



REFERENCES

- Alexander, R.McK. Proc. Austral. Inst. Min. Met. (N.S.) p 152, 202, 1949.
- Baxter, W.A. Air Pollution Control. vol 4. 3rd edition (A.C. Stern, ed.). Chapter 3. p 124
New York: Academic Press, 1977.
- Caplan and Knowlton. Air Pollution Control. III A.C. Stern(ed). 2nd ed. p 359 New York:
Academic Press, 1968.
- Coulson and Richradson, J.F. Chemical Engineering vol 2 Oxford: Pergamon Press, 1978.
- Drayer. PetrolChem. Eng. (May 1970): 39.
- Daniels, T.C. Engineer. 203, 358, (1975).
- Davies, C.N. Proc. Inst. Mech. Eng. 1. 185 (1952).
- Edgar and Himmelblau, D.M. Optimization of chemical process. Mc Graw-Hill. p 193,
1989.
- First, M.W. paper presented at ASME Con., 1949.
- Fuchs, N.A. The Mechanical of Aerosols. New York: Pergamon Press, 1964. p 33.
- Gerrard and Liddle, C.J. Powder Technol. 13 (1976): 251.
- _____. Eng. Proc. Econ. 1 (1976): 145.
- Hawksley, P.G.W., S. Badzioch, and Blackett, J.H. "Measurement of Solids in Flue Gases,"
British Coal Utilization Research Assn. England: Leatherhead, 1961.
- Casal and Martinez-Benet, J.M. Chem. Eng. 90 (1983): 99.
- Jotaki, T. Trans. Jap. S.M.E. 23, 640, 1957.
- Jerry Banks and John S. Carson, II. Discrete-event system simulation. Prentice-Hall. p.12,
1984.
- Kalen, R. and Zenz, F.A. AIChE Symposium Ser. 71(137), 388, 1974.
- Kermode. Eng. Proc. Econ. 7 (1982): 63.
- Koch and W. Licht. Chem. Eng. 84 (1977): 80.
- Lapple, CE. Chem. Eng. 58 (1951): 133.
- _____. "Prediction Efficiency of Cyclones," Air Pollution Eng. Manual.
USPHS 99-AP-40, p. 95 (1967).

- Leith, D. and W. Licht. The collection efficiency of cyclone type particle collectors. AIChE Symposium series. vol 68. No. 126, 1972.
- _____. Handbook of Pollution Abatement Engineering NJ: Humana Press. Clifton, 1978.
- _____. Cyclone performance and design, Atmos Environ. vol 7, 1973 p 527-549.
- Licht, W., loc. cit., p 250.
- _____. Air Pollution Control Engineering New York: Dekker, 1980. p 140, 177, 243.
- Mehta, Dilip. private communication to the authors, 1970.
- Mori, Y., S. Akira, and S. Tauaka. J. Chem. Eng. (Japan) 1. 92 (1968).
- Neveril, R.B., Price, J.V. and Engdahl, K.L. J. Air Poll. Contr. Assoc. 28. 963 (1978).
- Peterson, C.M. and K.T. Whitby. ASHRAE J. 42 (1965).
- Prockat, F. Glasers Ann. 107, 43 (1930).
- Shepherd, G.B., and Lapple, G.E. Ind. Eng. Chem. 31, 972 (1939).
- Sproull, W.T. Air Pollution Control Assn. J. 16,439 (1966).
- Sproull, W.T. Air Pollution and its Control. N.Y.: Exposition Press. Jericho. (1970). p 62.
- Stairmand, C.J. Trans. Ins. Chem. Engs. 29, 356 (1951).
- Strauss, W. Industrial Gas cleaning. Chap 6. New York: Pergamon Press, (1966).
- _____. Industrial Gas Cleaning. 3rd edition. Oxford: Pergamon Press, 1975 p 263.
- Swift and Peter Steam Heating Eng. 38, 453 (1969).
- Ter Liden, A.J. Proc. Inst Mech. Eng. 160, 233 (1949).
- TerLinden, A.J. Tonindustrie-Zeitung 22(iii) 49 (1953).
- Umney, L.E.R. Theory and Design of an Improved Centrifugal Air Cleaner. Nat Gas Turbine Estab. Rept. No. R33 (1948).
- Van Ebbenhorst Tengbergen, H.J. Staub 25, 44 (1965).
- Zenz, in J.J. McKetta (Ed). Encyclopedia of Chemical Processing and Design. vol. 7 New York: M. Dekker Inc., 1978. p 83.
- _____. Ind. and Eng. Chem. Fund., 3 (1964) 65.

APPENDIX A.

LIST OF COMPUTER PROGRAM

```
program cyclone;
uses dos,crt,graph,printer;
const max=30;
type screenarray = array[1..25,1..80,1..2] of char;
var coloscreen : screenarray absolute $B800:$0000;
    scrn : screenarray;
var Graphdriver,Graphmode : integer;
    Vmax,K,Ka,Kb,Nh : real;
    CD,D,Q,Pp,P,DL,T,U,Dp,N,Dcpr,Dcp,Vin : real;
    Obj,Acy,Wcy,Mcy,Fcy,Rcy,Tfc,Th,Toc,Tc : real;
    Tc_No : real;
    Dcp_No1,Tfc_No1,Toc_No1,Tc_No1,V_No1,PDr_No1,Vs_No1,Obj_No1 : real;
    Sum_Mi,Sum_nigi,nn,nn_No1 : real;
    Vs,VI,Ce,Yrs,Hpy : real;
    Da,Thk,Roc,Mac,Fac,W : real;
    A,PDr,C,X,Y : real;
    K1,Ka1,Kb1,Nh1,Cyh,B,H,De,S : real;
    tpe,AC,pwr_Eff,pwr,Nmin,Nmax,count,No,ypos : integer;
    ystep,ystep_Eff : longint;
    Cy,datfile : string[20];
ans:char;
prt,error,user,OverAll : boolean;
Max_Obj : real;
c_type : byte;
```

```

Nc : array[1..max] of integer;
Mi,Avg_Dp,gi : array[1..7] of real;
ni : array[1..7] of real;
mark : array[1..6] of boolean;
len: array [1..6] of byte;
Tfc_No,Toc_No,Vs135_No,PDr_No,D_No,nn_No,Tco,Dcp_No,Obj_No : array[1..max] of real;
Obj_graph,G_Eff : array[1..6,0..16] of longint;
data : text;

procedure writexy(x,y,co,bc : byte; st : string); forward;
procedure update_cyclone_data(i : byte); forward;
procedure update_oper_data; forward;
procedure update_cost_data; forward;
procedure Input_Cyclone_Dat; forward;
procedure trns_ch_val(var value : real; ch : char); forward;
procedure multi_Cyclone(typ : byte); forward;
procedure Shw_Dat(xpos,ypos,Cy_typ : integer); forward;
procedure cal; forward;
procedure Cal_Over; forward;
procedure Cal_PlotGraph(cy : integer); forward;
procedure Cal_PlotGraph_N(cy : integer); forward;
procedure Adj_Obj_ystep; forward;
procedure PlotGraph(typ,cy : integer); forward;
procedure print; forward;
procedure Drw_Cyclone; forward;
procedure Cyclone_txt_Dat; forward;
procedure gr_box(x1,y1,x2,y2,co,width : integer); forward;
procedure scale(typ,pwr: integer); forward;
procedure initgr; forward;
procedure chk_Vs_Error(ven : real); forward;
{-----}

```

```

procedure curseron;
var curmode : integer absolute $0040:$0060;
    vp : integer absolute $0040:$0063;
begin
    port[vp]:=10;
    port[vp+1]:=hi(curmode) and $44;
    port[vp]:=11;
    port[vp+1]:=lo(curmode);
end;
{-----}
procedure curseroff;
var curmode : integer absolute $0040:$0060;
    vp : integer absolute $0040:$0063;
begin
    port[vp]:=10;
    port[vp+1]:=hi(curmode) or $20;
    port[vp]:=11;
    port[vp+1]:=lo(curmode);
end;
{-----}
procedure initdata;
var i,ii : integer;
    Avg_step : real;
begin
    K:=0; Ka:=0; Kb:=0; Nh:=0; CD:=0; AC:=0;
    Cyh:=0; B:=0; H:=0; De:=0; S:=0;
    K1:=0; Ka1:=0; Kb1:=0; Nh1:=0;
    Q:=0; Pp:=0; P:=0; DL:=0; T:=0; U:=0; Dp:=0; N:=0; Dcpr:=0;
    Ce:=1.07; Yrs:=0; Hpy:=0;
    Roc:=0; Mac:=0; Fac:=0; W:=0;

```

```

Thk:=0; Vin:=15;
user:=false;
error:=false;
OverAll:=false;
for i:=1 to max do
  begin
    Dcp_No[i]:=0; Obj_No[i]:=0;
    Tfc_No[i]:=0; Toc_No[i]:=0;
    Vs135_No[i]:=0; PDr_No[i]:=0;
    D_No[i]:=0; nn_No[i]:=0;
    Tco[i]:=0; Dcp_No[i]:=0;
    Obj_No[i]:=0; Nc[i]:=0;
    count:=0; no:=0;
  end;
for i:=1 to 6 do
  begin
    mark[i]:=false;
    Obj_graph[i,0]:=0;
    G_Eff[i,0]:=0;
  end;
Avg_Dp[1]:=2.5E-6;
Avg_step:=5E-6;
for i:=2 to 4 do Avg_Dp[i]:=Avg_Dp[i-1]+Avg_step;
Avg_Dp[5]:=25E-6;
Avg_Dp[6]:=40E-6;
Avg_Dp[7]:=75E-6;
for i:=1 to 5 do
  for ii:=0 to 16 do
    Obj_graph[i,ii]:=0;
  end;
end;
{-----}

