,2]:8:5);
        writeln('       KBp2s[',n,',2]     := ',KBp2s[n,2]:8:5);
        writeln('       KBp2s*[',n,',2]    := ',KBp2ss[n,2]:8:5);
        writeln('       KBp3s[',n,',2]     := ',KBp3s[n,2]:8:5);
        writeln('       KBp1m[',n,',2]     := ',KBp1m[n,2]:8:5);
        writeln('       KBp2m[',n,',2]     := ',KBp2m[n,2]:8:5);
        writeln('       KBp2m*[',n,',2]    := ',KBp2mm[n,2]:8:5);
        writeln('       KBp3m[',n,',2]     := ',KBp3m[n,2]:8:5);
          n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Avg-Kp;

```
  begin
          n :=  1;
  While n <= RE DO
  begin
              KAp1s[n,1]      := (KAp1s[n,1]+KAp1s[n,2])/2;
              KAp2s[n,1]      := (KAp2s[n,1]+KAp2s[n,2])/2;
```

```
                KAp2ss[n,1]     := (KAp2ss[n,1]+KAp2ss[n,2])/2;
                KAp3s[n,1]      := (KAp3s[n,1]+KAp3s[n,2])/2;
                KAp1m[n,1]      := (KAp1m[n,1]+KAp1m[n,2])/2;
                KAp2m[n,1]      := (KAp2m[n,1]+KAp2m[n,2])/2;
                KAp2mm[n,1]     := (KAp2mm[n,1]+KAp2mm[n,2])/2;
                KAp3m[n,1]      := (KAp3m[n,1]+KAp3m[n,2])/2;

                KBp1s[n,1]      := (KBp1s[n,1]+KBp1s[n,2])/2;
                KBp2s[n,1]      := (KBp2s[n,1]+KBp2s[n,2])/2;
                KBp2ss[n,1]     := (KBp2ss[n,1]+KBp2ss[n,2])/2;
                KBp3s[n,1]      := (KBp3s[n,1]+KBp3s[n,2])/2;
                KBp1m[n,1]      := (KBp1m[n,1]+KBp1m[n,2])/2;
                KBp2m[n,1]      := (KBp2m[n,1]+KBp2m[n,2])/2;
                KBp2mm[n,1]     := (KBp2mm[n,1]+KBp2mm[n,2])/2;
                KBp3m[n,1]      := (KBp3m[n,1]+KBp3m[n,2])/2;
            n := n+1;
    end;
    end;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure ShowAvg-Kp;

```
    begin
            n :=  1;
    While n <= RE DO
    begin
        writeln('       KAp1s[',n,',1]     := ',KAp1s[n,1]:8:5);
        writeln('       KAp2s[',n,',1]     := ',KAp2s[n,1]:8:5);
        writeln('       KAp2s*[',n,',1]    := ',KAp2ss[n,1]:8:5);
        writeln('       KAp3s[',n,',1]     := ',KAp3s[n,1]:8:5);
        writeln('       KAp1m[',n,',1]     := ',KAp1m[n,1]:8:5);
```