```

```
procedure box(x1,y1,x2,y2,color,bkcolor : byte);
```

```
var i : byte;
```

```
begin
```

```
  for i:=x1 to x2 do
```

```
    begin
```

```
      writexy(i,y1,color,bkcolor,chr(205)); {ธ}
```

```
      writexy(i,y2,color,bkcolor,chr(205)); {ธ}
```

```
    end;
```

```
  for i:=y1 to y2 do
```

```
    begin
```

```
      writexy(x1,i,color,bkcolor,chr(179)); {ถ}
```

```
      writexy(x2,i,color,bkcolor,chr(179)); {ถ}
```

```
    end;
```

```
writexy(x1,y1,color,bkcolor,chr(213)); {}
```

```
writexy(x1,y2,color,bkcolor,chr(212)); {ภ}
```

```
writexy(x2,y1,color,bkcolor,chr(184)); {พ}
```

```
writexy(x2,y2,color,bkcolor,chr(190)); {๓}
```

```
window(x1+1,y1+1,x2-1,y2-1); clrscr;
```

```
window(1,1,80,25);
```

```
end;
```

```
{-----}
```

```
procedure writexy(x,y,co,bc : byte; st : string);
```

```
begin
```

```
  textcolor(co); textbackground(bc);
```

```
  gotoxy(x,y); write(st);
```

```
  textcolor(15); textbackground(0);
```

```
end;
```

```
{-----}
```

```
procedure save(x1,y1,x2,y2 : byte);
```

```
var i,ii,iii : byte;
```

```

begin
  for i:=y1 to y2 do
    for ii:=x1 to x2 do
      for iii:=1 to 2 do
        scrn[i,ii,iii]:=coloscreen[i,ii,iii];
      end;
    {-----}
  procedure restore(x1,y1,x2,y2 : byte);
  var i,ii,iii : byte;
  begin
    for i:=y1 to y2 do
      for ii:=x1 to x2 do
        for iii:=1 to 2 do
          coloscreen[i,ii,iii]:=scrn[i,ii,iii];
        end;
      {-----}
    procedure help;
    var ch : char;
        x1,y1,x2,y2,pg : byte;
    begin
      x1:=15; y1:=1;
      x2:=65; y2:=24;
      save(x1,y1,x2,y2);
      box(x1,y1,x2,y2,14,4);
      writexy(x2-9,y1,4,2, '{ Help }');
      writexy(x2-19,y2,4,2, '[Pg U/D] [Esc-Exit]');
      window(x1+1,y1+1,x2-1,y2-1);
      textbackground(4);
      pg:=1;
      repeat
        if pg=1 then

```




```

begin
  clrscr;
  writeln(' K = Configuration Parameter');
  writeln(' Ka = Inlet Height ; a/D');
  writeln(' Kb = Inlet Width ; b/D');
  writeln(' Nh = Inlet Velocity Head');
  writeln(' Cyh = Cyclone cylinder height per cyclone dia. ');
  writeln(' H = Cyclone height per cyclone dia. ');
  writeln(' B = Dust outlet dia. per cyclone dia. ');
  writeln(' De = Outlet dia. per cyclone dia. ');
  writeln(' S = Outlet length per cyclone dia. ');
  writeln(' Q = Volume Flowrate ; m^3/s');
  writeln(' PD = Particle Density ; Kg/m^3');
  writeln(' FD = Fluid Density ; Kg/m^3');
  writeln(' DL = Dust Loading ; Kg/m^3');
  writeln(' Temp = Absolute temperature (273+C) ; K');
  writeln(' μ = Viscosity of Gas ; Kg/m.s');
  writeln(' Dcpr = Required Cutsie Particle Dia. ; m');
  writeln(' Ce = Power Rate ; Bht/KW.hr');
  writeln(' Yrs = Amount Oper. Years');
  writeln(' Hpy = Amount Oper. Hours per Year ; hrs/yr');
  writeln(' Thk = Steel Thickness ; mm');
  writeln(' Roc = Steel Rolled Cost ; Bht/m^2');
  write (' Mac = Steel Material Cost ; Bht/Kg of Steel');
end else
begin
  clrscr;
  writeln(' Fac = Steel Fabrication Cost ; Bht/Kg of Steel');
  writeln(' Wt = Steel Weight ; Kg');
  writeln(' AC = Amount Cyclones');
  writeln(' CD = Cyclone Diameter ; m');

```

```

writeln(' mi = Mass of particle at specified diameter range ; g ');
writeln(' vi = Inlet velocity ; m/s ');
write (' gi = Mass fraction of particle at specified diameter range ; ');
end;
ch:=readkey;
if ch=#0 then ch:=readkey;
case ch of
  #73 : dec(pg);
  #81 : inc(pg);
end;
if pg<=1 then pg:=1 else pg:=2;
until ch=#27;
window(1,1,80,25);
restore(x1,y1,x2,y2);
end;
{-----}
procedure win_shw_cal;
var xpos,ypos,step,xstep,i,ii,x,y,count : integer;
begin
  initgr;
  gr_box(1,1,getmaxx,getmaxy-2,14,3);
  xpos:=15; ypos:=22; step:=20;
  count:=0;
  for i:=1 to 6 do
    if mark[i]=true then inc(count);
  if count>3 then ii:=2 else ii:=1;
  for i:=1 to ii do
    begin
      setcolor(5);
      outtextxy(xpos,ypos+2*step,'          (m)');
      outtextxy(xpos,ypos+3*step,'          (N/m2)');
    end;
  end;

```

```

outtextxy(xpos,ypos+4*step,'          (m)');
outtextxy(xpos,ypos+5*step,'          (m)');
outtextxy(xpos,ypos+7*step,'        (Baht)');
outtextxy(xpos,ypos+8*step,'        (Baht)');
outtextxy(xpos,ypos+9*step,'      (Baht)');
setcolor(2);
outtextxy(xpos,ypos,'Cyclone Type');
outtextxy(xpos,ypos+step,'Cyclone Amount');
outtextxy(xpos,ypos+2*step,'Cyclone Dia');
outtextxy(xpos,ypos+3*step,'Press. Drop');
outtextxy(xpos,ypos+4*step,'50% Cut Size Part. Dia. ');
outtextxy(xpos,ypos+5*step,'50% Cut Size Part. Dia. Req. ');
outtextxy(xpos,ypos+6*step,'Obj. Equa. Value');
outtextxy(xpos,ypos+7*step,'TTL Fixed Cost');
outtextxy(xpos,ypos+8*step,'TTL Oper. Cost');
outtextxy(xpos,ypos+9*step,'TTL Cost');
if OverAll=true then outtextxy(xpos,ypos+10*step,'Overall Eff. (%)');
inc(ypos,220);
end;
xpos:=230; ypos:=15; xstep:=134;
ii:=0;
if (count>3) and (OverAll=false) then count:=21 else
  if (count>3) and (OverAll=true) then count:=22
  else if OverAll=true then count:=11 else count:=10;
for y:=1 to count do
  begin
    i:=0;
    for x:=1 to 3 do
      begin
        gr_box(xpos+xstep*i,ypos+ii*step,xpos+xstep*(i+1),ypos+step*(ii+1),1,4);
        inc(i);
      end;
    inc(ii);
  end;

```

```

        end;
        gr_box(8,ypos+ii*step,8+xstep+88,ypos+step*(ii+1),4,4);
        inc(ii);
    end;
    setlinestyle(0,0,3);
    setcolor(14);
    if count>10 then line(9,(getmaxy div 2)-4,getmaxx-9,(getmaxy div 2)-4);
    setcolor(2);
    outtextxy(10,getmaxy-15,'±      ±      ±      ±      ±      ±      ±
±');
    setcolor(9);
    outtextxy(10,getmaxy-15,' G-Grade Eff.  O-Obj.  P-Print  V-View Cyc.  D-Data  Pg Up-
Dn  Esc-Exit');
end;
{-----}
procedure main_menu(var ypos : integer);
var  i,oldi : byte;
     ch : char;
     m : array[1..7] of string[27];
begin
    textbackground(0);
    box(1,1,79,24,14,0);
    writexy(30,1,10,0,'<< Cyclone Design >>');
    writexy(4,24,14,0,[' Alt H-Help ']);

    window(2,2,78,23);    { main window}
    textbackground(1);
    clrscr;
    window(1,1,80,25);
    cursoroff;

```

```

box(3,14,18,21,14,5); {cyclone window}
writexy(4,14,10,5,['Cyclone Dat']);
writexy(9,21,10,5,['Pg U/D']);
window(4,15,17,20);
clrscr;
window(1,1,80,25);

```

```

box(20,14,48,23,14,5); {oper window}
writexy(25,14,10,5,['Oper Condition Dat']);
window(21,15,47,22);
clrscr;
window(1,1,80,25);