```
        writeln('       KAp2m[',n,',1]    := ',KAp2m[n,1]:8:5);
        writeln('       KAp2m*[',n,',1]   := ',KAp2mm[n,1]:8:5);
        writeln('       KAp3m[',n,',1]    := ',KAp3m[n,1]:8:5);
        writeln;
        writeln('       KBp1s[',n,',1]    := ',KBp1s[n,1]:8:5);
        writeln('       KBp2s[',n,',1]    := ',KBp2s[n,1]:8:5);
        writeln('       KBp2s*[',n,',1]   := ',KBp2ss[n,1]:8:5);
        writeln('       KBp3s[',n,',1]    := ',KBp3s[n,1]:8:5);
        writeln('       KBp1m[',n,',1]    := ',KBp1m[n,1]:8:5);
        writeln('       KBp2m[',n,',1]    := ',KBp2m[n,1]:8:5);
        writeln('       KBp2m*[',n,',1]   := ',KBp2mm[n,1]:8:5);
        writeln('       KBp3m[',n,',1]    := ',KBp3m[n,1]:8:5);
          n := n+1;
  end;
  end;
{-------------------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Calculation;

```
  begin
       Trans-Ym;
       Kp-Calculation;
       ShowKp-Cal;
  readln;
       Avg-Kp;
  n := 1;
  While n<= RE Do
    begin
       YAt[(2*n)-1,2]    := ((VL*YAt[(2*n)-3,2])+(VS*KAp2ss[(n-1),1]*YAm[(4*n)-4,2]))
                            /(VL+(VS*KAp1s[n,1]));
```

```
YAm[(4*n)-3,2]  := ((VL*YAm[(4*n)-4,2])+(VS*KAp1m[(n-1),1]*YAb[(2*n)-1,2]))
                   /(VL+(VS*KAp2m[n,1]));
YAb[(2*n),2]    := ((VL*YAb[(2*n)-2,2])+(VS*KAp2m[n,1]*YAm[(4*n)-3,2]))
                   /(VL+(VS*KAp3m[n,1]));
YAm[(4*n)-2,2]  := ((VL*YAm[(4*n)-3,2])+(VS*KAp1s[n,1]*YAt[(2*n)-1,2]))
                   /(VL+(VS*KAp2s[n,1]));
YAt[(2*n),2]    := ((VL*YAt[(2*n)-2,2])+(VS*KAp2s[n,1]*YAm[(4*n)-2,2]))
                   /(VL+(VS*KAp3s[n,1]));
YAm[(4*n)-1,2]  := ((VL*YAm[(4*n)-2,2])+(VS*KAp3m[n,1]*YAb[(2*n),2]))
                   /(VL+(VS*KAp2mm[n,1]));
YAb[(2*n)+1,2]  := ((VL*YAb[(2*n)-1,2])+(VS*KAp2mm[n,1]*YAm[(4*n)-1,2]))
                   /(VL+(VS*KAp1m[n,1]));
YAm[(4*n),2]    := ((VL*YAm[(4*n)-1,2])+(VS*KAp3s[n,1]*YAt[(2*n),2]))
                   /(VL+(VS*KAp2ss[n,1]));

YBt[(2*n)-1,2]  := ((VL*YBt[(2*n)-3,2])+(VS*KBp2ss[(n-1),1]*YBm[(4*n)-4,2]))
                   /(VL+(VS*KBp1s[n,1]));
YBm[(4*n)-3,2]  := ((VL*YBm[(4*n)-4,2])+(VS*KBp1m[(n-1),1]*YBb[(2*n)-1,2]))
                   /(VL+(VS*KBp2m[n,1]));
YBb[(2*n),2]    := ((VL*YBb[(2*n)-2,2])+(VS*KBp2m[n,1]*YBm[(4*n)-3,2]))
                   /(VL+(VS*KBp3m[n,1]));
YBm[(4*n)-2,2]  := ((VL*YBm[(4*n)-3,2])+(VS*KBp1s[n,1]*YBt[(2*n)-1,2]))
                   /(VL+(VS*KBp2s[n,1]));
YBt[(2*n),2]    := ((VL*YBt[(2*n)-2,2])+(VS*KBp2s[n,1]*YBm[(4*n)-2,2]))
                   /(VL+(VS*KBp3s[n,1]));
YBm[(4*n)-1,2]  := ((VL*YBm[(4*n)-2,2])+(VS*KBp3m[n,1]*YBb[(2*n),2]))
                   /(VL+(VS*KBp2mm[n,1]));
YBb[(2*n)+1,2]  := ((VL*YBb[(2*n)-1,2])+(VS*KBp2mm[n,1]*YBm[(4*n)-1,2]))
                   /(VL+(VS*KBp1m[n,1]));
```

```
YBm[(4*n),2]    := ((VL*YBm[(4*n)-1,2])+(VS*KBp3s[n,1]*YBt[(2*n),2]))
                   /(VL+(VS*KBp2ss[n,1]));

writeln('   YAt[',n,',2]          := ',YAt[(2*n)-1,2]:8:5);
writeln('   YAm[',(4*n)-3,',2]    := ',YAm[(4*n)-3,2]:8:5);
writeln('   YAb*[',(n+1),',2]     := ',YAb[(2*n),2]:8:5);
writeln('   YAm[',(4*n)-2,',2]    := ',YAm[(4*n)-2,2]:8:5);
writeln('   YAt*[',n,',2]         := ',YAt[(2*n),2]:8:5);
writeln('   YAm[',(4*n)-1,',2]    := ',YAm[(4*n)-1,2]:8:5);
writeln('   YAb[',(n+1),',2]      := ',YAb[(2*n)+1,2]:8:5);
writeln('   YAm[',(4*n),',2]      := ',YAm[(4*n),2]:8:5);
writeln;

writeln('   YBt[',n,',2]          := ',YBt[(2*n)-1,2]:8:5);
writeln('   YBm[',(4*n)-3,',2]    := ',YBm[(4*n)-3,2]:8:5);
writeln('   YBb*[',(n+1),',2]     := ',YBb[(2*n),2]:8:5);
writeln('   YBm[',(4*n)-2,',2]    := ',YBm[(4*n)-2,2]:8:5);
writeln('   YBt*[',n,',2]         := ',YBt[(2*n),2]:8:5);
writeln('   YBm[',(4*n)-1,',2]    := ',YBm[(4*n)-1,2]:8:5);
writeln('   YBb[',(n+1),',2]      := ',YBb[(2*n)+1,2]:8:5);
writeln('   YBm[',(4*n),',2]      := ',YBm[(4*n),2]:8:5);
writeln;
n := n+1;
readln;
end;
end;
{-------------------------------------------------------------------------------------------------------------------------}
```

```
Begin {Main}
  clrscr;
  Textcolor(Yellow);
      write ('        VL    = ');     readln (VL);
      write ('        VS    = ');     readln (VS);
      write ('        RE    = ');     readln (RE);
      write ('      KAp1+   = ');     readln (KAdp1);
      write ('      KAp2+   = ');     readln (KAup2);
      write ('      KAp3+   = ');     readln (KAdp3);
      write ('      KAp1-   = ');     readln (KAup1);
      write ('      KAp2-   = ');     readln (KAdp2);
      write ('      KAp3-   = ');     readln (KAup3);
      writeln;
      write ('      KBp1+   = ');     readln (KBdp1);
      write ('      KBp2+   = ');     readln (KBup2);
      write ('      KBp3+   = ');     readln (KBdp3);
      write ('      KBp1-   = ');     readln (KBup1);
      write ('      KBp2-   = ');     readln (KBdp2);
      write ('      KBp3-   = ');     readln (KBup3);
      write ('       YAto   = ');     readln (YAto);
      write ('       YBto   = ');     readln (YBto);
            YAb1 := YAto;          YBb1 := YBto;
            YAmo := YAto;          YBmo := YBto;
      Initial-data;
      Estimate;
      Show-Estimate;
      Exp-data;
      Trans-Kp;
    ch := 1;
```

```
    While ch > 0 Do
    begin
        Calculation;
    write (' contitue (1/0) := ');
    readln(a);
    ch := a;
    end;
  readln
End.
{-----------------------------------------------------------------------------------------------------------------------}
```

ก.4 โปรแกรมคำนวณหาสมการสมดุลของการแยกสารผสมที่เหมาะสม

```
Program Curve-Fitting;
    Var
    j,n,w,ch : integer;
    A,B,C,D,L,Ek,Erms,Xmean,Ymean,SumX,SumXY : real;
    x,y : array[0..100,1..2] of real;
{-----------------------------------------------------------------------------------------------------------------------}
Procedure Exp-Data;
  begin
    FOR j := 1 TO n DO
      begin
            write('          x[',j,'] = '); readln(x[j,1]);
            write('          y[',j,'] = '); readln(y[j,1]);
      writeln;
      end;
  end;
{-----------------------------------------------------------------------------------------------------------------------}
```

```
Procedure Trans-Value1;
  begin
    FOR j := 1 TO n DO
    begin
      CASE Ch of
        1:    x[j,2]   := x[j,1];          2:    x[j,2]   := 1/x[j,1];
        3:    x[j,2]   := x[j,1]*y[j,1];   4:    x[j,2]   := x[j,1];
        5:    x[j,2]   := 1/x[j,1];        6:    x[j,2]   := ln(x[j,1]);
        7:    x[j,2]   := x[j,1];          8:    x[j,2]   := ln(x[j,1]);
        9:    x[j,2]   := x[j,1];          10:   x[j,2]   := x[j,1];
       ELSE
              x[j,2]   := x[j,1];
      End;
    End;
  End;
{-------------------------------------------------------------------------------------------------------------------------------}
Procedure Trans-Value2;
  begin
    FOR j := 1 TO n DO
    begin
      CASE Ch of
        1:    y[j,2]   := y[j,1];          2:    y[j,2]   := y[j,1];
        3:    y[j,2]   := y[j,1];          4:    y[j,2]   := 1/y[j,1];
        5:    y[j,2]   := 1/y[j,1];        6:    y[j,2]   := y[j,1];
        7:    y[j,2]   := ln(y[j,1]);      8:    y[j,2]   := ln(y[j,1]);
        9:    y[j,2]   := 1/sqrt(y[j,1]);  10:   y[j,2]   := ln(y[j,1]/x[j,1]);
       ELSE
       Begin
        write('    Enter Value of Constant : L = ':46);
```

```
            readln(L);
                    y[j,2]    := ln((L/y[j,1])-1);
          End;
        End;
      End;
    End;
{-------------------------------------------------------------------------------------------------------------------------------------------------------}
Procedure Least-Sqr;
  begin
    begin
      Xmean := 0;
      FOR j := 1 TO n DO
              Xmean := Xmean + x[j,2];
    end;
              Xmean := Xmean / n;
      writeln('       Xmean = ',Xmean:8:5);
    begin
      Ymean := 0;
      FOR j := 1 TO n DO
              Ymean := Ymean + y[j,2];
    end;
              Ymean := Ymean / n;
      writeln('       Ymean = ',Ymean:8:5);
      readln;
      SumX := 0;
      FOR j := 1 TO n DO
        begin
              SumX := SumX + Sqr(x[j,2]-Xmean);
        end;
```

```
    writeln('          SumX = ',SumX:8:5);
    SumXY := 0;
    FOR j := 1 TO n DO
      begin
          SumXY := SumXY + ((x[j,2]-Xmean)*(y[j,2]-Ymean));
      end;
    writeln('          SumXY = ',SumXY:8:5);
          A := SumXY / SumX;
          B := Ymean -(A*Xmean);
    writeln('          A = ',A:8:5);
    writeln('          B = ',B:8:5);
  end;
{-------------------------------------------------------------------------------------------------------------------------------------------------------}
Procedure Extra-Coeff1;
  begin
    CASE Ch of
      3:      C := (-1/A);
      7:      C := exp(B);
      8:      C := exp(B);
      10:     C := exp(B);
      11:     C := exp(B);
    End;
    writeln('          C = ',C:8:5);
  End;
{-------------------------------------------------------------------------------------------------------------------------------------------------------}
Procedure Extra-Coeff2;
  begin
    CASE Ch of
      3:      D := (-B/A);
```

```
        10:     D := -A;
    End;
    writeln('           D = ',D:8:5);
  End;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Equation;