```

```

box(50,14,77,23,14,5); {cost window}
writexy(55,14,10,5,['Cost & Thick Dat']);
window(51,15,76,22);
clrscr;
window(1,1,80,25);

```

```

update_cyclone_data(1);
update_oper_data;
update_cost_data;
m[1]:=' Cyclone Data      ';
m[2]:=' Operating Condition Data  ';
m[3]:=' Cost Data          ';
m[4]:=' Thickness Data      ';
m[5]:=' Cal. Overall Eff.     ';
m[6]:=' Calculate           ';
m[7]:=' Quit                ';
box(3,3,31,11,14,4);
writexy(12,3,10,4,['Main Menu']);

```

```

for i:=1 to 7 do writexy(4,3+i,15,0,m[i]);
i:=ypos;
textcolor(0); textbackground(15);
gotoxy(4,3+i); write(m[i]);
textcolor(15); textbackground(0);
repeat
  writexy(5,4,10,0,'C');
  writexy(5,5,10,0,'O');
  writexy(10,6,10,0,'D');
  writexy(5,7,10,0,'T');
  writexy(18,8,10,0,'E');
  writexy(7,9,10,0,'L');
  writexy(5,10,10,0,'Q');
  oldi:=i;
  ch:=readkey;
  if ch=#00 then ch:=readkey;
  case upcase(ch) of
    #72 : dec(i);
    #80 : inc(i);
    #35 : help;
  end;
  if upcase(ch) in['C','O','D','T','E','L','Q'] then
  begin
    case upcase(ch) of
      'C': i:=1;
      'O': i:=2;
      'D': i:=3;
      'T': i:=4;
      'E': i:=5;
      'L': i:=6;
      'Q': i:=7;

```

```

    end;
    ch:=#13;
    end;
if i<1 then i:=7;
if i>7 then i:=1;
textbackground(0); textcolor(15);
gotoxy(4,3+oldi); write(m[oldi]);
textbackground(15); textcolor(0);
gotoxy(4,3+i); write(m[i]);
textbackground(0); textcolor(15);
until ch=#13;
ypos:=i;
end;
{-----}
procedure sub_menu1;
var m : array[1..6] of string[28];
    i,ii,oldi : byte;
    ch : char;
begin
    m[1]:= ' High Eff -Stairmand a      ';
    m[2]:= ' High Eff -Swift b        ';
    m[3]:= ' Shepherd/Lapple c          ';
    m[4]:= ' Gen Pur -Swift b                ';
    m[5]:= ' Gen Pur -Peter n Whitby d      ';
    m[6]:= ' User Type                        ';
    box(38,3,67,10,14,2);
    writexy(37,24,10,0,[' M-Mark/Unmark  Enter-Mark  Esc-Quit  ']);
    for i:=1 to 6 do
        begin
            writexy(39,3+i,15,0,m[i]);
            if mark[i]=true then writexy(69,3+i,14+blink,1,chr(17));
        end;
    end;
end;

```

```

end;
textcolor(0); textbackground(15);
gotoxy(39,4); write(m[1]);
textcolor(15); textbackground(0);
i:=1; ii:=1;
repeat
  oldi:=i;
  multi_cyclone(i);
  update_cyclone_data(i);
  ch:=readkey;
  if upcase(ch)='M' then
    begin
      mark[i]:=not(mark[i]);
      if mark[i]=true then writexy(69,3+i,14+blink,1,chr(17))
      else writexy(69,3+i,14+blink,1,' ');
    end;
  if ch=#0 then ch:=readkey;
  case upcase(ch) of
    #72 : dec(i);
    #80 : inc(i);
    #35 : help;
  end;
  if (ch=#13) and (i=6) then
    begin
      Input_Cyclone_Dat; ch:=' ';
      mark[i]:=true;
      writexy(69,3+i,14+blink,1,chr(17))
    end;
  if i<1 then i:=6;
  if i>6 then i:=1;
  textbackground(0); textcolor(15);

```



```

gotoxy(39,3+oldi); write(m[oldi]);
textbackground(15); textcolor(0);
gotoxy(39,3+i); write(m[i]);
textbackground(0); textcolor(15);
until ch in[#27,#13];
if ch=#13 then mark[i]:=true;
box(38,3,67,9,14,2);
end;
{-----}
procedure sub_menu2;
var i : byte;
    ch : char;
begin
    curseron;
    box(38,3,62,12,14,2);
    window(39,4,61,11);
    clrscr;
    window(1,1,80,25);
    writexy(66,24,11,0,[' Esc-Quit ']);
    gotoxy(39,4); write(' Q  = ',Q:13:5);
    gotoxy(39,5); write(' PD = ',Pp:13:5);
    gotoxy(39,6); write(' FD = ',P:13:5);
    gotoxy(39,7); write(' DL = ',DL:13:5);
    gotoxy(39,8); write(' Temp = ',T:13:3);
    gotoxy(39,9); write(' ๑  = ',U:13:5);
    gotoxy(39,10); write(' Dcpr = ',Dcpr:13:7);
    gotoxy(39,11); write(' Vin = ',Vin:13:3);
    gotoxy(48,4);
    i:=1;
    repeat
        gotoxy(48,3+i);

```

```
ch:=readkey;
if ch=#27 then exit;
if ch=#0 then ch:=readkey;
case upcase(ch) of
  #72 : dec(i);
  #80 : inc(i);
  #35 : help;
end;
if i>8 then i:=1;
if i<1 then i:=8;
if ch in ['0'..'9','.'] then
begin
  writexy(48,3+i,15,0,' ');
  gotoxy(48,3+i);
  case i of
    1 : trns_ch_val(Q,ch);
    2 : trns_ch_val(Pp,ch);
    3 : trns_ch_val(P,ch);
    4 : trns_ch_val(DL,ch);
    5 : trns_ch_val(T,ch);
    6 : trns_ch_val(U,ch);
    7 : trns_ch_val(Dcpr,ch);
    8 : trns_ch_val(Vin,ch);
  end;
  inc(i);
  if i<1 then i:=7;
  if i>7 then i:=1;
  update_oper_data;
end;
until ch=#27;
curseroff;
```

```

end;
{-----}
procedure sub_menu3;
var i : byte;
    ch : char;
begin
    curseron;
    box(38,3,67,7,14,2);
    window(39,4,66,6);
    clrscr;
    window(1,1,80,25);
    writexy(66,24,10,0,[' Esc-Quit ']);
    gotoxy(39,4); write(' Ce = ',Ce:10:5);
    gotoxy(39,5); write(' Yrs = ',Yrs:10:5);
    gotoxy(39,6); write(' Hpy = ',Hpy:10:5);
    gotoxy(47,4); i:=1;
    repeat
        gotoxy(48,3+i);
        ch:=readkey;
        if ch=#0 then ch:=readkey;
        case upcase(ch) of
            #72 : dec(i);
            #80 : inc(i);
            #35 : help;
        end;
        if i>3 then i:=1;
        if i<1 then i:=3;
        if ch in ['0'..'9','.') then
            begin
                writexy(48,3+i,15,0,' ');
                gotoxy(48,3+i);
            end;
        end;
    until ch=#27;
end;

```

```

case i of
    1 : trns_ch_val(Ce,ch);
    2 : trns_ch_val(Yrs,ch);
    3 : trns_ch_val(Hpy,ch);
end;

inc(i);

if i<1 then i:=3;
if i>3 then i:=1;
update_cost_data;
end;

until ch=#27;

curseroff;
end;
{-----}
procedure sub_menu4;
var m : array[1..3] of string[28];
    i,oldi : byte;
    ch : char;
begin
    m[1]:= ' Thickness = 1.5 mm      ';
    m[2]:= ' Thickness = 2.0 mm      ';
    m[3]:= ' Thickness = 2.5 mm      ';
    box(38,3,67,7,14,2);
    textcolor(2);
    writexy(52,24,10,0,[' Enter-Mark  Esc-Quit  ']);
    textbackground(0); textcolor(15);
    for i:=1 to 3 do writexy(39,3+i,15,0,m[i]);
    textbackground(15); textcolor(0);
    gotoxy(39,4); write(m[1]);
    textbackground(0); textcolor(15);
    i:=1;

```

```

repeat
oldi:=i;
case i of
  1 : begin
      Thk:=1.5; Roc:=300 ; Mac:=25; Fac:=18; W:=14.72;
    end;
  2 : begin
      Thk:=2.0; Roc:=350 ; Mac:=25; Fac:=18; W:=19.62;
    end;
  3 : begin
      Thk:=2.5; Roc:=420 ; Mac:=25; Fac:=18; W:=24.53;
    end;
end;
update_cost_data;
ch:=readkey;
if ch=#0 then ch:=readkey;
case upcase(ch) of
  #72 : dec(i);
  #80 : inc(i);
  #35 : help;
end;
if i<1 then i:=3;
if i>3 then i:=1;
textbackground(0); textcolor(15);
gotoxy(39,3+oldi); write(m[oldi]);
textbackground(15); textcolor(0);
gotoxy(39,3+i); write(m[i]);
textbackground(0); textcolor(15);
until ch in[#27,#13];
end;
{-----}