```
  begin
    CASE Ch of
        1:      writeln('  y = ',A:5:4,'x+',B:5:4);
        2:      writeln('  y = (',A:5:4,'/x)+',B:5:4);
        3:      writeln('  y = ',D:5:4,'/(x+',C:5:4,')');
        4:      writeln('  y = 1/(',A:5:4,'+',B:5:4,'x)');
        5:      writeln('  y = x/(',A:5:4,'+',B:5:4,'x)');
        6:      writeln('  y = ',A:5:4,' ln(x)+',B:5:4);
        7:      writeln('  y = ',C:5:4,' exp(',A:5:4,'x)');
        8:      writeln('  y = ',C:5:4,'x^',A:5:4);
        9:      writeln('  y = (',A:5:4,'x+',B:5:4,')^-2');
       10:      writeln('  y = ',C:5:4,'x exp(-',D:5:4,'x)');
      ELSE
               writeln('  y = ',L:5:4,'/(1+',C:5:4,' exp(',A:5:4,'x)');
    End;
  End;
{-------------------------------------------------------------------------------------------------------------------------------}
```

Procedure Error;

```
  begin
    Textcolor(Yellow);
    CASE Ch of
        1: begin
               Ek := 0;
```

```
For j := 1 To n Do
 begin
      Ek := Ek + Sqr(Abs(((A*x[j,1])+B)-y[j,1]));
 end;
      Erms := Sqrt(Ek/n);
 writeln(Erms:5:4);
End;
2: begin
      Ek := 0;
For j := 1 To n Do
 begin
      Ek := Ek + Sqr(Abs(((A/x[j,1])+B)-y[j,1]));
 end;
      Erms := Sqrt(Ek/n);
 writeln(Erms:5:4);
End;
3: begin
      Ek := 0;
For j := 1 To n Do
 begin
      Ek := Ek + Sqr(Abs((D/(x[j,1]+C))-y[j,1]));
 end;
      Erms := Sqrt(Ek/n);
 writeln(Erms:5:4);
End;
4: begin
      Ek := 0;
For j := 1 To n Do
 begin
```

```
        Ek := Ek + Sqr(Abs((1/((A*x[j,1])+B))-y[j,1]));
    end;
        Erms := Sqrt(Ek/n);
    writeln(Erms:5:4);
  End;
5: begin
        Ek := 0;
  For j := 1 To n Do
    begin
        Ek := Ek + Sqr(Abs((x[j,1]/(A+(B*x[j,1])))-y[j,1]));
    end;
    Erms := Sqrt(Ek/n);
    writeln(Erms:5:4);
  End;
6: begin
        Ek := 0;
  For j := 1 To n Do
    begin
        Ek := Ek + Sqr(Abs(((A*ln(x[j,1]))+B)-y[j,1]));
    end;
        Erms := Sqrt(Ek/n);
    writeln(Erms:5:4);
  End;
7: begin
        Ek := 0;
  For j := 1 To n Do
    begin
        Ek := Ek + Sqr(Abs((C*exp(A*x[j,1]))-y[j,1]));
    end;
```

```
        Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
8: begin
        Ek := 0;
    For j := 1 To n Do
      begin
        Ek := Ek + Sqr(Abs((exp((A*ln(x[j,1]))+ln(C)))-y[j,1]));
      end;
        Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
9: begin
        Ek := 0;
    For j := 1 To n Do
      begin
        Ek := Ek + Sqr(Abs((1/Sqr((A*x[j,1])+B))-y[j,1]));
      end;
        Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
10: begin
        Ek := 0;
    For j := 1 To n Do
      begin
        Ek := Ek + Sqr(Abs((C*x[j,1]*exp((-D)*x[j,1]))-y[j,1]));
      end;
        Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
```

```
            End;
          11: begin
                  Ek := 0;
              For j := 1 To n Do
               begin
                  Ek := Ek + Sqr(Abs((L/(1+(C*exp(A*x[j,1])))))-y[j,1]));
               end;
                  Erms := Sqrt(Ek/n);
               writeln(Erms:5:4);
              End;
       End;
    End;
{-------------------------------------------------------------------------------------------------------------------------------------------------}
BEGIN {Main}
  Clrscr;  Textcolor(lightgreen);
       writeln('          *** Curve Fitting Program ***':45);
       writeln;
       write('            Number of Data Point  =  ':45);
       readln(n);    writeln;
       Exp-Data;
       REPEAT
       REPEAT
       writeln('  ***Select Function Form You Wish To Fit***':52);
       writeln('  1.  y = Ax+B                    2.  y = (A/x)+B':50);
       writeln('  3.  y = D/(x+C)                 4.  y = 1/(Ax+B)':51);
       writeln('  5.  y = x/(A+Bx)                6.  y = A ln(x)+B':52);
       writeln('  7.  y = C exp(Ax)               8.  y = Cx^A':47);
       writeln('  9.  y = (Ax+B)^-2               10. y = Cx exp(-Dx)':54);
       writeln('  11. y = L/(1+C exp(Ax))':34);
```

```
        writeln;
        write('   Select [ 1,2,...,11 ] ... choice = ':45);
        readln(ch);
        UNTIL    (ch=1) or (ch=2) or (ch=3) or (ch=4) or (ch=5) or (ch=6)
                 or (ch=7) or (ch=8) or (ch=9) or (ch=10) or (ch=11);
        Trans-Value1;
        Trans-Value2;
        Least-Sqr;
        Extra-Coeff1;
        Extra-Coeff2;
        writeln;
        Textcolor(lightgreen);
        write('       *** The Least-Square Curve for the data is : ':50);
        Equation;
        write('       *** Root Mean Square Error,Erms = ':58);
        Error;
        writeln;
        write('       *** Continue 1/0  : ':52);
        readln(w);
        UNTIL    (w=0);
        readln
  END. {Main}
   End;
{-------------------------------------------------------------------------------------------------------------------------}
```

ก.5 โปรแกรมคำนวณค่าความผิดพลาดของผลการคำนวณ เมื่อเปรียบเทียบกับผลการทดลอง

```
Program Curve_Error;
    Uses Crt, Printer;
    Var
      n,j,w,ch : integer;
      A,B,C,D,L,Ek,Erms : real;
      x,y : array[0..100,1..2] of real;
    {-------------------------------------------------------------------------------------------------------------------}
    Procedure Exp_Data;
      begin
        Textcolor(Yellow);
         FOR j := 1 TO n DO
           begin
           write('              x[',j,'] = '); readln(x[j,1]);
           write('              y[',j,']. = '); readln(y[j,1]);
           writeln;
           end;
      end;
    {-------------------------------------------------------------------------------------------------------------------}
    Procedure Error;
      begin
        Textcolor(Yellow);
        CASE Ch of
            1: begin
               Ek := 0;
               For j := 1 To n Do
                 begin
                 Ek := Ek + Sqr(Abs(((A*x[j,1])+B)-y[j,1]));
                 Ek := Ek;
```

```
      end;
      Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
  2: begin
    Ek := 0;
    For j := 1 To n Do
      begin
      Ek := Ek + Sqr(Abs(((A/x[j,1])+B)-y[j,1]));
      Ek := Ek;
      end;
      Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
  3: begin
    Ek := 0;
    For j := 1 To n Do
      begin
      Ek := Ek + Sqr(Abs((D/(x[j,1]+C))-y[j,1]));
      Ek := Ek;
      end;
      Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
  4: begin
    Ek := 0;
    For j := 1 To n Do
      begin
      Ek := Ek + Sqr(Abs((1/((A*x[j,1])+B))-y[j,1]));
```

```
      Ek := Ek;
      end;
      Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
5: begin
    Ek := 0;
    For j := 1 To n Do
      begin
      Ek := Ek + Sqr(Abs((x[j,1]/(A+(B*x[j,1])))-y[j,1]));
      Ek := Ek;
      end;
      Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
6: begin
    Ek := 0;
    For j := 1 To n Do
      begin
      Ek := Ek + Sqr(Abs(((A*ln(x[j,1]))+B)-y[j,1]));
      Ek := Ek;
      end;
      Erms := Sqrt(Ek/n);
      writeln(Erms:5:4);
    End;
7: begin
    Ek := 0;
    For j := 1 To n Do
      begin
```

```
    Ek := Ek + Sqr(Abs((C*exp(A*x[j,1]))-y[j,1]));
    Ek := Ek;
    end;
    Erms := Sqrt(Ek/n);
    writeln(Erms:5:4);
  End;
8: begin
  Ek := 0;
  For j := 1 To n Do
    begin
    Ek := Ek + Sqr(Abs((exp((A*ln(x[j,1]))+ln(C)))-y[j,1]));
    Ek := Ek;
    end;
    Erms := Sqrt(Ek/n);
    writeln(Erms:5:4);
  End;
9: begin
  Ek := 0;
  For j := 1 To n Do
    begin
    Ek := Ek + Sqr(Abs((1/Sqr((A*x[j,1])+B))-y[j,1]));
    Ek := Ek;
    end;
    Erms := Sqrt(Ek/n);
    writeln(Erms:5:4);
  End;
10: begin
  Ek := 0;
  For j := 1 To n Do
```

```
            begin
            Ek := Ek + Sqr(Abs((C*x[j,1]*exp((-D)*x[j,1]))-y[j,1]));
            Ek := Ek;
            end;
            Erms := Sqrt(Ek/n);
            writeln(Erms:5:4);
          End;
       11: begin
          Ek := 0;
          For j := 1 To n Do
            begin
            Ek := Ek + Sqr(Abs((L/(1+(C*exp(A*x[j,1])))))-y[j,1]));
            Ek := Ek;
            end;
            Erms := Sqrt(Ek/n);
            writeln(Erms:5:4);
          End;
    End;
  End;
{-------------------------------------------------------------------------------------------------------------------------------}
  BEGIN {Main}
   Clrscr;   Textcolor(lightgreen);
   writeln('          *** Curve Error Program ***':45);
   writeln;
    write('          Number of Data Point  = ':45);
    readln(n);     writeln;
  Exp_Data;
  REPEAT
  REPEAT
```