```

```

procedure chk_Cal_Over;
var  x1,y1,x2,y2,x,y,i : byte;
     ch : char;
begin
  curseron;
  x1:=20; x2:=60;
  y1:=8;  y2:=12;
  save(x1,y1,x2,y2);
  box(x1,y1,x2,y2,14,2);
  window(x1+1,y1+1,x2-1,y2-1);
  clrscr;
  window(1,1,80,25);
  writexy(25,10,15,0,'Calculate OverAll Eff. ? <Y/N>');
  repeat
    writexy(40,11,15,0,''); ch:=readkey;
  until upcase(ch) in['Y','N'];
  restore(x1,y1,x2,y2);
  case upcase(ch) of
    'Y' : OverAll:=true;
    'N' : OverAll:=false;
  end;
  curseroff;
end;
{-----}
procedure sub_menu5;
var  x1,y1,x2,y2,x,y,i : byte;
     ch : char;
begin

  x1:=15; x2:=65;
  y1:=6;  y2:=17;

```

```

save(x1,y1,x2,y2);
box(x1,y1,x2,y2,14,4);
writexy(32,6,10,0,[' Overall Eff.']);
writexy(53,17,10,0,[' Esc-Quit ']);
window(x1+1,y1+1,x2-1,y2-1);
clrscr;
window(1,1,80,25);
writexy(x1+1,y1+1,10,1, '      Range      Avg. Dp      Mi      ');
writexy(x1+1,y1+2,1,10, '      (x E-6 m.) . (x E-6 m.) (g.)  ');
writexy(x1+1,y1+4,15,0, '      0 - 5      2.5');
writexy(x1+1,y1+5,15,0, '      <5 - 10     7.5');
writexy(x1+1,y1+6,15,0, '      <10 - 15    12.5');
writexy(x1+1,y1+7,15,0, '      <15 - 20    17.5');
writexy(x1+1,y1+8,15,0, '      <20 - 30    25.0');
writexy(x1+1,y1+9,15,0, '      <30 - 50    40.0');
writexy(x1+1,y1+10,15,0, '      <50 - 100   75.0');
x:=55; y:=y1+3; i:=1;
cursoron;
repeat
  gotoxy(x,y+i);
  ch:=readkey;
  if ch=#0 then ch:=readkey;
  case upcase(ch) of
    #72 : dec(i);
    #80 : inc(i);
    #35 : help;
  end;
  if i<1 then i:=7;
  if i>7 then i:=1;
  if ch in ['0'..'9','.'] then
    begin

```

```

writexy(x,y+i,15,0,' ');
writexy(x,y+i,15,0,'');
trns_ch_val(Mi[i],ch);
inc(i);
end;
if i<1 then i:=7;
if i>7 then i:=1;
until ch=#27;
curseroff;
restore(x1,y1,x2,y2);
end;
{-----}
procedure Input_Cyclone_Dat;
var ch : char;
    i,x,y,pg,ii,maxi : byte;
    C : real;
begin
x:=9; y:=14;
i:=1; pg:=1;
maxi:=6;
curseron;
repeat
gotoxy(x,y+i);
ch:=readkey;
if ch=#0 then ch:=readkey;
case upcase(ch) of
#72 : dec(i);
#80 : inc(i);
#73 : dec(pg);
#81 : begin
inc(pg); i:=1;

```



```

        end;
        #35 : help;
    end;
    if i<1 then i:=maxi;
    if i>maxi then i:=1;
    if pg<=1 then
        begin
            pg:=1; maxi:=6;
        end;
    if pg>=2 then
        begin
            pg:=2; maxi:=5;
            gotoxy(4,15); write('Cyh= ',Cyh:7:3);
            gotoxy(4,16); write('B = ',B:7:3);
            gotoxy(4,17); write('H = ',H:7:3);
            gotoxy(4,18); write('De = ',De:7:3);
            gotoxy(4,19); write('S = ',S:7:3);
            gotoxy(4,20); write(' ');
        end;
    if pg=1 then
        begin
            multi_cyclone(6);
            update_cyclone_data(6);
        end;
    ii:=i+((pg-1)*6);
    if ch in ['0'..'9','.'] then
        begin
            writexy(x,y+i,15,0,' ');
            writexy(x,y+i,15,0,'');
            case ii of
                1 : trns_ch_val(K1,ch);

```

```

2 : trns_ch_val(Ka1,ch);
3 : trns_ch_val(Kb1,ch);
4 : trns_ch_val(Nh1,ch);
5 : trns_ch_val(CD,ch);
6 : begin
    trns_ch_val(C,ch); Ac:=round(C);
    end;
7 : trns_ch_val(Cyh,ch);
8 : trns_ch_val(B,ch);
9 : trns_ch_val(H,ch);
10 : trns_ch_val(De,ch);
11 : trns_ch_val(S,ch);
    end;
inc(i);
if i<1 then i:=maxi;
if i>maxi then i:=1;
    end;
until ch=#27;
multi_cyclone(i);
curseroff;
end;
{-----}
procedure chk_printer;
var Regs : registers;
    x1,x2,y1,y2 : byte;
    ch : char;
    t : word;
begin
    prt:=true;
    curseroff;
    Regs.AH:=2;

```

```

Regs.DX:=0;
Intr($17,Regs);
if (Regs.AH = $80) or (Regs.Ah=$39) then
begin
  textmode(co80);
  x1:=20; x2:=60;
  y1:=8; y2:=12;
  save(x1,y1,x2,y2);
  box(x1,y1,x2,y2,10,0);
  window(x1+1,y1+1,x2-1,y2-1);
  textbackground(5);
  clrscr;
  window(1,1,80,25);
  writexy(37,8,9,0,['ERROR']);
  writexy(31,10,9+blink,5,'Printer Not Ready');
  write(chr(7));
  delay(4000);
  prt:=false;
end;
end;
{-----}
procedure trns_ch_val(var value : real; ch : char);
label last;
var st,tmp : string;
    code,i : integer;
begin
  st:='';
  repeat
    if not(ch in['0'..'9','.']) then
      begin
        if (ch=#08) and (length(st)>0) then

```

```

begin
  tmp:=st; st:='';
  for i:=1 to length(tmp)-1 do
    st:=st+tmp[i];
    gotoxy(wherex-1,wherey); write(' ');
    gotoxy(wherex-1,wherey);
  end
  else write(chr(7));
  goto last;
end else
begin
  write(ch);
  st:=st+ch;
  end;
  last:ch:=readkey;
  until ch=#13;
  val(st,value,code);
end;
{-----}
procedure update_cyclone_data(i : byte);
begin
  gotoxy(4,15); write('K = ',K:7:3);
  gotoxy(4,16); write('Ka = ',Ka:7:3);
  gotoxy(4,17); write('Kb = ',Kb:7:3);
  gotoxy(4,18); write('Nh = ',Nh:7:3);
  if i=6 then
    begin
      writexy(4,19,15,0,'CD = '); write(CD :7:3,' ');
      writexy(4,20,15,0,'AC = '); write(AC :7,' ');
    end else
    begin

```

```

        writexy(4,19,0,5,' ');
        writexy(4,20,0,5,' ');
    end;
end;
{-----}
procedure update_oper_data;
begin
    gotoxy(21,15); write('Q = ',Q:10:7,' m3/s');
    gotoxy(21,16); write('PD = ',Pp:10:5,' Kg/m3');
    gotoxy(21,17); write('FD = ',P:10:7,' Kg/m3');
    gotoxy(21,18); write('DL = ',DL:10:7,' Kg/m3');
    gotoxy(21,19); write('Temp = ',T:10:3,' K(Temp)');
    gotoxy(21,20); write('η = ',U:10:7,' Kg/s.m');
    gotoxy(21,21); write('Dcpr = ',Dcpr:10:7,' m');
    gotoxy(21,22); write('Vin = ',Vin :10:3,' m/s');
end;
{-----}
procedure update_cost_data;
begin
    gotoxy(51,15); write('Ce = ',Ce:10:5,' B/Kw.hr');
    gotoxy(51,16); write('Yrs = ',Yrs:10:5,' Years');
    gotoxy(51,17); write('Hpy = ',Hpy:10:5,' Hr/Yrs');
    gotoxy(51,18); write('Thk = ',Thk:10:3,' mm');
    gotoxy(51,19); write('Roc = ',Roc:10:3,' B/m2');
    gotoxy(51,20); write('Mac = ',Mac:10:3,' B/Kg');
    gotoxy(51,21); write('Fac = ',Fac:10:3,' B/Kg');
    gotoxy(51,22); write('Wt = ',W:10:3,' Kg/m2');
end;
{-----}
procedure multi_Cyclone(typ : byte);
begin

```

```

case typ of
  1 : begin
      Ka:=0.5; Kb:=0.2; K:=551.3; Nh:=6.4;
    end;
  2 : begin
      Ka:=0.44; Kb:=0.21; K:=699.2; Nh:=9.24;
    end;
  3 : begin
      Ka:=0.5; Kb:=0.25; K:=402.9; Nh:=8.0;
    end;
  4 : begin
      Ka:=0.5; Kb:=0.25; K:=381.8; Nh:=8.0;
    end;
  5 : begin
      Ka:=0.583; Kb:=0.208; K:=342.3; Nh:=7.76;
    end;
  6 : begin
      Ka:=Ka1; Kb:=Kb1; K:=K1; Nh:=Nh1;
    end;
end;
end;
{-----}
function power(base,pwr : real) : real;
begin
  power:=exp(pwr*ln(base));
end;
{-----}
procedure cal_vmax;
begin
  Vmax:=power(5000/((DL+P)*Nh),1/2);
  writeln(data,'Vmax = ',Vmax:6:3,' m/sec .... (1)');