```
  Textcolor(lightgreen);
  writeln('   ***Select Function Form You Wish To Fit***':52);
  writeln('   1.  y = Ax+B                    2.  y = (A/x)+B':50);
  writeln('   3.  y = D/(x+C)                 4.  y = 1/(Ax+B)':51);
  writeln('   5.  y = x/(A+Bx)                6.  y = A ln(x)+B':52);
  writeln('   7.  y = C exp(Ax)               8.  y = Cx^A':47);
  writeln('   9.  y = (Ax+B)^-2               10. y = Cx exp(-Dx)':54);
  writeln('   11. y = L/(1+C exp(Ax))':34);
  writeln;
  write('   Select [ 1,2,...,11 ] ... choice = ':45);
  readln(ch);
 UNTIL   (ch=1) or (ch=2) or (ch=3) or (ch=4) or (ch=5) or (ch=6)
         or (ch=7) or (ch=8) or (ch=9) or (ch=10) or (ch=11);
writeln;
  Textcolor(lightgreen);
  write('        *** Constant value : A = ':45);      readln(a);
  write('        *** Constant value : B = ':45);      readln(b);
  write('        *** Constant value : C = ':45);      readln(c);
  write('        *** Constant value : D = ':45);      readln(d);
  write('        *** Root Mean Square Error, Erms = ':45);
Error;
writeln;
  write('        *** Continue 1/0  : ':52);
  readln(w);
 UNTIL   (w=0);
 readln
 END. {Main}
{-------------------------------------------------------------------------------------------------------------------------}
```

ภาคผนวก ข.

ข.1 ตัวอย่างการคำนวณโดยใช้โปรแกรมคอมพิวเตอร์

การทำงานของโปรแกรมคอมพิวเตอร์ที่ใช้ในการคำนวณค่าความเข้มข้นในแต่ละรอบของการทดลอง เป็นการทำงานแบบทำซ้ำจนได้ผลการคำนวณที่พอใจ ข้อมูลพื้นฐานที่ต้องการ คือ ปริมาตรในแต่ละถังพัก และฟังก์ชันสมดุลของการแยกสารผสมแบบเชิงเส้น ขั้นตอนการทำงานของโปรแกรมแสดงไว้ในตัวอย่างการคำนวณของรูปแบบที่ 1.(Calculation)

I. คำนวณค่าความเข้มข้นในสภาวะของเหลว เมื่อฟังก์ชันสมดุลของการแยกสารผสมเป็นแบบเชิงเส้น

n	Kp1+	Kp2-	Kp1-	Kp2+	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)
0	3.00	2.00	1.40	1.00	1.0000	-	1.0000	1.0000
1	3.00	2.00	1.40	1.00	0.5000	0.8000	1.0833	1.1500
2	3.00	2.00	1.40	1.00	0.4125	0.8889	1.1921	1.0631
3	3.00	2.00	1.40	1.00	0.3689	0.9107	1.2557	1.0087
4	3.00	2.00	1.40	1.00	0.3444	0.9222	1.2917	0.9777
5	3.00	2.00	1.40	1.00	0.3305	0.9287	1.3121	0.9602
6	3.00	2.00	1.40	1.00	0.3227	0.9324	1.3237	0.9502
7	3.00	2.00	1.40	1.00	0.3182	0.9345	1.3303	0.9446

II. แทนค่า Yt(n) และ Yb(n+1) ที่ได้จากการคำนวณด้วย ค่าที่ได้จากผลการทดลองจริง จากนั้นหาค่าสมดุลของการแยกสารผสมใหม่

n	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)	Kp1+	Kp2-	Kp1-	Kp2+
0	1.0000	-	1.0000	1.0000	3.00	2.00	1.40	1.00
1	0.5715	0.8000	1.1152	1.1500	2.50	2.00	1.33	0.94
2	0.4596	0.8889	1.2029	1.0631	2.59	1.96	1.38	0.96
3	0.4046	0.9107	1.2574	1.0087	2.65	1.99	1.40	0.96
4	0.3696	0.9222	1.2975	0.9777	2.73	2.00	1.39	0.97

5	0 3446	0 9287	1.3295	0 9602	2 84	1.99	1 37	0.99
6	0.3254	0 9324	1.3563	0 9502	2.97	1.98	1.34	1.00
7	0.3100	0.9345	1.3793	0.9446	3.11	1.97	1.31	1 01

III. หาค่าความเข้มข้นในสภาวะของเหลวเมื่อสมดุลของการแยกสารผสมเป็นค่าเฉลี่ยระหว่าง I กับ II

n	Kp1+	Kp2-	Kp1-	Kp2+	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)
0	3.00	2.00	1.40	1.00	1.0000	-	1.0000	1.0000
1	2.75	2.00	1.37	0.97	0.5334	0.8000	1.0991	1.1512
2	2.79	1.98	1.39	0 98	0.4345	0.8894	1.1978	1 0636
3	2.82	1.99	1.40	0.98	0.3856	0.9110	1.2569	1.0089
4	2.86	2.00	1.39	0.99	0.3563	0.9224	1.2949	0.9777
5	2.92	2.00	1.38	0.99	0.3372	0.9289	1.3210	0.9600
6	2.98	1.99	1.37	1.00	0.3239	0.9325	1.3399	0.9501
7	3.05	1.98	1.36	1.00	0.314	0.9346	1.3544	0.9445

IV. ทำข้อ II และ III ซ้ำจนกระทั่งค่าความเข้มข้นในสภาวะของเหลวในแต่ละรอบของการทดลองคง ที่ ผลการคำนวณสุดท้ายที่ได้ คือ

n	Kp1+	Kp2-	Kp1-	Kp2+	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)
0.	3.00	2.00	1.40	1.00	1.0000	-	1.0000	1.0000
1	2.50	2.00	1.33	0.94	0.5715	0.8000	1.1152	1.1517
2	2.59	1.96	1.38	0.95	0.4596	0.8896	1.2029	1.0638
3	2.64	1.99	1.40	0.96	0.4046	0.9111	1.2574	1.0089
4	2.72	2.00	1.39	0.97	0.3696	0.9225	1.2975	0.9777
5	2.83	1.99	1.37	0.98	0.3446	0.9289	1.3295	0.9600
6	2.96	1.98	1.34	1.00	0.3254	0.9325	1.3562	0.9500
7	3.10	1.97	1.31	1.01	0.3100	0.9346	1.3793	0.9445

ข 2 ผลการคำนวณค่าความเข้มข้นในแต่ละรอบของการทดลองในสภาวะของเหลว สำหรับกรณีที่สมดุลของการแยกสารผสมเป็นฟังก์ชันแบบไม่เชิงเส้น

รูปแบบที่ 1 (calculation)

n	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)
0	1.0000	-	1.0000	1.0000
1	0.5720	0.8001	1.0909	1.1413
2	0.4559	0.8882	1.2067	1.0570
3	0.3933	0.9069	1.2758	1.0023
4	0.3562	0.9162	1.3156	0.9713
5	0.3342	0.9216	1.3386	0.9537
6	0.3212	0.9248	1.3519	0.9436
7	0.3134	0.9268	1.3598	0.9378

รูปแบบที่ 1 (experinmention)

n	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)
0	1.0000	-	1.0000	1.0000
1	0.7160	0.7537	1.0827	1.0325
2	0.6065	0.8152	1.1538	0.9395
3	0.5538	0.8319	1.1883	0.8921
4	0.5273	0.8399	1.2050	0.8698
5	0.5139	0.8440	1.2131	0.8591
6	0.5071	0.8462	1.2172	0.8538
7	0.5037	0.8473	1.2193	0.8511

รูปแบบที่ 2 โปรตีนฮีโมโกลบิน (calculation)

n	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)	Ys(n+1)
0	1.0000	-	1.0000	1.0000	1.0000
1	0.5592	0.8070	1.0908	1.1452	1.0712
2	0.4490	0.8957	1.2093	1.0630	1.0778
3	0.3877	0.9133	1.2810	1.0078	1.0551
4	0.3613	0.8214	1.3224	0.9856	1.0289
5	0.3470	0.9296	1.3495	0.9765	1.0062
6	0.3378	0.9368	1.3687	0.9718	0.9882
7	0.3312	0.9425	1.3829	0.9690	0.9738
8	0.3263	0.9469	1.3935	0.9669	0.9633

รูปแบบที่ 2 โปรตีนอัลบูมิน (calculation)

n	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)	Ys(n+1)
0	1.0000	-	1.0000	1.0000	1.0000
1	0.6886	1.0078	0.9975	0.8316	1.2310
2	0.6354	0.9426	0.9617	0.8114	1.3138
3	0.6825	0.9066	0.9308	0.8008	1.3396
4	0.7602	0.8783	0.9054	0.7928	1.3430
5	0.8400	0.8556	0.8847	0.7864	1.3384
6	0.9118	0.8372	0.8678	0.7814	1.3314
7	0.9728	0.8224	0.8542	0.7773	1.3244
8	1.0234	0.8105	0.8431	0.7740	1.3182

รูปแบบที่ 2 โปรตีนฮีโมโกลบิน (experimention)

n	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)	Ys(n+1)
0	1.0000	-	1.0000	1.0000	1.0000
1	0.7186	0.7876	1.0694	1.1190	0.9556
2	0.5430	0.8638	1.1508	1.0590	0.9323
3	0.4309	0.8966	1.2015	1.0091	0.9225
4	0.3591	0.9143	1.2308	0.9760	0.9201
5	0.3135	0.9235	1.2469	0.9538	0.9213
6	0.2851	0.9276	1.2549	0.9383	0.9243
7	0.2679	0.9284	1.2580	0.9269	0.9279
8	0.2579	0.9271	1.2579	0.9182	0.9316

รูปแบบที่ 2 โปรตีนอัลบูมิน (experimention)

n	Yt(n)	Ym(2n-1)	Yb(n+1)	Ym(2n)	Ys(n+1)
0	1.0000	-	1.0000	1.0000	1.0000
1	0.8193	0.7740	1.0223	0.9036	1.0363
2	0.7744	0.8050	1.0307	0.7062	1.1375
3	0.7738	0.7741	1.0250	0.6312	1.2379
4	0.7854	0.7331	1.0158	0.6072	1.3175
5	0.7980	0.6969	1.0070	0.6015	1.3749
6	0.8082	0.6690	0.9997	0.6015	1.4145
7	0.8157	0.6491	0.9943	0.6025	1.4415
8	0.8209	0.9353	0.9904	0.6040	1.4595

รูปแบบที่ 2 โปรตีนฮีโมโกลบิน (calculation)

n	Yt(n)	Ym(4n-3)	Yb(n+1)*	Ym(4n-2)	Yt(n)*	Ym(4n-1)	Yb(n+1)	Ym(4n)
0	1.0000	-	1.0000	1.0000	1.0000	-	1.0000	1.0000
1	0.8426	0.9311	0.5737	1.4581	1.8569	1.2345	0.8302	0.8180
2	0.5475	0.9779	0.5913	1.1323	2.2469	1.2169	0.8809	0.8524
3	0.4813	0.9604	0.5729	1.0588	2.4622	1.1679	0.9267	0.8669
4	0.4763	0.9450	0.5547	1.0456	2.5889	1.1293	0.9652	0.8753
5	0.4866	0.9329	0.5401	1.0481	2.6657	1.1011	0.9941	0.8801
6	0.4975	0.9241	0.5296	1.0534	2.7131	1.0816	1.0144	0.8829
7	0.5057	0.9182	0.5224	1.0579	2.7424	1.0686	1.0280	0.8846
8	0.5113	0.9143	0.5178	1.0610	2.7607	1.0603	1.0368	0.8856

รูปแบบที่ 2 โปรตีนอัลบูมิน (calculation)

n	Yt(n)	Ym(4n-3)	Yb(n+1)*	Ym(4n-2)	Yt(n)*	Ym(4n-1)	Yb(n+1)	Ym(4n)
0	1.0000	-	1.0000	1.0000	1.0000	-	1.0000	1.0000
1	1.0927	1.0523	0.7601	1.0327	1.1173	1.2049	1.0066	0.9859
2	1.2236	1.0500	0.5511	1.0244	1.2053	1.1601	1.0104	1.0045
3	1.3240	1.0605	0.3911	1.0266	1.2740	1.1248	1.0129	1.0195
4	1.3999	1.0685	0.2767	1.0285	1.3576	1.0893	1.0149	1.0397
5	1.4389	1.0785	0.2050	1.0334	1.3706	1.0627	1.0165	1.0541
6	1.4506	1.0858	0.1644	1.0379	1.4051	1.0442	1.0176	1.0629
7	1.4449	1.0903	0.1431	1.0413	1.4327	1.0333	1.0184	1.0656
8	1.4331	1.0920	0.1318	1.0430	1.4546	1.0267	1.0189	1.0656

รูปแบบที่ 3 โปรตีนฮีโมโกลบิน (experimention)

n	Yt(n)	Ym(4n-3)	Yb(n+1)*	Ym(4n-2)	Yt(n)*	Ym(4n-1)	Yb(n+1)	Ym(4n)
0	1.0000	-	1.0000	1.0000	1.0000	-	1.0000	1.0000
1	0.4196	1.1220	0.6529	1.3541	1.3483	1.4303	0.9848	0.6854
2	0.3878	1.1100	0.6244	1.3312	1.7046	1.3595	1.0309	0.6651
3	0.3235	1.0688	0.5943	1.2570	2.0880	1.2785	1.0831	0.6581
4	0.2530	1.0273	0.5643	1.1779	1.5026	1.1978	1.1417	0.6443
5	0.1662	0.9842	0.5336	1.0890	2.9553	1.1147	1.2112	0.6697
6	0.1278	0.9426	0.5042	1.0045	3.4475	1.0362	1.2857	0.6904
7	0.1289	0.9030	0.4767	0.9246	3.9805	0.9629	1.3664	0.7087
8	0.1647	0.8655	0.4509	0.8486	4.5545	0.8945	1.4542	0.7149

รูปแบบที่ 3 โปรตีนอัลบูมิน (experimention)

n	Yt(n)	Ym(4n-3)	Yb(n+1)*	Ym(4n-2)	Yt(n)*	Ym(4n-1)	Yb(n+1)	Ym(4n)
0	1.0000	-	1.0000	1.0000	1.0000	-	1.0000	1.0000
1	1.0448	1.0345	0.3928	1.0105	1.1111	1.2293	1.1355	1.0188
2	1.1129	1.0762	0.1418	1.0271	1.2010	1.1515	1.1472	1.0361
3	1.1744	1.0829	0.0634	1.0500	1.3234	1.0998	1.1208	1.0539
4	1.2149	1.0794	0.0422	1.0675	1.4051	1.0779	1.0960	1.0644
5	1.2366	1.0749	0.0351	1.0778	1.4814	1.0693	1.0801	1.0727
6	1.2513	1.0723	0.0322	1.0848	1.5407	1.0663	1.0714	1.0787
7	1.2685	1.0732	0.0323	1.0918	1.5966	1.0694	1.0695	1.0834
8	1.2802	1.0716	0.0314	1.0972	1.6435	1.0696	1.0689	1.0878

ประวัติผู้เขียน

ชื่อ	นายอิศรา โล่สุวรรณ
เกิด	29 ตุลาคม 2514
การศึกษา	จบการศึกษาระดับปริญญาตรี คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมเคมี มหาวิทยาลัยรังสิต ปีการศึกษา 2535