```

```

end;
{-----}
procedure _1st_cons;
var x1,x2,y1,y2 : byte;
    ch : char;
begin
    if Vmax>30 then Vmax:=30;
    if Vmax<15 then
        begin
            restorecrtmode;
            cursoroff;
            write(chr(7));
            x1:=15; y1:=5;
            x2:=65; y2:=10;
            save(x1,y1,x2,y2);
            box(x1,y1,x2,y2,14,4);
            writexy(x2-9,y1,9+blink,4,['Error!']);
            window(x1+1,y1+1,x2-1,y2-1);
            textbackground(4); clrscr;
            writeln;
            writeln('          PLEASE INPUT NEW DATA  ');
            write('          CAUSE OF INLET VELOCITY <15 m/s');
            delay(4000);
            error:=true;
            window(1,1,80,25);
            restore(x1,y1,x2,y2);
        end;
    end;
end;
{-----}
procedure find_Da;
var M : real;

```

```

begin
  Da:=0.3;
  repeat
    Da:=Da+0.001;
    nn_No1:=1-(1-0.67*power(Da,0.14))*power(T/283,0.3);
    M:=2*power((K*Q*Pp*(nn_No1+1)/(18*U*power(Da,3))),1/(2*nn_No1+2));
    Dcp_No1:=power((0.6931/M),(nn_No1+1));
  until (Dcpr-Dcp_No1)<0.0000001;
  D:=Da;
  writeln(data,'D = ',D:6:3,' m .... (2)');
  writeln(data,'nn_No1 = ',nn_No1:8:6);
  writeln(data,'M = ',M:10:3);
  writeln(data,'Dcp_No1 = ',Dcp_No1:9:7,' m .... (2)');
end;
{-----}
procedure find_check_Vs;
var ven : real;
begin
  Vs:=4.91*power(39.24*u*(Pp-P)/(3*P*P),1/3)*power(Kb,0.4)*power(D,0.067)
    *power(Vmax,2/3)/power(1-Kb,1/3);
  if Vmax>1.35*Vs then chk_Vs_Error(ven);
end;
{-----}
procedure chk_Vs_Error(ven : real);
var ans : char;
    size : integer;
    pic : pointer;
begin
  error:=true;
  writeln(data,'NOW PARTICLE RE-ENTRAINMENT .....(5)');
end;

```



```

{-----}
procedure _3rd_5th_cons;
var temp : real;
begin
  temp:=power(Dcpr,4);
  Nmin:=trunc((O/(Ka*Kb*Vmax*power(D,2))));
  if Nmin=0 then Nmin:=1;
  Nmax:=trunc((O/(Ka*Kb*15*power(D,2)))+1);
  writeln(data,'Nmin = ',Nmin,' ...'(3));
  writeln(data,'Nmax = ',Nmax,' ...'(3));
end;
{-----}
procedure cyclone_error;
var ans : char;
begin
  textmode(co80);
  cursoroff;
  ans:=' ';
  save(22,10,58,14);
  box(22,10,58,14,14,10);
  window(23,11,59,13);
  writexy(12,1,4+blink,0,'WARNING');
  writexy(6,2,14,0,'CYCLONE TOO SMALL/LARGE');
  ans:=readkey;
  window(1,1,80,25);
  restore(22,10,58,14);
  error:=true;
  writeln(data,'CYCLONE TOO SMALL/LARGE');
end;
{-----}
procedure flow_error;

```

```

var ans : char;
    cy_txt : string[17];
begin
    textmode(co80);
    cursoroff;
    ans:= ' ';
    case tpe of
        1 : Cy_txt:='Stairmand a';
        2 : Cy_txt:='Hi_Eff swift b';
        3 : Cy_txt:='Lapple c';
        4 : Cy_txt:='Gen_Pur Swift b';
        5 : Cy_txt:='Peter & Whitby d';
        6 : Cy_txt:='User Type';
    end;
    save(22,10,58,14);
    box(22,10,58,14,14,10);
    window(23,11,59,13);
    writexy(14,1,4+blink,0,'WARNING');
    writexy(14,2,15,0,Cy_txt);
    writexy(6,3,14,0,'VOLUME FLOW RATE TOO HIGH');
    ans:=readkey;
    window(1,1,80,25);
    restore(22,10,58,14);
    writeln('VOLUME FLOW RATE TOO HIGH ....(10)');
    error:=true;
end;
{-----}

procedure loop;
label start,last;
var N : integer;
    M,Ven,cont,RPa,Lo : real;

```

```
golast: boolean;
txt : string[20];
begin
count:=0;
golast := false;
case c_type of
  1 : begin
      txt:='Stairmaid a';
      cont:=5.1875;
      end;
  2 : begin
      txt:='Hi Eff Swift b';
      cont:=5.1;
      end;
  3 : begin
      txt:='Lapple c';
      cont:=4.8125;
      end;
  4 : begin
      txt:='Gen Pur Swift b';
      cont:=4.85;
      end;
  5 : begin
      txt:='Peter & Whitby d';
      cont:=4.38;
      end;
  6 : begin
      txt:='User Type';
      cont:=(Cyh+(1+B)*(H-Cyh)+De*S)
      end;
end;
```

```

writeln(data,'');
writeln(data,'Type : ',txt);
writeln(data,'=====');
if user = true then
begin
  No:=AC;
  Nmin:=AC; Nmax:=AC;
  D:=CD;
end;
N := Nmin;
repeat
  Ven := Q/(N*Ka*D*Kb*D);
  Vs:=4.91*power(39.24*u*(Pp-P)/(3*P*P),1/3)*power(Kb,0.4)*power(D,0.067)
    *power(Ven,2/3)/power(1-Kb,1/3);
  writeln(data,'N = ',N,' Ven = ',Ven:6:3,' Vs = ',Vs:6:3,' m/s ....(4)');
  if (Ven>1.35*Vs) and (user=false) then
  begin
    chk_Vs_Error(ven);
    exit;
  end;
  if (Ven < Vin) and (user=false) then
  begin
    start:repeat
      D := D-0.00001;
      Ven := Q/(N*Ka*D*Kb*D);
      Vs:=4.91*power(39.24*u*(Pp-P)/(3*P*P),1/3)*power(Kb,0.4)*power(D,0.067)
        *power(Ven,2/3)/power(1-Kb,1/3);
      until abs(ven-Vin)<=0.0002;
      if Ven>1.35*Vs then
      begin

```

```

    chk_Vs_Error(ven);
    exit;
end;
writeln(data,'Ven = ',Ven:6:3,' m/s ....(6)');
writeln(data,'D = ',D:6:3,' m ....(6)');
golast := true
end;
nn:=1-(1-0.67*power(D,0.14))*power(T/283,0.3);
writeln(data,'nn = ',nn:7:5,' ....(7)');
if (nn<0.5) or (nn>0.9) then
begin
    cyclone_error;
    goto last;
end;
PDr:=0.5*(DL+P)*Nh*power(Q/(Ka*Kb*power(D,2)*N),2);
writeln(data,'PDr = ',PDr:9:4,' N/m2 ....(9)');
if (PDr>2500) and (user=true) then
begin
    flow_error;
    goto last;
end;
if PDr>2500 then goto last;
M:=2*power((K*Q/N*Pp*(nn+1)/(18*U*power(D,3))),1/(2*nn+2));
Dcp:=power((0.6931/M),(nn+1));
writeln(data,'M = ',M:10:3,' ....(11)');
writeln(data,'Dcp = ',Dcp:10:8,' m ....(11)');
if user=false then
begin
    if Dcp>0.85*Dcpr then inc(Nmax);
    if Dcp>Dcpr then
begin

```

```

writeln(data,'Now Dcp>Dcpr ....(12)');
goto start;
end;
end;
end;
Obj:=(1-exp(-M*power(Dcpr,1/(nn+1))))/PDr;
Wcy:=Acy*W;
Mcy:=Wcy*Mac;
Fcy:=Wcy*Fac;
RPa:=0;
Lo:=0;
Lo:=trunc(D/0.385)+1;
if (Lo>=1) and (2>=Lo) then RPa:=1.20;
if (Lo>=3) and (5>=Lo) then RPa:=1.10;
if (Lo>=6) then RPa:=1.15;
Rcy:=Roc*RPa*Acy;
Tfc:=(Mcy+Fcy+Rcy);
Th:=Yrs*Hpy;
Toc:=Q*Ce*PDr*Th,

inc(count);
Nc[count]:=N;
D_No[Nc[count]]:=D;
nn_No[Nc[count]]:=nn;

PDr_No[Nc[count]]:=PDr;
Dcp_No[Nc[count]]:=Dcp;
Obj_No[Nc[count]]:=Obj;
Vs135_No[count]:=1.35*Vs;
Tfc_No[Nc[count]]:=Tfc;
Toc_No[Nc[count]]:=Toc/1000;
Tco[Nc[count]]:=Tfc_No[Nc[count]]+Toc_No[Nc[count]];

```



```

writeln(data,'Obj = ',Obj:10:8,' ....(13)');
writeln(data,'Acy = ',Acy:10:5,' m2');
writeln(data,'Wcy = ',Wcy:10:5,' kg');
writeln(data,'Mcy = ',Mcy:11:2,' baht');
writeln(data,'Fcy = ',Fcy:11:2,' baht');
writeln(data,'Rcy = ',Rcy:11:2,' baht');
writeln(data,'Tfc = ',Tfc:11:2,' baht');
writeln(data,'Th = ',Th:11:2,' hr');
writeln(data,'Toc = ',Toc/1000:11:2,' baht');
writeln(data,'1.35Vs_No = ',1.35*Vs:8:4,' m/s');
writeln(data,'Tco = ',Tco[Nc[count]]:11:2,' baht ....(13)');
if user=true then Tc_No:=Tco[Nc[count]];
if golast then goto last;
last.inc(N);
until (N=Nmax+1) or (user=true);
user:=false;
end;
{-----}
procedure _7th_cons_step15;
var i,ii : integer;
    temp : real;
begin
    No:=0;
    for i:=1 to count do
        if Tco[nc[1]]>Tco[nc[i]] then
            begin
                No:=nc[i];
                temp:=Tco[nc[1]];
                Tco[nc[1]]:=Tco[nc[i]];
                Tco[nc[i]]:=temp;
                Tc_No:=Tco[nc[1]];
            end
        end
    end;

```

ศูนย์วิทยทรัพยากร
มหาวิทยาลัยราชภัฏมหาสารคาม

```

end;
if (No=0) and (count=2) and (Tco[nc[1]]<Tco[nc[2]]) then
begin
  No:=nc[1];
  temp:=Tco[nc[1]];
  Tco[nc[1]]:=Tco[nc[i]];
  Tco[nc[i]]:=temp;
  Tc_No:=Tco[2];
end;
if (No=0) and (count =1) then
begin
  No:=Nc[1];
  Tc_No:=Tco[No];
end;
writeln(data,'Tc_no = ',Tc_no:11:2,' baht ....(14)');
if (No=0) then
begin
  for i:=1 to count do
  if Obj_No[nc[1]]<Obj_No[nc[i]] then
  begin
    No:=Nc[i];
    temp:= Tco[nc[1]];
    Tco[nc[1]]:=Tco[nc[i]];
    Tco[nc[i]]:=temp;
    Tc_No:=Tco[nc[1]];
  end;
  writeln(data,'Same Tc_no Then Max Obj_no ',Obj_no[no]:10:8,' ....(14.1)');
end;

end;
{-----}

```



```

procedure data_file;
begin
if mark[6]=true then
    begin
        writeln(data,'CD = ',CD:10:3);
        writeln(data,'AC = ',AC:10);
    end;
writeln(data,'Q = ',Q:10:7,' m3/s');
writeln(data,'PD = ',Pp:10:5,' Kg/m3');
writeln(data,'FD = ',P:10:7,' Kg/m3');
writeln(data,'DL = ',DL:10:7,' Kg/m3');
writeln(data,'Temp = ',T:10:3,' K(Temp)');
writeln(data,'ϒ = ',U:10:7,' Kg/s.m');
writeln(data,'Dcpr = ',Dcpr:10:7,' m');
writeln(data,'Ce = ',Ce:10:5,' B/Kw.hr');
writeln(data,'Yrs = ',Yrs:10:5,' Years');
writeln(data,'Hpy = ',Hpy:10:5,' Hr/Yrs');
writeln(data,'Thk = ',Thk:10:3,' mm');
writeln(data,'Roc = ',Roc:10:3,' B/m2');
writeln(data,'Mac = ',Mac:10:3,' B/Kg');
writeln(data,'Fac = ',Fac:10:3,' B/Kg');
writeln(data,'Wt = ',W:10:3,' Kg/m2');
end;
{-----}
procedure Shw_Dat(xpos, ypos, Cy_Typ : integer);
var step : byte;
    st : string[20];
    i : integer;
begin
    step:=20;
    setttextjustify(righttext,centertext);

```

```

setcolor(3);
case Cy_Typ of
  1 : Cy:='Stairmand a';
  2 : Cy:='Hi Eff Swift b';
  3 : Cy:='Lapple c';
  4 : Cy:='Gen Pur Swift b';
  5 : Cy:='Peter & Whitby d';
  6 : Cy:='User Type';
end;
setcolor(14);
outtextxy(xpos,ypos,Cy);
setcolor(15);
str(No:3,st); outtextxy(xpos,ypos+step,st);
str(D_No[No]:5:6,st); outtextxy(xpos,ypos+2*step,st);
str(PDr_No[No]:10:5,st); outtextxy(xpos,ypos+3*step,st);
str(trunc(PDr_No[No]),st);
  for i:=1 to length(st)-1 do
    if ((i mod 3)=0) then outtextxy(xpos-44-8*(i),ypos+3+3*step,',');
str(Dcp_No[No]:10:8,st); outtextxy(xpos,ypos+4*step,st);
str(Dcpr:10:8,st); outtextxy(xpos,ypos+5*step,st);
str(Obj_No[No]:12:8,st); outtextxy(xpos,ypos+6*step,st);
str(Tfc_No[No]:10:2,st); outtextxy(xpos,ypos+7*step,st);
str(trunc(Tfc_No[No]),st);
    for i:=1 to length(st)-1 do
      if ((i mod 3)=0) then outtextxy(xpos-20-8*i,ypos+3+7*step,',');
str(Toc_No[No]:10:2,st); outtextxy(xpos,ypos+8*step,st);
str(trunc(Toc_No[No]),st);
    for i:=1 to length(st)-1 do
      if ((i mod 3)=0) then outtextxy(xpos-20-8*i,ypos+3+8*step,',');
str(Tc_No:10:2,st); outtextxy(xpos,ypos+9*step,st);
str(trunc(Tc_No),st);

```

```

for i:=1 to length(st)-1 do
  if ((i mod 3)=0) then outtextxy(xpos-20-8*i,ypos+3+9*step,',');
if OverAll=true then
  begin
    str(Sum_nigi:10:2,st); outtextxy(xpos,ypos+10*step,st);
  end;
end;
{-----}
procedure auto_cal;
label again,next;
var i,ii,xpos,ypos,xstep,x,y,t,col : integer;
    ch : char;
    Cy_txt,cy : string[15];
begin
  ii:=0;
  for i:=1 to 6 do
    if mark[i]=true then inc(ii);
  if ii=0 then
    begin
      write(chr(7));
      exit;
    end;
  again:xpos:=360; ypos:=25; xstep:=134;
  win_shw_cal; ii:=0;
  col:=0;
  for i:=1 to 6 do
    begin
      if mark[i]=true then
        begin
          tpe:=i;
          multi_cyclone(i);

```

```
c_type:=i;
inc(ii);
  if ii=4 then
    begin
      inc(ypos,220);
      xpos:=360;
    end;
  if i<>6 then cal else
    begin
      user:=true;
      loop;
    end;
  if (error=true) and (i<>6) then
    begin
      setcolor(4);
      case i of
        1 : Cy:=' Stairmand a';
        2 : Cy:='Hi Eff Swift b';
        3 : Cy:=' Lapple c';
        4 : Cy:='Gen Pur Swift b';
        5 : Cy:='Peter & Whitby d';
        6 : Cy:=' User Type';
      end;
      outtextxy(xpos-120,ypos,cy);
      goto next;
    end;
  Cal_PlotGraph(i);
  Cal_PlotGraph_N(i);
  if OverAll=true then Cal_Over;
  shw_dat(xpos,ypos,i);
next:inc(xpos,xstep);
```

```

    end;
end;
if ii=0 then exit;
i:=0;
Adj_Obj_ystep;
ch:=readkey;
repeat
case upcase(ch) of
'D': goto again;
'P': begin
    chk_printer;
    if prt=true then
    begin
        print;
        ch:=readkey;
    end else exit;
end;
'V': begin
    Drw_Cyclone;
    Cyclone_txt_Dat;
    ch:=readkey;
end;
'G','O' : begin
    case upcase(ch) of
        'O' : t:=1;
        'G' : t:=2;
    end;
    clearviewport;
    setttextjustify(0,0);
    for ii:=1 to 6 do
        if mark[ii]=true then PlotGraph(t,ii);

```

```

repeat
  ch:=readkey;
  if ch=#0 then
    begin
      ch:=readkey;
      case upcase(ch) of
        #81 : inc(i);
        #73 : dec(i);
      end;
      repeat
        if (mark[i]<>true) and (ch=#81) then inc(i);
        if (mark[i]<>true) and (ch=#73) then dec(i);
      until (mark[i]=true) or (i=7);
      if i>7 then i:=7;
      if i<1 then i:=1;
      if (mark[i]=true) or (i<7) then

        begin
          clearviewport;
          PlotGraph(t,i);
        end;
      case i of
        1 : Cy_txt:='Stairmaid a';
        2 : Cy_txt:='Hi Eff Swift b';
        3 : Cy_txt:='Lapple c';
        4 : Cy_txt:='Gen Pur Swift b';
        5 : Cy_txt:='Peter & Whitby d';
        6 : Cy_txt:='User Type';
      end;
      setcolor(13);
      settxtstyle(0,0,2);

```

```

    settextjustify(1,1);
    outtextxy(getmaxx div 2,25,Cy_txt);
    settextjustify(0,0);
    settextstyle(0,0,1);
    if i=7 then
        begin
            clearviewport;
            for ii:=1 to 6 do
                if mark[ii]=true then PlotGraph(t,ii);
            end;
        end;
    until upcase(ch) in ['G','O','V','D','P',#27];
end;
end;
until ch=#27;
textmode(co80);
end;
{-----}

```

```

procedure cal;
var ch : char;
begin
    cal_Vmax;
    _1st_cons;
    if user=false then find_Da;
    _3rd_5th_cons;
    loop;
    _7th_cons_step15;
end;
{-----}

```

```

procedure scale(typ,pwr: integer);

```

```

var  x,y : integer;
     st : string;
     yscale : integer;
begin
  settextstyle(0,0,1);
  setlinestyle(1,0,1);
  for x:=1 to 16 do
    begin
      setcolor(8);
      line(x*30+70,380,x*30+70,80);
      str((x-1),st);
      setcolor(1);
      outtextxy(x*30+65,400,st);
    end;
  case typ of
    1 : for y:=1 to 11 do
        begin
          setcolor(8);
          line(100,y*30+50,550,y*30+50);
          yscale:=ystep*(y-1);
          str(yscale,st);
          setcolor(1);
          outtextxy(50,(11-(y-1))*30+50,st);
        end;
    2 : for y:=1 to 11 do
        begin
          setcolor(8);
          line(100,y*30+50,550,y*30+50);
          yscale:=ystep_Eff*(y-1);
          str(yscale,st);
          setcolor(1);

```



```

    outtextxy(50,(11-(y-1))*30+50,st);
end;
end;
settextstyle(0,0,1);
setcolor(5);
outtextxy(550,425,'x 10');
outtextxy(580,415,'- 6');
outtextxy(70,60,'x 10');
str(pwr,st);
outtextxy(100,50,'- '+st);
setcolor(14);
settextstyle(1,0,2);
outtextxy(525,445,'Dp (m)');
settextstyle(1,1,2);
case typ of
  1 : outtextxy(15,150,'Objective. ');
  2 : outtextxy(15,150,'Grade Eff. ');
end;
setlinestyle(0,0,1);
end;
{-----}
procedure Cal_Over;
var M,t,tt : real;
    i : byte;
begin
  Sum_Mi:=0;
  Sum_nigi:=0;
  M:=2*power((K*Q/No*Pp*(nn_No[No]+1)/(18*U*power(D_No[No],3))),1/(2*nn_No[No]+2));
  for i:=1 to 7 do Sum_Mi:=mi[i]+Sum_Mi;
  for i:=1 to 7 do
    begin

```

```

gi[i]:=Mi[i]*100/Sum_Mi;
t:=power(Avg_Dp[i],1/(nn_No[No]+1));
tt:=exp(-M*t);
ni[i]:=1-tt;
Sum_nigi:=gi[i]*ni[i]+Sum_nigi;
writeln(data,'Mi['',i,''] = ',mi[i]:8:4,' Gi['',i,''] = ',gi[i]:6:4,
          ' Ni['',i,''] = ',ni[i]:8:4,' S_nigi['',i,''] = ',Sum_nigi:6:2);
end;
end;
{-----}
procedure Cal_PlotGraph_N(cy : integer);
var count,D : byte;
    Obj_N,n,Dp,M : real;
    st : string;
begin
n:=nn_No[No];
pwr:=6;
repeat
D:=0;
st:='';
for count:=1 to 15 do
begin
inc(D);
Dp:=D*power(10,-6);
M:=2*power((K*Q/No*Pp*(n+1))/(18*U*power(D_No[No],3)),1/(2*n+2));
Obj_N:=(1-exp(-M*power(Dp,1/(n+1))))/PDr_No[No];
Obj_graph[cy,count]:=round(Obj_N*power(10,pwr));
str(Obj_graph[cy,count],st);
len[cy]:=length(st);
writeln(data,count:2,' Obj_N = ',Obj_N:10:8,' M = ',M:10:3,
        ' v_n = ',n:7:5);

```

```

    end;
    dec(pwr);
    until len[cy]<=6;
    inc(pwr);
end;
{-----}
procedure Adj_Obj_ystep;
var maxlen,i,temp,cy : byte;
begin
    maxlen:=len[1];
    for i:=2 to 6 do
        if len[1]<len[i] then
            begin
                temp:=len[i];
                len[i]:=len[i+1];
                len[i+1]:=temp;
                maxlen:=len[1];
                case maxlen of
                    1 : ystep:=1;
                    2 : ystep:=10;
                    3 : ystep:=100;
                    4 : ystep:=1000;
                    5 : ystep:=10000;
                    6 : ystep:=100000;
                end;
            end;
    end;
    Max_Obj:=Obj_Graph[1,15];
    for i:=1 to 5 do
        begin
            if mark[i]=true then
                if Obj_Graph[i,15]>Obj_Graph[i+1,15] then

```

```

Max_Obj:=Obj_Graph[i,15] else Max_Obj:=Obj_Graph[i+1,15];
end;
ystep:=round((Max_Obj/ystep)+1)*ystep div 10;
end;
{-----}
procedure Cal_PlotGraph(cy : integer);
var count,D : byte;
    G,n,Dp,M : real;
    st : string;
    len: integer;
begin
n:=nn_No[No];
pwr_Eff:=4;
repeat
D:=0;
st:='';
ystep:=1000;
for count:=1 to 15 do
begin
inc(D);
Dp:=D*power(10,-6);
M:=2*power((K*Q/No*Pp*(n+1)/(18*U*power(D_No[No],3))),1/(2*n+2));
G:=(1-exp(-M*power(Dp,1/(n+1)))));
G_Eff[cy,count]:=round(G*power(10,pwr_Eff));
str(G_Eff[cy,count],st);
len:=length(st);
if pwr_eff=4 then writeln(data,count:2,' G_Eff = ',G:10:8,' M = ',M:10:3,
' v_n = ',n:7:5);
end;
dec(pwr_Eff);
until len<=3;

```

```

inc(pwr_Eff);
case len of
  1 : ystep_Eff:=1;
  2 : ystep_Eff:=10;
  3 : ystep_Eff:=100;
  4 : ystep_Eff:=1000;
end;
end;
{-----}
procedure PlotGraph(typ,cy : integer);
begin
  gr_box(2,2,getmaxx-2,getmaxy-2,14,3);
  setcolor(2);
  outtextxy(10,getmaxy-7,'± ± ± ± ± ± ±
±');
  setcolor(9);
  outtextxy(10,getmaxy-7,' G-Grade Eff. O-Obj. P-Print V-View Cyc. D-Data Pg Up-
Dn Esc-Exit');
  case typ of
    1 : scale(typ,pwr);
    2 : scale(typ,pwr_Eff);
  end;
  settextstyle(0,0,1);
  setfillstyle(1,1); bar(40,410,50,420);
  setfillstyle(1,2); bar(40,430,50,440);
  setfillstyle(1,3); bar(190,410,200,420);
  setfillstyle(1,4); bar(190,430,200,440);
  setfillstyle(1,5); bar(340,410,350,420);
  setfillstyle(1,6); bar(340,430,350,440);

  setcolor(11);

```

```

outtextxy(55,420,'Stairmand a');
outtextxy(55,440,'High Eff-Swift b');
outtextxy(205,420,'Lapple c');
outtextxy(205,440,'Gen Pur-Swift b');
outtextxy(355,420,'Perter & Whitby d');
outtextxy(355,440,'User Define Type');
for count:=0 to 15 do
begin
  setcolor(cy);
  case typ of
  1 : begin
      if count<15 then
        line((count+1)*30+70,380-((30*Obj_graph[cy,count]) div ystep)
              ,(count+2)*30+70,380-((30*Obj_graph[cy,count+1]) div ystep))
      else
        line((count+1)*30+70,380-((30*Obj_graph[cy,count]) div ystep)
              ,(count+1)*30+70,380-((30*Obj_graph[cy,count]) div ystep))
      end;
    2 : begin
      if count<15 then
        line((count+1)*30+70,380-((30*G_Eff[cy,count]) div ystep_Eff)
              ,(count+2)*30+70,380-((30*G_Eff[cy,count+1]) div ystep_Eff))
      else
        line((count+1)*30+70,380-((30*G_Eff[cy,count]) div ystep_EFf)
              ,(count+1)*30+70,380-((30*G_Eff[cy,count]) div ystep_Eff))
      end;
    end;
  end;
end;
end;
{-----}
procedure print;

```

```

var piccode,n1,n2: byte;
    col,row,strow : integer;
    ch : char;
begin
    n2:=getmaxx div 256;
    n1:=getmaxx mod 256;
    strow:=0;
    repeat
        write(lst,#27#51#23);
        write(lst,#27#42#6+chr(n1)+chr(n2));
        for col:=0 to getmaxx do
            begin
                piccode:=0;
                for row:=0+strow to 7+strow do
                    if getpixel(col,row)<>0 then
                        case row-strow of
                            0 : piccode:=piccode+128;
                            1 : piccode:=piccode+64;
                            2 : piccode:=piccode+32;
                            3 : piccode:=piccode+16;
                            4 : piccode:=piccode+8;
                            5 : piccode:=piccode+4;
                            6 : piccode:=piccode+2;
                            7 : piccode:=piccode+1;
                        end;
                write(lst,chr(piccode));
            end;
        write(lst,#10);
        inc(strow,8);
    until strow>=getmaxx;
end;

```

```
{-----}
```

```
procedure initgr;
```

```
begin
```

```
  graphdriver := detect;
```

```
  initgraph(graphdriver,graphmode,'');
```

```
  clearviewport;
```

```
end;
```

```
{-----}
```

```
procedure gr_box(x1,y1,x2,y2,co,width : integer);
```

```
begin
```

```
  setlinestyle(0,0,width);
```

```
  setcolor(co);
```

```
  line(x1,y1,x2,y1);
```

```
  line(x1,y1,x1,y2);
```

```
  line(x2,y1,x2,y2);
```

```
  line(x1,y2,x2,y2);
```

```
  setlinestyle(0,0,1);
```

```
end;
```

```
{-----}
```

```
procedure line_arrow(tp,x1,y1,x2,y2 : integer);
```

```
begin
```

```
  line(x1,y1,x2,y2);
```

```
  case tp of
```

```
    0 : begin
```

```
      outtextxy(x1,y1-3,chr(17));
```

```
      outtextxy(x2-7,y2-3,chr(16));
```

```
    end;
```

```
    1 : begin
```

```
      outtextxy(x1-3,y1,chr(30));
```

```
      outtextxy(x2-3,y2-5,chr(31));
```

```
    end;
```



```

end;
end;
{-----}
procedure Drw_Cyclone;
var x,y,i : integer;

begin
  initgr;
  x:=100; y:=50;
  setcolor(14);
  line(x,y,x,y+100); line(x+80,y,x+80,y+100);
  line(x,y+100,x+30,y+190); line(x+80,y+100,x+50,y+190);
  line(x+25,y-7,x+25,y); line(x+55,y-7,x+55,y);

  ellipse(x+40,y-7,360,0,15,3);
  ellipse(x+40,y,180,0,15,3);
  ellipse(x+40,y,0,360,40,10);
  ellipse(x+40,y+100,180,0,40,10);
  ellipse(x+40,y+190,180,0,10,2);
  ellipse(x+40,y+190,130,45,20,10);
  setcolor(9);
  rectangle(x,y+10,x+20,y+65);

  setcolor(7);
  line(x+25,y,x+25,y+65);
  line(x+55,y,x+55,y+65);
  ellipse(x+40,y+65,180,0,15,3);
  ellipse(x+40,y+100,0,180,40,10);

  setcolor(8);
  line(x,y-40,x,y); line(x+80,y-40,x+80,y);

```

```

line(x+25,y-25,x+25,y-12); line(x+55,y-25,x+55,y-12); {De}
line(x+20,y-25,x+20,y+4); {b}
line(x+30,y+205,x+30,y+215); line(x+50,y+205,x+50,y+215); {B}

```

```

line(x+90,y,x+150,y); line(x+60,y+200,x+150,y+200); {H}
line(x+90,y+100,x+120,y+100); {Ch}
line(x+90,y+65,x+100,y+65); {s}
line(x-20,y+10,x-5,y+10); line(x-20,y+65,x-5,y+65); {a}

```

```

{write '<----->'}
setcolor(8);
line_arrow(0,x,y-35,x+80,y-35); {D}
line_arrow(0,x+25,y-20,x+55,y-20); {De}
line_arrow(0,x,y-20,x+20,y-20); {b}
line_arrow(0,x+30,y+210,x+50,y+210); {B}
line_arrow(1,x+145,y,x+145,y+200); {H}
line_arrow(1,x+115,y,x+115,y+100); {Ch}
line_arrow(1,x+95,y,x+95,y+65); {s}
line_arrow(1,x-15,y+10,x-15,y+65); {a}

```

```

{write Term character }
setcolor(9);
outtextxy(x+40,y-45,'D');
outtextxy(x+35,y-30,'De');
outtextxy(x+8,y-30,'b');
outtextxy(x+38,y+215,'B');
outtextxy(x+150,y+100,'H');
outtextxy(x+120,y+50,'Ch');
outtextxy(x+100,y+30,'s');
outtextxy(x-25,y+30,'a');

```

```

settextstyle(2,0,7);
settextjustify(centertext,0);
setcolor(8);
for i:=5 downto 1 do
begin
    if i=1 then setcolor(14);
    outtextxy(450+i,70-i,'C Y C L O N E');
    outtextxy(450+i,120-i,'D E S I G N');
    outtextxy(450+i,170-i,'C O N F I G U R A T I O N S');
end;
settextstyle(0,0,1);
settextjustify(0,0);
end;
{-----}
procedure Cyclone_Txt_Dat;
var y,i : integer;
    fi : text;
    txt : string[90];
    ch : char;
begin
    y:=290;
    assign(fi,'cyclone.dat');
    reset(fi); i:=0;
    while not Eof(fi) do
    begin
        inc(i);
        if i=2 then setcolor(14);
        if i=3 then setcolor(5);
        if i>=4 then setcolor(9);
        if (i in[6..17]) then setcolor(2);
        if i=18 then y:=351;

```

```

    if i>=18 then setcolor(15);
    readln(fi,txt);
    outtextxy(4,y,txt);
    inc(y,10);
end;
close(fi);
gr_box(3,283,getmaxx-3,335,4,1);
gr_box(3,283,getmaxx-3,getmaxy-25,4,3);
gr_box(1,1,getmaxx,getmaxy-3,5,3);
setcolor(2);
outtextxy(10,getmaxy-7,'±      ±      ±      ±      ±      ±      ๓');
setcolor(9);
outtextxy(10,getmaxy-7,' G-Grade Eff.  O-Obj.  P-Print  V-View Cyc.  D-Data  Esc-
Exit');
end;
{-----}
procedure shw_thesis;
var i : integer;
    ch : char;
begin
    initgr;
    setfillstyle(1,8);
    for i:=1 to 10 do bar(35+i,35-i,getmaxx-35+i,getmaxy-35-i);
    setfillstyle(10,9);
    bar(35,35,getmaxx-35,getmaxy-35);
    settextstyle(2,horizdir,9);
    settextjustify(center,center);
    for i:=1 to 6 do
        begin
            gr_box(1+i,10-i,getmaxx-10+i,getmaxy-i,1,3);
            setcolor(4);

```

```

    outtextxy((getmaxx div 2)+i,100-i,'OPTIMIZATION DESIGN');
    outtextxy((getmaxx div 2)+i,180-i,'FOR PARALLEL CYCLONES');
    outtextxy((getmaxx div 2)+i,260-i,'HAVING A TANGENTIAL GAS INLET');
end;

gr_box(1,10,getmaxx-10,getmaxy,6,3);
setcolor(14);
outtextxy((getmaxx div 2),100,'OPTIMIZATION DESIGN');
outtextxy((getmaxx div 2),180,'FOR PARALLEL CYCLONES');
outtextxy((getmaxx div 2),260,'HAVING A TANGENTIAL GAS INLET');
ch:=readkey;
textmode(co80);
end;
{-----}

procedure eoj;
begin
    curseron;
    ans:= ' ';
    save(28,10,52,14);
    box(28,10,52,14,14,10);
    window(29,11,51,13);
    writeln; writeln('  Quit ? < Y/N >');
    write('      ');
    repeat
        ans:=readkey;
    until upcase(ans) in['Y','N',#27];
    window(1,1,80,25);
    restore(28,10,52,14);
    curseroff;
end;
{-----}

```

```
procedure opendatfile;
begin
  if paramcount=0 then datfile:='data.dat';
  if paramcount>0 then
    datfile:=paramstr(1);
  assign(data,datfile);
  rewrite(data);
end;
{-----}
begin
  shw_thesis;
  chk_printer;
  textbackground(0);
  clrscr;
  initdata;
  opendatfile;
  repeat
    ypos:=1;
    repeat
      main_menu(ypos);
      case ypos of
        1 : sub_menu1;
        2 : sub_menu2;
        3 : sub_menu3;
        4 : sub_menu4;
        5 : begin
            chk_Cal_Over;
            if OverAll=true then sub_menu5;
          end;
        6 : begin
            data_file;
```

```
Auto_cal;  
initdata;  
end;  
end;  
until ypos=7;  
eoj;  
error:=false;  
until upcase(ans)='Y';  
close(data);  
textmode(co80);  
clrscr;  
end.
```




ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

VITA



Mr. Wirote Khomphatraporn graduated high school from Darun Pittaya school in 1984 and received Bachelor Degree in Chemical Engineering from the Department of Chemical Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Thonburi in 1991. After then he subsequently studied for a requirement of the Master's Degree in Chemical Engineering at the Department of Chemical Engineering, Faculty of Engineering, Chulalongkorn University from 1992 till 1995.

He also has an experience in working as Project Engineer at the Thai Urethane Group. in 1990 till 1995 and Neotec & Trading Co., Ltd. uptill the present.



ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย