บรรณานุกรม

ภาษาไทย

เกรียงศักดิ์ อุดมสินโรจน์. วิศวกรรมการกำจัดน้ำเสีย เล่มที่ 2. กรุงเทพมหานคร: มิตรนราการพิมพ์, 2533.

ธีระ เกรอต. บ่อคงตัว. กรุงเทพมหานคร: ภาควิชาวิศวกรรมสิ่งแวดล้อม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. (อัดสำเนา)

บุญเลิศ เอี่ยมทัศนา, ยืน ภู่วรวรรณ และสมนึก คีรีโต. โปรแกรมคอมพิวเตอร์ ภาษาซี. พิมพ์ครั้งที่ 4. กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น, 2531.

ไบรอัน ดับบลิว เคอร์นิกแฮน และเอนนิส เอ็ม ริตชี. ภาษาและการโปรแกรมซี. แปลโดย วิทยา วัชระวิทย์ากุล. กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น, 2534.

ผิจักขณ์ ตันเหิมหงส์. ขบวนการบำบัดน้ำทิ้งทางกายภาพและฆ่าเชื้อโรค. ในสุเทพ สิริวิทยาปกรณ์ (บรรณาธิการ). การออกแบบระบบบำบัดน้ำเสียชุมชน ครั้งที่ 2. หน้า 3/1-55. กรุงเทพมหานคร: ภาควิชาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์, 2530.

มนตรี พจนารถลาวัฒย์. การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี. กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น, 2535.

มั่นสิน ตัณฑุลเวศม์. วิศวกรรมการประปา เล่ม 2. กรุงเทพมหานคร : โรงพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, 2532.

สุรพล สายพานิช. หลักการทำงานของกระบวนการบำบัดน้ำเสียแบบใช้ออกซิเจน. เอกสารประกอบการฝึกอบรมทางวิชาการ เรื่อง การควบคุมดูแลระบบบำบัดน้ำเสียแบบชีวภาพของโรงงานประกอบกิจการอาหารและเครื่องดื่ม จัดโดย กรมโรงงานอุตสาหกรรม, 2529.

————. การออกแบบและควบคุมการทำงานของกระบวนการตะกอนเร่ง. เอกสารประกอบการฝึกอบรมทางวิชาการเรื่อง น้ำเสีย จัดโดย วิศวกรรมสถานแห่งประเทศไทย, 2531.

————. การเติมออกซิเจน. เอกสารประกอบโครงการฝึกอบรมระยะสั้น เรื่อง เทคนิค

การบำบัดทางฟิสิกส์และเคมี. จัดโดย ภาควิชาวิศวกรรมสิ่งแวดล้อม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. (ม.ป.ป)

เสริมพล รัตตสุข และไชยยุทธ กลิ่นสุคนธ์. การออกแบบระบบบำบัดน้ำเสียจากชุมชนและโรงงานอุตสาหกรรม. กรุงเทพมหานคร: สถาบันวิจัยวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย, 2527.

ออล สตีเวนส์. การพัฒนาระบบฐานข้อมูลด้วยภาษาซี. เรียบเรียงโดยอนุชิต สุเมธิวิทย์ และวีร์ตั้งมั่นภักดีพงศ์. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2534.

เฮอร์เบิร์ต ชิลด์. การประยุกต์ใช้งานภาษาซี. เรียบเรียงโดย ศิวัฒน์ ศิวะบวร, พรชัย จักรธำรงค์ และจิรศักดิ์ ชัยวิริยะกุล. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2535.

ภาษาอังกฤษ

Borland. Turbo C 2.0 reference and user's guide. Borland international., Inc, U.S.A., 1986.

Kriengsak Udomsinrot. Wastewater engineering design. Bangkok : Mitrnara printing, 1989.

Mara, D. Sewage treatment in hot climates. John Wiley & Sons, Ltd., 1982.

McCarty, P.L. Anaerobic treatment of soluble wastes. in Gloyna E.F. and Eckenfelder W.W., Jr. (eds.). Advances in water quality improvement. Austin : University of Texas press, 1968.

Metcalf and Eddy, Inc. Wastewater engineering: treatment, disposal, reuse. NewYork: McGraw-Hill, 1979.

Schildt, H. Using turbo C 2nd editions. Borland-Osborne/McGraw- Hill, 1989.

WPCF. Operation and maintenance of sludge dewatering systems. manual of practice no. OM-8, 1987.

WPCF & ASCE. Wastewater treatment plant design. Lancaster press, Inc, 1977.

ภาคผนวก ก.

รายละเอียดแสดงโปรแกรมฟังก์ชันหลัก

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define PI 3.141592654
float flow, pf, p_flow, bod, ss, tkn;
float miss, mirss;
float elev[19];
float pri_sludge, as_sludge, sludge_vol, solid;
float wall, base, degree, rc, excav, pond, struc, earth;
float equip_c;
int n;
int anp, fac, mat, k;
int x1 = 3, y1 = 2, x2 = 78, y2 = 24;
char answer, out, bio, slud;
char profile_name[19][25];
char r_name[12];
char pname[28];
FILE *fp;
FILE *fl;
FILE *ft;
float trial (float w_vol, float depth, float lw, float slope, float a_width);
/*****************************************************************************/
main()
{
  int i;
  char ch;
  logo();
  input_data();
  p_flow = flow * pf;
  ins_frame();
  do {
     del_message (23);
     gotoxy (10, 23); printf ("Do you want to save result of program ");
                      printf ("to a file ? <Y/N> : ");
     out = getch();
  } while (toupper(out) != 'Y' && toupper(out) != 'N');
  if (toupper(out) == 'Y') {
     do {
        del_message (23);
        gotoxy (5, 23); printf ("Result-file name : "); scanf ("%s", r_name);
        if ((fl = fopen(r_name, "w")) == NULL) {
           gotoxy (30, 20); printf ("Error in open file '%s' ", r_name);
        }
     } while (fl == NULL);
     fprintf (fl,"\t\t\t      ============================ \n");
     fprintf (fl,"\t\t\t       Wastewater characteristics \n");
     fprintf (fl,"\t\t\t      ============================ \n\n");
     fprintf (fl,"\t\t average flow rate              =  %8.2f      cu.m/d \n", flow);
     fprintf (fl,"\t\t peak factor                    =  %8.2f \n", pf);
     fprintf (fl,"\t\t influent BOD5                  =  %8.2f      mg/l \n", bod);
     fprintf (fl,"\t\t influent SS                    =  %8.2f      mg/l \n", ss);
     fprintf (fl,"\t\t influent TKN                   =  %8.2f      mg/l \n\n\n", tkn);
     fprintf (fl,"\t\t\t\t ------------------ \n");
     fprintf (fl,"\t\t\t\t  Processes design \n");
     fprintf (fl,"\t\t\t\t ------------------ \n\n");
```

```
}
ft = fopen ("file.tmp","w");
i = 0;
do {
   if (i == 0) { strcpy(pname, "grit chamber"); }
   if (i == 1) { strcpy(pname, "equalization tank"); }
   if (i == 2) { strcpy(pname, "primary sedimentation tank"); }
   frame (x1, y1, x2, y2);
   gotoxy (28, 3); printf (" PRIMARY TREATMENT PROCESS ");
   do {
      gotoxy (18-3*i, 12); printf ("Do you want to have %s ? <Y/N> : ", pname);
      answer = getche();
   } while (toupper(answer) != 'Y' && toupper(answer) != 'N');
   if (toupper(answer) == 'N') {
      ins_frame();
      do {
         gotoxy (17, 23); printf ("Are you sure to neglect this system ? <Y/N> : ");
         ch = getch();
      } while (toupper(ch) != 'Y' && toupper(ch) != 'N');
      if (toupper(ch) == 'N')
         answer = 'Y';
      del_message(22); del_message(23);
   }
   if (toupper(answer) == 'Y') {
      switch(i) {
         case 0 : gritcham();
                  break;
         case 1 : equal();
                  break;
         case 2 : pri_sed();
      }
   }
   i++;
} while (i <= 2);
do {
   answer = menu_2nd();
   switch(answer) {
      case '1' : an_pond();
                 break;
      case '2' : fac_pond();
                 break;
      case '3' : mat_pond();
                 break;
      case '4' : al();
                 break;
      case '5' : as();
      case '6' : sec_cla();
      case '7' : bio = 'n';
   }
} while (toupper(bio) == 'Y');
answer = menu_disinfect();
switch(answer) {
   case '1' : chlorine();
              break;
   case '2' : oz_uv(1);
              break;
```

```
      case '3' : oz_uv(2);
   }
   do {
      answer = menu_sludge();
      switch(answer) {
         case '1' : an_digest();
                    slud = 'Y';
                    break;
         case '2' : dry_bed();
                    slud = 'n';
                    break;
         case '3' :
         case '4' :
         case '5' :
         case '6' : dewater();
         case '7' : slud = 'n';
      }
   } while (slud == 'Y');
   fclose(ft);
   cost();
   fclose(fl);
   system("del *.cvl");
   system("del *.eqp");
   system("del file.tmp");
   gotoxy (1, 25);
   exit(0);
}
logo()
{
   register int i;
   clrscr();
   highvideo();
   gotoxy (3, 2); cprintf ("%c", 201);
   for (i = 1; i <= 74; i++)
      cprintf ("%c", 205);
   cprintf ("%c", 187);
   for (i = 1; i <= 20; i++) {
      gotoxy (3, 2+i); cprintf ("%c", 186);
      gotoxy (78, 2+i); cprintf ("%c", 186);
   }
   gotoxy (3, 23); cprintf ("%c", 200);
   for (i = 1; i <= 74; i++)
      cprintf ("%c", 205);
   cprintf ("%c", 188);
   gotoxy (9, 17); cprintf ("Adviser    :  Prof. Dr. Thongchai  Panswad");
   gotoxy (9, 18); cprintf ("              Department of Environmental Engineering");
   gotoxy (9, 19); cprintf ("              Chulalongkorn University");
   normvideo();
   gotoxy (9, 21); printf ("Programmer :  Mr. Somsak  Surachaipitak");
   gotoxy (27, 4); printf ("W E L C O M E  T O");
   gotoxy (15, 6); printf ("P R E L I M I N A R Y  C O S T  E S T I M A T E");
   gotoxy (39, 8); printf ("O F");
   gotoxy (9, 10); printf ("B I O L O G I C A L  W A S T E W A T E R  ");
                   printf ("T R E A T M E N T");
   gotoxy (29, 12); printf ("P R O G R A M M I N G");
   gotoxy (3, 25); printf ("Press any key to continue "); getch();
```

```
}
input_data()
{
  int j = 1;
  char ch, sure;
  do {
     frame (x1, y1, x2, y2);
     ins_frame();
     gotoxy (33, 3); printf (" Input raw data ");
     if (j == 1) {
        gotoxy (14, 23); printf ("press [E] to exit,  others key to input raw data :");
        ch = getch();
        if (toupper(ch) == 'E') {
           del_message(23);
           gotoxy (25, 23); printf ("Are you sure to exit ? <Y/N> : ");
           sure = getch();
           if (toupper(sure) == 'Y')
              exit(1);
        }
     }
     gotoxy (18, 8); printf ("[1] Average flow rate   =              cu.m/d");
     gotoxy (18, 10); printf ("[2] Peak factor         =               ");
     gotoxy (18, 12); printf ("[3] influent BOD5       =               mg/l");
     gotoxy (18, 14); printf ("[4] influent SS         =               mg/l");
     gotoxy (18, 16); printf ("[5] influent TKN        =               mg/l");
     gotoxy (5, 23); printf ("input ONLY numeric number without comma. EXAM :- 1, ");
                     printf ("12345, 12345.67 etc.");
     gotoxy (45, 8); scanf ("%f", &flow);
     gotoxy (45, 10); scanf ("%f", &pf);
     gotoxy (45, 12); scanf ("%f", &bod);
     gotoxy (45, 14); scanf ("%f", &ss);
     gotoxy (45, 16); scanf ("%f", &tkn);
     del_message(23);
     gotoxy (13, 23); printf ("press [R] to re-input data,  others key to continue :");
     ch = getch();
     j++;
  } while (toupper(ch) == 'R');
}
save_file()
{
  char ans, file_name[12];
  do {
     del_message (23);
     gotoxy (5, 23); printf ("Do you want to save new parameters' ");
                     printf ("value to another one file ? <Y/N> : ");
     ans = getche();
  } while (toupper(ans) != 'Y' && toupper(ans) != 'N');
  if (toupper(ans) == 'Y') {
     do {
        del_message (23);
        gotoxy (5, 23); printf ("File name : "); scanf ("%s", file_name);
        if ((fp = fopen(file_name, "w")) == NULL) {
           del_message (20);
           gotoxy (30, 20); printf ("Error in open file '%s'\n", file_name);
        }
     } while (fp == NULL);
```

```
  }
  return (ans);
}
read_txt (filename)
char filename[];
{
  int chr, chrter;
  FILE *fl1;
  if ((fl1 = fopen(filename, "r")) == NULL) {
     del_message (23);
     gotoxy (27, 23); printf ("Error in read file '%s'\n", filename);
     exit(1);
  }
  clrscr();
  while ( !feof(fl1) ) {
     chrter = getc(fl1);
     if ( ferror(fl1) ) {
        del_message (23);
        gotoxy (28, 23); printf ("File '%s' error", filename);
        exit(1);
     }
     putchar(chrter);
  }
  gotoxy (3, 25); printf ("Press any key to continue ");
  getch();
  fclose (fl1);
}
read_default (col, sys_name, dat_name)
int col;
char sys_name[], dat_name[];
{
  char sure, f_name[12];
  frame (x1, y1, x2, y2);
  ins_frame();
  gotoxy (col, 3); printf ("%s", sys_name);
  do {
     do {
        gotoxy (15, 23); printf ("press ENTER to use default data_file, [E] to exit.");
        del_message (13);
        gotoxy (25, 13); printf ("Data-file name  : ");
        gets (f_name);
        if (n == 0)
           gets (f_name);
        n++;
        sure = 'Y';
        if ((strcmp(f_name, "E") && strcmp(f_name, "e")) == 0) {
           del_message(23);
           gotoxy (25, 23); printf ("Are you sure to exit ? <Y/N> : ");
           sure = getch();
           printf ("%c", sure);
           if (toupper(sure) == 'Y') {
              fclose(fl);
              exit(1);
           }
           del_message(18); del_message(23);
        }
```

```
    } while (toupper(sure) != 'Y');
    if (strcmp(f_name, "\0") == 0)
       strcpy (f_name, dat_name);
    if ((fp = fopen(f_name, "r")) == NULL) {
       del_message (18);
       gotoxy (29, 18); printf ("File '%s' not founded\n", f_name);
    }
  } while (fp == NULL);
}
menu_2nd()
{
  frame (x1, y1, x2, y2);
  ins_frame();
  gotoxy (28, 3); printf (" 2nd TREATMENT PROCESSES ");
  gotoxy (28, 7); printf ("[1]  Anaerobic pond ");
  gotoxy (28, 9); printf ("[2]  Facutative pond ");
  gotoxy (28, 11); printf ("[3]  Maturation pond ");
  gotoxy (28, 13); printf ("[4]  Aerated lagoon ");
  gotoxy (28, 15); printf ("[5]  Activated sludge ");
  gotoxy (28, 17); printf ("[6]  Secondary clarifier ");
  gotoxy (28, 19); printf ("[7]  EXIT this system ");
  do {
     gotoxy (26, 23); printf ("Select the process  [1-7]  : ");
     answer = getch();
  } while (answer < '1' || answer > '7');
  return (answer);
}
menu_disinfect()
{
  frame (x1, y1, x2, y2);
  ins_frame();
  gotoxy (28, 3); printf (" DISINFECTION PROCESSES ");
  gotoxy (30, 8); printf ("[1]  Chlorination ");
  gotoxy (30, 10); printf ("[2]  Ozonation ");
  gotoxy (30, 12); printf ("[3]  UV ");
  gotoxy (30, 14); printf ("[4]  Neglect this system ");
  do {
     gotoxy (26, 23); printf ("Select the process  [1-4]  : ");
     answer = getch();
  } while (answer < '1' || answer > '4');
  return (answer);
}
menu_sludge()
{
  frame (x1, y1, x2, y2);
  ins_frame();
  gotoxy (27, 3); printf (" SLUDGE DISPOSAL PROCESSES ");
  gotoxy (25, 7); printf ("[1]  Anaerobic sludge-digestion ");
  gotoxy (25, 9); printf ("[2]  Drying bed ");
  gotoxy (25, 11); printf ("[3]  Filter press ");
  gotoxy (25, 13); printf ("[4]  Centifuge ");
  gotoxy (25, 15); printf ("[5]  Vacuum filter ");
  gotoxy (25, 17); printf ("[6]  Belt press ");
  gotoxy (25, 19); printf ("[7]  EXIT this system ");
  do {
     gotoxy (26, 23); printf ("Select the process  [1-7]  : ");
```

```
    answer = getch();
  } while (answer < '1' || answer > '7');
  return (answer);
}
frame (x1, y1, x2, y2)
int x1, y1, x2, y2;
{
  register int i;
  clrscr();
  gotoxy (x1, y1); printf ("%c", 201);
  for (i = 1; i <= x2 - x1 - 1; i++)
     printf ("%c", 205);
  printf ("%c", 187);
  gotoxy (x1, y1+1); printf ("%c", 186);
  for (i = 1; i <= x2-41; i++) {
     gotoxy (x1+i, y1+1); printf ("%c", 177);
     gotoxy (x2-i, y1+1); printf ("%c", 176);
  }
  gotoxy (x2, y1+1); printf ("%c", 186);
  gotoxy (x1, y1+2); printf ("%c", 204);
  for (i = 1; i <= x2 - x1 - 1; i++)
     printf ("%c", 205);
  printf ("%c", 185);
  for (i = 3; i <= y2 - y1; i++) {
     gotoxy (x1, y1+i); printf ("%c", 186);
     gotoxy (x2, y1+i); printf ("%c", 186);
  }
  gotoxy (x1, y2); printf ("%c", 200);
  for (i = 1; i <= x2 - x1 - 1; i++)
     printf ("%c", 205);
  printf ("%c", 188);
}
ins_frame()
{
  int i;
  gotoxy (x1, y2-2); printf ("%c", 204);
  for (i = 1; i <= x2 - x1 - 1; i++)
     printf ("%c", 205);
  printf ("%c", 185);
}
del_message (row)
int row;
{
  gotoxy (1, row); delline(); insline();
  gotoxy (x1, row); printf ("%c", 186);
  gotoxy (x2, row); printf ("%c", 186);
}
float trial(float w_vol, float depth, float lw, float slope, float a_width)
{
  float w_width, b_width, b_length;
  float vol=0, b_area, w_area;
  for (w_width = a_width; vol < w_vol; ++w_width) {
     b_width = w_width - 2*slope*depth;
     if (b_width <= 0)
        continue;
     b_length = lw*w_width - 2*slope*depth;
```

```
    b_area = b_width * b_length;
    w_area = lw * pow(w_width,2);
    vol = (w_area + b_area + sqrt(w_area * b_area)) * depth / 3;
  }
  w_width -= 1;
  if (vol > w_vol) {
    vol = 0;
    for (w_width = w_width-1; vol < w_vol; w_width += 0.005) {
      b_width = w_width - 2*slope*depth;
      if (b_width <= 0)
        return(0);
      b_length = lw*w_width - 2*slope*depth;
      b_area = b_width * b_length;
      w_area = lw * pow(w_width,2);
      vol = (w_area + b_area + sqrt(w_area * b_area)) * depth / 3;
    }
    w_width -= 0.005;
  }
  return(w_width);
}
```

ภาคผนวก ข.

รายละเอียดแสดงโปรแกรมฟังก์ชันย่อยต่าง ๆ

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define PI 3.141592654
extern float flow, pf, p_flow, bod, ss, tkn;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char pname[28];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *f1;
/*****************************************************************************/
gritcham()
{
  int col, count;
  float val;
  float hrt, vh, vs, no, percent, free;
  float t_flow, vol, t_vol, area, dep, depth, width, length;
  char ans, ch;
  char s_name[22], d_name[10];
  FILE *fi;
  col = 31;
  strcpy(s_name, " Grit chamber system ");
  strcpy(d_name, "d_grit.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f %f", &hrt, &vh, &vs, &no, &percent, &free);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (23, 3); printf (" Grit chamber's design parameters ");
        gotoxy (17, 7); printf ("[1] Hydraulic retention time  =  %6.2f   sec.", hrt);
        gotoxy (17, 9); printf ("[2] Horizontal velocity       =  %6.2f   m/sec", vh);
        gotoxy (17, 11); printf ("[3] Settling velocity         =  %6.2f   m/min", vs);
        gotoxy (17, 13); printf ("[4] No. of chamber            =  %3.0f ", no);
        gotoxy (17, 15); printf ("[5] Percent of peak flow      =  %3.0f ", percent);
        gotoxy (17, 16); printf ("    for each chamber");
        gotoxy (17, 18); printf ("[6] Free board                =  %6.2f   m.", free);
        gotoxy (5, 23); printf ("ENTER = accept parameters ,  [1-6] for change ");
                        printf ("parameter ,  H = HELP : ");
        ch = getche();
        if (toupper(ch) == 'H')
           read_txt ("c_grit.txt");
        if (ch >= '1' && ch <= '6') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           count++;
           n = 0;
           switch (ch) {
```

```
            case '1' : hrt = val;
                       break;
            case '2' : vh = val;
                       break;
            case '3' : vs = val;
                       break;
            case '4' : no = val;
                       break;
            case '5' : percent = val;
                       break;
            case '6' : free = val;
          }
       }
    } while (ch != '\r');
    if (count != 0) {
       ans = save_file();
       if (toupper(ans) == 'Y') {
          fprintf (fp, "%f %f %f %f %f %f", hrt, vh, vs, no, percent, free);
          fclose (fp);
       }
    }
    t_flow = p_flow * percent / 100;
    vol = t_flow * hrt / 86400;
    area = t_flow / (vh * 86400);
    dep = hrt * vs / 60;
    depth = dep + free;
    width = area / dep;
    length = vol / area;
    t_vol = width * length * depth;
    count = 0;
    do {
       frame (x1, y1, x2, y2);
       ins_frame();
       gotoxy (21, 3); printf (" Result of GRIT CHAMBER process design ");
       gotoxy (12, 6); printf ("Each chamber : ");
       gotoxy (22, 8); printf ("Water volume     =  %6.2f      cu.m ", vol);
       gotoxy (22, 10); printf ("Tank volume      =  %6.2f      cu.m ", t_vol);
       gotoxy (22, 12); printf ("Dimensions : ");
       gotoxy (25, 14); printf ("Width          =  %6.2f      m. ", width);
       gotoxy (25, 16); printf ("Depth          =  %6.2f      m. ", depth);
       gotoxy (25, 18); printf ("Length         =  %6.2f      m. ", length);
       gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                        printf ("change parameters : ");
       ch = getche();
    } while (ch != '\r' && toupper(ch) != 'C');
} while (ch != '\r');
if (toupper(out) == 'Y') {
    fprintf (fl,"\t GRIT CHAMBER SYSTEM : \n");
    fprintf (fl,"\t grit chamber's design parameters \n");
    fprintf (fl,"\t\t hydraulic retention time          =  %7.2f    sec. \n", hrt);
    fprintf (fl,"\t\t horizontal velocity               =  %7.2f    m/sec \n", vh);
    fprintf (fl,"\t\t settling velocity                 =  %7.2f    m/min \n", vs);
    fprintf (fl,"\t\t no. of chamber                    =  %4.0f \n", no);
    fprintf (fl,"\t\t percent of peak flow -            =  %4.0f \n", percent);
    fprintf (fl,"\t\t  for each chamber \n");
    fprintf (fl,"\t\t free board                        =  %7.2f    m. \n", free);
```

```
    fprintf (fl,"\t result of grit chamber process design \n");
    fprintf (fl,"\t\t water volume                    =  %7.2f    cu.m \n", vol);
    fprintf (fl,"\t\t tank volume                     =  %7.2f    cu.m \n", t_vol);
    fprintf (fl,"\t\t dimensions : \n");
    fprintf (fl,"\t\t    width                       =  %7.2f    m. \n", width);
    fprintf (fl,"\t\t    depth                       =  %7.2f    m. \n", depth);
    fprintf (fl,"\t\t    length                      =  %7.2f    m. \n\n", length);
  }
  strcpy (profile_name[1],"Grit chamber");
  putc ('g',ft);
  fi = fopen ("grit.cvl","w");
  fprintf (fi,"%f %f %f %f %f %f %f %f", 0.0, 0.0, width, length, depth, dep, 0.0, no);
  fclose (fi);
  fi = fopen ("grit.eqp","w");
  fprintf (fi,"%f %f %f %f",width, length, depth, no);
  fclose (fi);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
extern float flow, pf, p_flow, bod, ss, tkn;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char pname[28];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
extern float trial(float w_vol, float depth, float lw, float slope, float a_width);
/*****************************************************************************/
equal()
{
  int col, count;
  float val;
  float hrt, dep, lw, slope, free;
  float w_vol, p_vol, a_area, w_area, s_area, depth, a_width;
  float b_width, w_width, s_width, b_length, w_length, s_length;
  char ans, ch;
  char s_name[28], d_name[12];
  FILE *f2;
  col = 27;
  strcpy(s_name, " Equalization tank system ");
  strcpy(d_name, "d_equal.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f", &hrt, &dep, &lw, &slope, &free);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (21, 3); printf (" Equalization tank's design parameters ");
        gotoxy (19, 9); printf ("[1] Detention time          = %6.2f   hr.", hrt);
        gotoxy (19, 11); printf ("[2] Water depth            = %6.2f   m.", dep);
        gotoxy (19, 13); printf ("[3] Length/width (approx.)  = %6.2f ", lw);
        gotoxy (19, 15); printf ("[4] Embankment slope       1  : %6.2f ", slope);
        gotoxy (19, 17); printf ("[5] Free board             = %6.2f   m.", free);
        gotoxy (10, 23); printf ("ENTER = accept parameters, [1-5] change parameter ");
        ch = getche();
        if (ch >= '1' && ch <= '5') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           count++;
           n = 0;
           switch (ch) {
              case '1' : hrt = val;
                         break;
              case '2' : dep = val;
                         break;
```

```
            case '3' : lw = val;
                       break;
            case '4' : slope = val;
                       break;
            case '5' : free = val;
          }
       }
    } while (ch != '\r');
    if (count != 0) {
       ans = save_file();
       if (toupper(ans) == 'Y') {
          fprintf (fp, "%f %f %f %f %f", hrt, dep, lw, slope, free);
          fclose (fp);
       }
    }
    w_vol = flow * hrt / 24;
    a_area = w_vol / dep;
    a_width = sqrt (a_area / lw);
    w_width = trial (w_vol, dep, lw, slope, a_width);
    if (w_width > 0) {
       w_length = lw * w_width;
       w_area = w_width * w_length;
       s_width = w_width + 2*slope*free;
       s_length = w_length + 2*slope*free;
       s_area = s_width * s_length;
       p_vol = w_vol + (s_area + w_area + sqrt (s_area*w_area)) * free / 3;
       b_width = w_width - 2*slope*dep;
       b_length = w_length - 2*slope*dep;
       depth = dep + free;
    }
    count = 0;
    do {
       frame (x1, y1, x2, y2);
       ins_frame();
       gotoxy (19, 3); printf (" Result of EQUALIZATION TANK process design ");
       gotoxy (20, 8); printf ("Water volume          =  %8.2f      cu.m ", w_vol);
       if (w_width > 0) {
          gotoxy (20, 10); printf ("Tank volume           =  %8.2f      cu.m ", p_vol);
          gotoxy (20, 12); printf ("Dimensions : ");
          gotoxy (23, 13); printf ("Surface width      =  %8.2f      m. ", s_width);
          gotoxy (23, 14); printf ("Surface length     =  %8.2f      m. ", s_length);
          gotoxy (23, 15); printf ("Bottom width       =  %8.2f      m. ", b_width);
          gotoxy (23, 16); printf ("Bottom length      =  %8.2f      m. ", b_length);
          gotoxy (23, 17); printf ("Depth              =  %8.2f      m. ", depth);
       }
       else {
          gotoxy (10, 18); printf ("Water depth, Length/Width and Embankment slope ");
                           printf ("are mismatch.");
          gotoxy (22, 19); printf ("One or more of them must bechanged.");
       }
       gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                        printf ("change parameters : ");
       ch = getche();
    } while (ch != '\r' && toupper(ch) != 'C');
} while (ch != '\r');
if (toupper(out) == 'Y') {
```

```
      fprintf (f1,"\t EQUALIZATION TANK SYSTEM : \n");
      fprintf (f1,"\t equalization tank's design parameters \n");
      fprintf (f1,"\t\t detention time                    =  %8.2f      hr.\n", hrt);
      fprintf (f1,"\t\t water depth                       =  %8.2f      m. \n", dep);
      fprintf (f1,"\t\t length/width (approx.)            =  %8.2f \n", lw);
      fprintf (f1,"\t\t embankment slope              1  :  %8.2f \n", slope);
      fprintf (f1,"\t\t free board                       =  %8.2f      m.\n", free);
      fprintf (f1,"\t result of equalization tank process design \n");
      fprintf (f1,"\t\t water volume                     =  %8.2f     cu.m \n", w_vol);
      if (w_width > 0) {
         fprintf (f1,"\t\t pond volume                  =  %8.2f      cu.m \n", p_vol);
         fprintf (f1,"\t\t dimensions : \n");
         fprintf (f1,"\t\t    surface width             =  %8.2f      m. \n", s_width);
         fprintf (f1,"\t\t    surface length            =  %8.2f      m. \n", s_length);
         fprintf (f1,"\t\t    bottom width              =  %8.2f      m. \n", b_width);
         fprintf (f1,"\t\t    bottom length             =  %8.2f      m. \n", b_length);
         fprintf (f1,"\t\t    depth                     =  %8.2f      m. \n\n", depth);
      }
   }
   strcpy (profile_name[2],"Equalization tank");
   putc ('e',ft);
   f2 = fopen ("equal.cv1","w");
   fprintf (f2,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
   fprintf (f2,"%f %f", 0.0, 1.0);
   fclose (f2);
   f2 = fopen ("equal.eqp","w");
   fprintf (f2,"%f %f %f %f", s_width, s_length, depth, w_vol);
   fclose (f2);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define PI 3.141592654
extern float flow, pf, p_flow, bod, ss, tkn;
extern float pri_sludge;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char pname[28];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
/*****************************************************************************/
pri_sed()
{
  int col, count, i, j;
  float val;
  float ofr, dep, lw, free, no, percent, bod_r, ss_r;
  float t_flow, area, volume, t_vol, hrt, depth, ofr_p, hrt_p, bod1, ss1;
  double dia, width, length;
  char ans, ch, che, cha;
  char s_name[36], d_name[10];
  FILE *fi;
  do {
     frame (x1, y1, x2, y2);
     ins_frame();
     gotoxy (23, 3); printf (" Primary sedimentation tank system ");
     gotoxy (30, 11); printf ("[1]  Circular tank");
     gotoxy (30, 13); printf ("[2]  Rectangular tank");
     do {
        gotoxy (30, 23); printf ("Select type  [1,2] : ");
        che = getch();
     } while (che < '1' || che > '2');
     col = 22;
     strcpy(s_name, " Primary sedimentation tank system ");
     strcpy(d_name, "d_pri.dat");
     read_default (col, s_name, d_name);
     fscanf (fp,"%f %f %f %f %f %f ", &ofr, &dep, &lw, &free, &no, &percent);
     fscanf (fp,"%f %f", &bod_r, &ss_r);
     fclose (fp);
     count = 0;
     i = 0;
     do {
        do {
           frame (x1, y1, x2, y2);
           ins_frame();
           gotoxy (16, 3); printf (" Primary sedimentation tank's design parameters ");
           gotoxy (16, 8); printf ("[1]  Overflow rate (ave. Q)    =  %6.2f    ", ofr);
                           printf ("cu.m/sq.m-d ");
           gotoxy (16, 9); printf ("[2]  Water depth               =  %6.2f    m.", dep);
           if (che == '2') {
              i = 1;
```

```
   gotoxy (16, 10); printf ("[3]  Length/width ratio       =  %6.2f ", lw);
}
gotoxy (16, 10+i); printf ("[%d]  free board              =  ", 3+i);
                   printf ("%6.2f   m.", free);
gotoxy (16, 11+i); printf ("[%d]  No. of tank             =  ", 4+i);
                   printf ("%3.0f ", no);
gotoxy (16, 12+i); printf ("[%d]  Percent of flow rate -  =  ", 5+i);
                   printf ("%3.0f ", percent);
gotoxy (16, 13+i); printf ("      for each tank");
gotoxy (16, 14+i); printf ("[%d]  Percent of BOD removal  =  ", 6+i);
                   printf ("%3.0f ", bod_r);
gotoxy (16, 15+i); printf ("[%d]  Percent of SS removal   =  ", 7+i);
                   printf ("%3.0f ", ss_r);
gotoxy (5, 23); printf ("ENTER = accept parameters,  [1-%d] for ", 7+i);
                printf ("change parameter,  [H] = HELP : ");
ch = getche();
if (toupper(ch) == 'H') {
   read_txt ("c_pri_1.txt");
   read_txt ("c_pri_2.txt");
   read_txt ("c_pri_3.txt");
}
if (che == '1') {
   if (ch >= '1' && ch <= '7') {
      del_message (23);
      gotoxy (15, 23); printf ("Input new value of parameter no.[%c] ", ch);
      scanf ("%f", &val);
      count++;
      n = 0;
      switch (ch) {
         case '1' : ofr = val;  break;
         case '2' : dep = val;  break;
         case '3' : free = val; break;
         case '4' : no = val;  break;
         case '5' : percent = val;  break;
         case '6' : bod_r = val;  break;
         case '7' : ss_r = val;
      }
   }
}
else {
   if (ch >= '1' && ch <= '8') {
      del_message (23);
      gotoxy (15, 23); printf ("Input new value of parameter no.[%c] ", ch);
      scanf ("%f", &val);
      count++;
      n = 0;
      switch (ch) {
         case '1' : ofr = val;  break;
         case '2' : dep = val;  break;
         case '3' : lw = val;  break;
         case '4' : free = val;  break;
         case '5' : no = val;  break;
         case '6' : percent = val;  break;
         case '7' : bod_r = val;  break;
         case '8' : ss_r = val;
      }
```

```
      }
   }
} while (ch != '\r');
if (count != 0) {
   ans = save_file();
   if (toupper(ans) == 'Y') {
      fprintf (fp, "%f %f %f %f %f", ofr, dep, lw, free, no);
      fprintf (fp, "%f %f %f", percent, bod_r, ss_r);
      fclose (fp);
   }
}
t_flow = flow * percent / 100;
area = t_flow / ofr;
volume = area * dep;
hrt = volume / t_flow * 24;
depth = dep + free;
t_vol = area * depth;
dia = width = length = 0;
if (che == '1')
   dia = sqrt (area * 4 / PI);
else {
   width = sqrt (area / lw);
   length = width * lw;
}
ofr_p = p_flow / (no * area);
hrt_p = volume * no * 24 / p_flow;
bod1 = bod * (1 - (bod_r / 100));
ss1 = ss * (1 - (ss_r /100));
pri_sludge = (ss - ss1) / 1000 * flow;
count = 0;
do {
   frame (x1, y1, x2, y2);
   ins_frame();
   gotoxy (14, 3); printf (" Result of PRIMARY SEDIMENTATION TANK process ");
   gotoxy (20, 7); printf ("Detention time      = %7.2f      hr. ", hrt);
   gotoxy (20, 8); printf ("Water volume        = %7.2f      cu.m ", volume);
   gotoxy (20, 9); printf ("Tank volume         = %7.2f      cu.m ", t_vol);
   gotoxy (20, 10); printf ("Dimensions : ");
   if (che == '1') {
      gotoxy (23, 11); printf ("Diameter         = %7.2f      m. ", dia);
      gotoxy (23, 12); printf ("Depth            = %7.2f      m. ", depth);
      j = 14;
   }
   else {
      gotoxy (23, 11); printf ("Width            = %7.2f      m. ", width);
      gotoxy (23, 12); printf ("Depth            = %7.2f      m. ", depth);
      gotoxy (23, 13); printf ("Length           = %7.2f      m. ", length);
      j = 15;
   }
   gotoxy (14, j); printf ("Overflow rate (peak Q)    = %7.2f      ", ofr_p);
                   printf ("cu.m/sq.m-d ");
   gotoxy (14, j+1); printf ("Detention time (peak Q)   = %7.2f hr.", hrt_p);
   gotoxy (14, j+3); printf ("Effluent BOD5             = %7.2f mg/l ", bod1);
   gotoxy (14, j+4); printf ("Effluent SS               = %7.2f mg/l ", ss1);
   gotoxy (5, 23); printf ("Enter = accept result,  [T,P] = back to change ");
                   printf ("tank Type, Parameters : ");
```

```
            cha = getche();
         } while (cha != '\r' && toupper(cha) != 'T' && toupper(cha) != 'P');
      } while (toupper(cha) == 'P');
   } while (toupper(cha) == 'T');
   bod = bod1;
   ss = ss1;
   if (toupper(out) == 'Y') {
      fprintf (fl,"\t PRIMARY SEDIMENTATION TANK SYSTEM :\n");
      fprintf (fl,"\t primary sedimentation tank's design parameters \n");
      fprintf (fl,"\t\t Overflow rate (ave. Q)      = %7.2f       cu.m/sq.m-d \n", ofr);
      fprintf (fl,"\t\t Water depth                 = %7.2f       m. \n", dep);
      if (che == '2')
         fprintf (fl,"\t\t Length/width ratio         = %7.2f \n", lw);
      fprintf (fl,"\t\t free board                  = %7.2f       m. \n", free);
      fprintf (fl,"\t\t No. of tank                 = %4.0f \n", no);
      fprintf (fl,"\t\t Percent of flow rate -      = %4.0f \n", percent);
      fprintf (fl,"\t\t  for each tank \n");
      fprintf (fl,"\t\t Percent of BOD removal      = %4.0f \n", bod_r);
      fprintf (fl,"\t\t Percent of SS removal       = %4.0f \n", ss_r);
      fprintf (fl,"\t result of primary sedimentation tank process design \n");
      fprintf (fl,"\t\t detention time              = %7.2f       hr. \n", hrt);
      fprintf (fl,"\t\t water volume               = %7.2f       cu.m \n", volume);
      fprintf (fl,"\t\t tank volume                = %7.2f       cu.m \n", t_vol);
      fprintf (fl,"\t\t dimensions : \n");
      if (che == '1') {
         fprintf (fl,"\t\t    diameter                 = %7.2f       m. \n", dia);
         fprintf (fl,"\t\t    depth                    = %7.2f       m. \n", depth);
      }
      else {
         fprintf (fl,"\t\t    width                    = %7.2f       m. \n", width);
         fprintf (fl,"\t\t    length                   = %7.2f       m. \n", length);
         fprintf (fl,"\t\t    depth                    = %7.2f       m. \n", depth);
      }
      fprintf (fl,"\t\t overflow rate (peak Q)     = %7.2f       cu.m/sq.m-d\n",ofr_p);
      fprintf (fl,"\t\t detention time (peak Q)    = %7.2f       hr. \n", hrt_p);
      fprintf (fl,"\t\t effluent BOD5              = %7.2f       mg/l \n", bod);
      fprintf (fl,"\t\t effluent SS                = %7.2f       mg/l \n\n", ss);
   }
   strcpy (profile_name[3],"Primary sedimentation");
   putc ('p',ft);
   fi = fopen ("pri_sed.cvl","w");
   fprintf (fi,"%f %f %f %f %f %f %f %f", 0.0, 0.0, width, length, depth, dep, dia, no);
   fclose (fi);
   fi = fopen ("pri.eqp","w");
   fprintf (fi,"%f %f %f %f %f", width, length, dia, depth, no);
   fclose (fi);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define PI 3.141592654
extern float flow, pf, p_flow, bod, ss, tkn;
extern int n, anp, x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
extern float trial(float w_vol, float depth, float lw, float slope, float a_width);
/*****************************************************************************/
an_pond()
{
  int col, count;
  float val;
  float hrt, dep, lw, slope, free, bod_r;
  float w_vol, p_vol, a_area, w_area, s_area, depth, a_width;
  float b_width, w_width, s_width, b_length, w_length, s_length, bod1;
  char ans, ch;
  char s_name[24], d_name[11];
  FILE *fi;
  col = 26;
  strcpy(s_name, " Anaerobic pond system ");
  strcpy(d_name, "d_an_p.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f %f", &hrt, &dep, &lw, &slope, &free, &bod_r);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (22, 3); printf (" Anaerobic pond's design parameters ");
        gotoxy (17, 8); printf ("[1] Retention time            =  %6.2f   day.", hrt);
        gotoxy (17, 10); printf ("[2] Water depth               =  %6.2f   m.", dep);
        gotoxy (17, 12); printf ("[3] Length/width (approx.)    =  %6.2f ", lw);
        gotoxy (17, 14); printf ("[4] Embankment slope       1  :  %6.2f ", slope);
        gotoxy (17, 16); printf ("[5] Free board                =  %6.2f   m.", free);
        gotoxy (17, 18); printf ("[6] Percent of BOD removal    =  %3.0f ", bod_r);
        gotoxy (5, 23); printf ("ENTER = accept parameters,  [1-6] for change ");
                        printf ("parameter,  [H] = HELP : ");
        ch = getche();
        if (toupper(ch) == 'H')
           read_txt ("c_an_p.txt");
        if (ch >= '1' && ch <= '6') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           count++;
           n = 0;
           switch (ch) {
              case '1' : hrt = val; break;
```

```
            case '2' : dep = val; break;
            case '3' : lw = val;  break;
            case '4' : slope = val; break;
            case '5' : free = val; break;
            case '6' : bod_r = val;
         }
      }
   } while (ch != '\r');
   if (count != 0) {
      ans = save_file();
      if (toupper(ans) == 'Y') {
         fprintf (fp, "%f %f %f %f %f %f", hrt, dep, lw, slope, free, bod_r);
         fclose (fp);
      }
   }
   w_vol = flow * hrt;
   a_area = w_vol / dep;
   a_width = sqrt (a_area / lw);
   w_width = trial(w_vol, dep, lw, slope, a_width);
   if (w_width > 0) {
      w_length = lw * w_width;
      w_area = w_width * w_length;
      s_width = w_width + 2*slope*free;
      s_length = w_length + 2*slope*free;
      s_area = s_width * s_length;
      p_vol = w_vol + (s_area + w_area + sqrt(s_area*w_area))*free/3;
      b_width = w_width - 2*slope*dep;
      b_length = w_length - 2*slope*dep;
      depth = dep + free;
   }
   bod1 = bod * (1 - (bod_r / 100));
   count = 0;
   do {
      frame (x1, y1, x2, y2);
      ins_frame();
      gotoxy (20, 3); printf (" Result of ANAEROBIC POND process design ");
      gotoxy (20, 7); printf ("Effluent BOD5          =  %8.2f      mg/l ", bod1);
      gotoxy (20, 9); printf ("Water volume           =  %8.2f      cu.m ", w_vol);
      if (w_width > 0) {
         gotoxy (20, 11); printf ("Pond volume         =  %8.2f      cu.m ", p_vol);
         gotoxy (20, 13); printf ("Dimensions : ");
         gotoxy (23, 14); printf ("Surface width       =  %8.2f      m. ", s_width);
         gotoxy (23, 15); printf ("Surface length      =  %8.2f      m. ", s_length);
         gotoxy (23, 16); printf ("Bottom width        =  %8.2f      m. ", b_width);
         gotoxy (23, 17); printf ("Bottom length       =  %8.2f      m. ", b_length);
         gotoxy (23, 18); printf ("Depth               =  %8.2f      m. ", depth);
      }
      else {
         gotoxy (10, 18); printf ("Water depth, Length/Width and Embankment slope ");
                          printf ("are mismatch.");
         gotoxy (22, 19); printf ("One or more of them must be changed.");
      }
      gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                       printf ("change parameters : ");
      ch = getche();
   } while (ch != '\r' && toupper(ch) != 'C');
```

```
} while (ch != '\r');
bod = bod1;
anp++;
do {
   del_message(23);
   gotoxy (6, 23); printf ("Do you want to have others 2nd treatment process ");
                   printf ("following ? <Y/N> : ");
   bio = getch();
} while (toupper(bio) != 'Y' && toupper(bio) != 'N');
if (toupper(out) == 'Y') {
   fprintf (fl,"\t ANAEROBIC POND SYSTEM :\n");
   fprintf (fl,"\t anaerobic pond's design parameters \n");
   fprintf (fl,"\t\t retention time                          = %8.2f      day.\n", hrt);
   fprintf (fl,"\t\t water depth                             = %8.2f      m. \n", dep);
   fprintf (fl,"\t\t length/width (approx.)                  = %8.2f \n", lw);
   fprintf (fl,"\t\t embankment slope                  1  :  %8.2f \n", slope);
   fprintf (fl,"\t\t free board                             = %8.2f      m.\n", free);
   fprintf (fl,"\t\t percent of BOD removal                  = %5.0f \n", bod_r);
   fprintf (fl,"\t result of anaerobic pond process design \n");
   fprintf (fl,"\t\t effluent BOD5                          = %8.2f      mg/l \n", bod);
   fprintf (fl,"\t\t water volume                           = %8.2f     cu.m \n", w_vol);
   if (w_width > 0) {
      fprintf (fl,"\t\t pond volume                       = %8.2f      cu.m \n", p_vol);
      fprintf (fl,"\t\t dimensions : \n");
      fprintf (fl,"\t\t    surface width                  = %8.2f      m. \n", s_width);
      fprintf (fl,"\t\t    surface length                 = %8.2f      m. \n", s_length);
      fprintf (fl,"\t\t    bottom width                   = %8.2f      m. \n", b_width);
      fprintf (fl,"\t\t    bottom length                  = %8.2f      m. \n", b_length);
      fprintf (fl,"\t\t    depth                          = %8.2f      m. \n\n", depth);
   }
}
if (anp == 1) {
   strcpy (profile_name[4],"Anaerobic pond");
   putc ('a',ft);
   fi = fopen ("an_p.cvl","w");
   fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
   fprintf (fi,"%f %f, 0.0, 1.0);
   fclose (fi);
}
else if (anp == 2) {
   strcpy (profile_name[5],"Anaerobic pond #2");
   putc ('i',ft);
   fi = fopen ("an_p2.cvl","w");
   fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
   fprintf (fi,"%f %f, 0.0, 1.0);
   fclose (fi);
}
else if (anp == 3) {
   strcpy (profile_name[6],"Anaerobic pond #3");
   putc ('j',ft);
   fi = fopen ("an_p3.cvl","w");
   fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
   fprintf (fi,"%f %f, 0.0, 1.0);
   fclose (fi);
}
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
extern float flow, pf, p_flow, bod, ss, tkn;
extern int n;
extern int fac;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
extern float trial(float w_vol, float depth, float lw, float slope, float a_width);
/*****************************************************************************/
fac_pond()
{
  int col, count;
  float val;
  float bod_eff, k20, temp, coef, dep, lw, slope, free;
  float w_vol, p_vol, a_area, w_area, s_area, depth, a_width;
  float b_width, w_width, s_width, b_length, w_length, s_length, t0, load;
  double kt;
  char ans, ch;
  char s_name[25], d_name[12];
  FILE *fi;
  col = 30;
  strcpy(s_name, " Facutative pond system ");
  strcpy(d_name, "d_fac_p.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f ", &bod_eff, &k20, &temp, &coef, &dep);
  fscanf (fp,"%f %f %f", &lw, &slope, &free);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (22, 3); printf (" Facutative pond's design parameters ");
        gotoxy (12, 6); printf ("[1] Effluent BOD5 requirement    = %6.2f  ",bod_eff);
                        printf ("mg/l");
        gotoxy (12, 8); printf ("[2] k constant @ 20 'c           = %6.2f  1/day",k20);
        gotoxy (12, 10); printf ("[3] Pond water temperature      =  %6.2f   'c",temp);
        gotoxy (12, 12); printf ("[4] Temperature coefficient (θ) =  %6.2f ", coef);
        gotoxy (12, 14); printf ("[5] Water depth                 =  %6.2f   m.", dep);
        gotoxy (12, 16); printf ("[6] Length/Width (approx.)      =  %6.2f ", lw);
        gotoxy (12, 18); printf ("[7] Embankment slope         1  :  %6.2f ",slope);
        gotoxy (12, 20); printf ("[8] Free board                  =  %6.2f   m.",free);
        gotoxy (5, 23); printf ("ENTER = accept parameters ,  [1-8] for change ");
                        printf ("parameter ,  H = HELP : ");
        ch = getche();
        if (toupper(ch) == 'H')
           read_txt ("c_fac_p.txt");
        if (ch >= '1' && ch <= '8') {
           del_message (23);
```

```
      gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
      scanf ("%f", &val);
      count++;
      n = 0;
      switch (ch) {
         case '1' : bod_eff = val;
                    break;
         case '2' : k20 = val;
                    break;
         case '3' : temp = val;
                    break;
         case '4' : coef = val;
                    break;
         case '5' : dep = val;
                    break;
         case '6' : lw = val;
                    break;
         case '7' : slope = val;
                    break;
         case '8' : free = val;
      }
   }
} while (ch != '\r');
if (count != 0) {
   ans = save_file();
   if (toupper(ans) == 'Y') {
      fprintf (fp, "%f %f %f %f ", bod_eff, k20, temp, coef);
      fprintf (fp, "%f %f %f %f", dep, lw, slope, free);
      fclose (fp);
   }
}
kt = k20 * pow(coef, temp - 20);
t0 = (bod / bod_eff -1) / (float)kt;
w_vol = flow * t0;
a_area = w_vol / dep;
a_width = sqrt (a_area / lw);
w_width = trial(w_vol, dep, lw, slope, a_width);
if (w_width > 0) {
   w_length = lw * w_width;
   w_area = w_width * w_length;
   s_width = w_width + 2*slope*free;
   s_length = w_length + 2*slope*free;
   s_area = s_width * s_length;
   p_vol = w_vol + (s_area + w_area + sqrt(s_area*w_area))*free/3;
   b_width = w_width - 2*slope*dep;
   b_length = w_length - 2*slope*dep;
   depth = dep + free;
}
load = bod / t0;
count = 0;
do {
   frame (x1, y1, x2, y2);
   ins_frame();
   gotoxy (20, 3); printf (" Result of FACUTATIVE POND process design ");
   gotoxy (15, 7); printf ("Detention time              =  %7.2f    day. ", t0);
   gotoxy (15, 8); printf ("Volumetric loading          =  %7.2f    ",load);
```

```
                    printf ("g BOD5/cu.m-d ");
      gotoxy (15, 10); printf ("Water volume              =  %7.0f   cu.m ", w_vol);
      if (w_width > 0) {
         gotoxy (15, 11); printf ("Pond volume            =  %7.0f   cu.m ", p_vol);
         gotoxy (15, 13); printf ("Dimensions : ");
         gotoxy (15, 14); printf ("   Surface width       =  %7.2f   m. ", s_width);
         gotoxy (15, 15); printf ("   Surface length      =  %7.2f   m. ",s_length);
         gotoxy (15, 16); printf ("   Bottom width        =  %7.2f   m. ", b_width);
         gotoxy (15, 17); printf ("   Bottom length       =  %7.2f   m. ",b_length);
         gotoxy (15, 18); printf ("   Depth               =  %7.2f   m. ", depth);
      }
      else {
         gotoxy (10, 18); printf ("Water depth, Length/Width and Embankment slope ");
                          printf ("are mismatch.");
         gotoxy (22, 19); printf ("One or more of them must be changed.");
      }
      gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                       printf ("change parameters : ");
      ch = getche();
   } while (ch != '\r' && toupper(ch) != 'C');
} while (ch != '\r');
bod = bod_eff;
fac++;
do {
   del_message(23);
   gotoxy (6, 23); printf ("Do you want to have others 2nd treatment process ");
                   printf ("following ? <Y/N> : ");
   bio = getch();
} while (toupper(bio) != 'Y' && toupper(bio) != 'N');
if (toupper(out) == 'Y') {
   fprintf (fl,"\t FACUTATIVE POND SYSTEM :\n");
   fprintf (fl,"\t facutative pond's design parameters \n");
   fprintf (fl,"\t\t effluent BOD5 requirement   =  %7.2f     mg/l \n", bod_eff);
   fprintf (fl,"\t\t k constant @ 20 °c          =  %7.2f     1/day \n", k20);
   fprintf (fl,"\t\t pond water temperature      =  %7.2f     °c \n", temp);
   fprintf (fl,"\t\t temperature coefficient (θ) =  %7.2f \n", coef);
   fprintf (fl,"\t\t water depth                 =  %7.2f     m. \n", dep);
   fprintf (fl,"\t\t length/width (approx.)      =  %7.2f \n", lw);
   fprintf (fl,"\t\t embankment slope         1  :  %7.2f \n", slope);
   fprintf (fl,"\t\t free board                 =  %7.2f     m. \n", free);
   fprintf (fl,"\t result of facutative pond process design \n");
   fprintf (fl,"\t\t detention time             =  %7.2f     day. \n", t0);
   fprintf (fl,"\t\t volumetric loading         =  %7.2f     g BOD5/cu.m-d\n",load);
   fprintf (fl,"\t\t water volume               =  %7.0f     cu.m \n", w_vol);
   if (w_width > 0) {
      fprintf (fl,"\t\t tank volume              =  %7.0f     cu.m \n", p_vol);
      fprintf (fl,"\t\t dimensions : \n");
      fprintf (fl,"\t\t    surface width         =  %7.2f     m. \n", s_width);
      fprintf (fl,"\t\t    surface length        =  %7.2f     m. \n", s_length);
      fprintf (fl,"\t\t    bottom width          =  %7.2f     m. \n", b_width);
      fprintf (fl,"\t\t    bottom length         =  %7.2f     m. \n", b_length);
      fprintf (fl,"\t\t    depth                 =  %7.2f     m. \n\n", depth);
   }
}
if (fac == 1) {
   strcpy (profile_name[7],"Facutative pond");
```

```
        putc ('f',ft);
        fi = fopen ("fac_p.cvl","w");
        fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
        fprintf (fi,"%f %f, 0.0, 1.0);
        fclose (fi);
    }
    else if (fac == 2) {
        strcpy,(profile_name[8],"Facutative pond #2");
        putc ('k',ft);
        fi = fopen ("fac_p2.cvl","w");
        fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
        fprintf (fi,"%f %f, 0.0, 1.0);
        fclose (fi);
    }
    else if (fac == 3) {
        strcpy (profile_name[9],"Facutative pond #3");
        putc ('n',ft);
        fi = fopen ("fac_p3.cvl","w");
        fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
        fprintf (fi,"%f %f, 0.0, 1.0);
        fclose (fi);
    }
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
extern float flow, pf, p_flow, bod, ss, tkn;
extern int n;
extern int mat;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
extern float trial(float w_vol, float depth, float lw, float slope, float a_width);
/*****************************************************************************/
mat_pond()
{
  int col, count;
  float val;
  float hrt, dep, lw, slope, free, no;
  float w_vol, p_vol, a_area, w_area, s_area, depth, a_width;
  float b_width, w_width, s_width, b_length, w_length, s_length;
  char ans, ch;
  char s_name[25], d_name[12];
  FILE *fi;
  col = 26;
  strcpy(s_name, " Maturation pond system ");
  strcpy(d_name, "d_mat_p.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f %f", &hrt, &dep, &lw, &slope, &free, &no);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (22, 3); printf (" Maturation pond's design parameters ");
        gotoxy (18, 8); printf ("[1] Retention time            =  %6.2f   day.", hrt);
        gotoxy (18, 10); printf ("[2] Water depth               =  %6.2f   m.", dep);
        gotoxy (18, 12); printf ("[3] Length/width (approx.)    =  %6.2f ", lw);
        gotoxy (18, 14); printf ("[4] Embankment slope       1  :  %6.2f ", slope);
        gotoxy (18, 16); printf ("[5] Free board                =  %6.2f   m.", free);
        gotoxy (18, 18); printf ("[6] No. of pond in series     =  %3.0f ", no);
        gotoxy (5, 23); printf ("ENTER = accept parameters,  [1-6] for change ");
                        printf ("parameter,  [H] = HELP : ");
        ch = getche();
        if (toupper(ch) == 'H')
           read_txt ("c_mat_p.txt");
        if (ch >= '1' && ch <= '6') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ",ch);
           scanf ("%f", &val);
           count++;
           n = 0;
           switch (ch) {
```

```
            case '1' : hrt = val;
                       break;
            case '2' : dep = val;
                       break;
            case '3' : lw = val;
                       break;
            case '4' : slope = val;
                       break;
            case '5' : free = val;
                       break;
            case '6' : no = val;
         }
      }
   } while (ch != '\r');
   if (count != 0) {
      ans = save_file();
      if (toupper(ans) == 'Y') {
         fprintf (fp, "%f %f %f %f %f %f", hrt, dep, lw, slope, free, no);
         fclose (fp);
      }
   }
   w_vol = flow * hrt;
   a_area = w_vol / dep;
   a_width = sqrt (a_area / lw);
   w_width = trial(w_vol, dep, lw, slope, a_width);
   if (w_width > 0) {
      w_length = lw * w_width;
      w_area = w_width * w_length;
      s_width = w_width + 2*slope*free;
      s_length = w_length + 2*slope*free;
      s_area = s_width * s_length;
      p_vol = w_vol + (s_area + w_area + sqrt(s_area*w_area))*free/3;
      b_width = w_width - 2*slope*dep;
      b_length = w_length - 2*slope*dep;
      depth = dep + free;
   }
   count = 0;
   do {
      frame (x1, y1, x2, y2);
      ins_frame();
      gotoxy (20, 3); printf (" Result of MATURATION POND process design ");
      gotoxy (20, 8); printf ("Water volume          =  %8.2f    cu.m ", w_vol);
      if (w_width > 0) {
         gotoxy (20, 10); printf ("Pond volume           =  %8.2f    cu.m ", p_vol);
         gotoxy (20, 12); printf ("Dimensions : ");
         gotoxy (20, 13); printf ("   Surface width      =  %8.2f    m. ", s_width);
         gotoxy (20, 14); printf ("   Surface length     =  %8.2f    m. ", s_length);
         gotoxy (20, 15); printf ("   Bottom width       =  %8.2f    m. ", b_width);
         gotoxy (20, 16); printf ("   Bottom length      =  %8.2f    m. ", b_length);
         gotoxy (20, 17); printf ("   Depth              =  %8.2f    m. ", depth);
      }
      else {
         gotoxy (10, 18); printf ("Water depth, Length/Width and Embankment slope ");
                          printf ("are mismatch.");
         gotoxy (22, 19); printf ("One or more of them must be changed.");
      }
```

```
      gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                       printf ("change parameters : ");
      ch = getche();
    } while (ch != '\r' && toupper(ch) != 'C');
  } while (ch != '\r');
  mat++;
  do {
    del_message(23);
    gotoxy (6, 23); printf ("Do you want to have others 2nd treatment process ");
                    printf ("following ? <Y/N> : ");
    bio = getch();
  } while (toupper(bio) != 'Y' && toupper(bio) != 'N');
  if (toupper(out) == 'Y') {
    fprintf (fl,"\t MATURATION POND SYSTEM :\n");
    fprintf (fl,"\t maturation pond's design parameters \n");
    fprintf (fl,"\t\t retention time                    = %8.2f      day.\n", hrt);
    fprintf (fl,"\t\t water depth                       = %8.2f      m. \n", dep);
    fprintf (fl,"\t\t length/width (approx.)            = %8.2f \n", lw);
    fprintf (fl,"\t\t embankment slope              1  : %8.2f \n", slope);
    fprintf (fl,"\t\t free board                        = %8.2f      m.\n", free);
    fprintf (fl,"\t\t no. of ponds in series            = %5.0f \n", no);
    fprintf (fl,"\t result of maturation pond process design \n");
    fprintf (fl,"\t\t water volume                      = %8.2f     cu.m \n", w_vol);
    if (w_width > 0) {
      fprintf (fl,"\t\t pond volume                     = %8.2f      cu.m \n", p_vol);
      fprintf (fl,"\t\t dimensions : \n");
      fprintf (fl,"\t\t    surface width               = %8.2f      m. \n", s_width);
      fprintf (fl,"\t\t    surface length              = %8.2f      m. \n", s_length);
      fprintf (fl,"\t\t    bottom width                = %8.2f      m. \n", b_width);
      fprintf (fl,"\t\t    bottom length               = %8.2f      m. \n", b_length);
      fprintf (fl,"\t\t    depth                       = %8.2f      m. \n\n", depth);
    }
  }
  if (mat == 1) {
    strcpy (profile_name[10],"Maturation pond");
    putc ('m',ft);
    fi = fopen ("mat_p.cvl","w");
    fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
    fprintf (fi,"%f %f", 0.0, no);
    fclose (fi);
  }
  else if (mat == 2) {
    strcpy (profile_name[11],"Maturation pond #2");
    putc ('q',ft);
    fi = fopen ("mat_p2.cvl","w");
    fprintf (fi,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
    fprintf (fi,"%f %f", 0.0, no);
    fclose (fi);
  }
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
extern float flow, pf, p_flow, bod, ss, tkn;
extern float mlss, mlrss;
extern int n;
extern int k;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
extern float trial(float w_vol, float depth, float lw, float slope, float a_width);
/*****************************************************************************/
a1()
{
  int col, count;
  float val;
  float bod_eff, k20, temp, coef, dep, lw, slope, free, ob, aerate, mix;
  float w_vol, p_vol, a_area, w_area, s_area, depth, a_width;
  float b_width, w_width, s_width, b_length, w_length, s_length, t0;
  float bod_r, o2_req, o2_hp, mix_hp, hp;
  double kt;
  char ans, ch;
  char s_name[24], d_name[10];
  FILE *f0;
  col = 30;
  strcpy(s_name, " Aerated lagoon system ");
  strcpy(d_name, "d_a1.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f", &bod_eff, &k20, &temp, &coef, &dep);
  fscanf (fp,"%f %f %f %f %f %f", &lw, &slope, &free, &ob, &aerate, &mix);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (22, 3); printf (" Aerated lagoon's design parameters ");
        gotoxy (9, 8); printf ("[1] Effluent BOD5 requirement    = %6.2f  ", bod_eff);
                       printf ("mg/l");
        gotoxy (9, 9); printf ("[2] k constant @ 20 'c           = %6.2f  1/day",k20);
        gotoxy (9, 10); printf ("[3] Lagoon water temperature     = %6.2f  'c", temp);
        gotoxy (9, 11); printf ("[4] Temperature coefficient (θ)  = %7.3f", coef);
        gotoxy (9, 12); printf ("[5] Water depth                  = %6.2f  m.", dep);
        gotoxy (9, 13); printf ("[6] Length/Width (approx.)       = %6.2f ", lw);
        gotoxy (9, 14); printf ("[7] Embankment slope          1  :  %6.2f ", slope);
        gotoxy (9, 15); printf ("[8] Free board                   = %6.2f  m.", free);
        gotoxy (9, 16); printf ("[9] O2-required/BOD5-removed ratio    = %6.2f ", ob);
        gotoxy (9, 17); printf ("[A] O2 transfer capacity of aerator   = ");
                        printf ("%6.2f  kg O2/HP-hr",aerate);
        gotoxy (9, 18); printf ("[B] Mixing energy requirement         = %6.2f", mix);
                        printf ("  HP/1000 cu.m");
```

```
  gotoxy (5, 23); printf ("ENTER = accept parameters, [1-9,A-B] for change ");
                  printf ("parameter, H = HELP : ");
  ch = getche();
  ch = toupper(ch);
  if (toupper(ch) == 'H') {
     read_txt ("c_al.txt");
     read_txt ("c_aerate.txt");
  }
  if ((ch >= '1' && ch <= '9') || (ch >= 'A' && ch <= 'B')) {
     del_message (23);
     gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
     scanf ("%f", &val);
     count++;
     n = 0;
     switch (ch) {
        case '1' : bod_eff = val;  break;
        case '2' : k20 = val;  break;
        case '3' : temp = val;  break;
        case '4' : coef = val;  break;
        case '5' : dep = val;  break;
        case '6' : lw = val;  break;
        case '7' : slope = val;  break;
        case '8' : free = val;  break;
        case '9' : ob = val;  break;
        case 'A' : aerate = val; break;
        case 'B' : mix = val;
     }
  }
} while (ch != '\r');
if (count != 0) {
   ans = save_file();
   if (toupper(ans) == 'Y') {
      fprintf (fp, "%f %f %f %f %f ", bod_eff, k20, temp, coef, dep);
      fprintf (fp, "%f %f %f %f %f %f", lw, slope, free, ob, aerate, mix);
      fclose (fp);
   }
}
kt = k20 * pow(coef, temp - 20);
if (k > 0)
   kt = kt * pow(0.8, k);
t0 = (bod / bod_eff -1) / (float)kt;
w_vol = flow * t0;
a_area = w_vol / dep;
a_width = sqrt (a_area / lw);
w_width = trial (w_vol, dep, lw, slope, a_width);
if (w_width > 0) {
   w_length = lw * w_width;
   w_area = w_width * w_length;
   s_width = w_width + 2*slope*free;
   s_length = w_length + 2*slope*free;
   s_area = s_width * s_length;
   p_vol = w_vol + (s_area + w_area + sqrt(s_area*w_area))*free/3;
   b_width = w_width - 2*slope*dep;
   b_length = w_length - 2*slope*dep;
   depth = dep + free;
}
```

```
bod_r = (bod - bod_eff) * flow / 1000;
o2_req = ob * bod_r;
o2_hp = o2_req / 24 / aerate;
mix_hp = mix * w_vol / 1000;
hp = (o2_hp > mix_hp) ? o2_hp : mix_hp;
count = 0;
do {
   frame (x1, y1, x2, y2);
   ins_frame();
   gotoxy (20, 3); printf (" Result of AERATED LAGOON process design ");
   gotoxy (19, 7); printf ("Detention time          =  %8.2f   day. ", t0);
   gotoxy (19, 8); printf ("Energy requirement      =  %8.2f   HP. ", hp);
   gotoxy (19, 10); printf ("Water volume            =  %8.2f   cu.m ", w_vol);
   if (w_width > 0) {
      gotoxy (19, 11); printf ("Lagoon volume          =  %8.2f   cu.m ", p_vol);
      gotoxy (19, 13); printf ("Dimensions : ");
      gotoxy (19, 14); printf ("   Surface width       =  %8.2f   m. ", s_width);
      gotoxy (19, 15); printf ("   Surface length      =  %8.2f   m. ", s_length);
      gotoxy (19, 16); printf ("   Bottom width        =  %8.2f   m. ", b_width);
      gotoxy (19, 17); printf ("   Bottom length       =  %8.2f   m. ", b_length);
      gotoxy (19, 18); printf ("   Depth               =  %8.2f   m. ", depth);
   }
   else {
      gotoxy (10, 18); printf ("Water depth, Length/Width and Embankment slope ");
                       printf ("are mismatch.");
      gotoxy (22, 19); printf ("One or more of them must be changed.");
   }
   gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                    printf ("change parameters : ");
   ch = getche();
  } while (ch != '\r' && toupper(ch) != 'C');
} while (ch != '\r');
bod = bod_eff;
mlss = 700;
k++;
do {
   del_message(23);
   gotoxy (6, 23); printf ("Do you want to have others 2nd treatment process ");
                   printf ("following ? <Y/N> : ");
   bio = getch();
} while (toupper(bio) != 'Y' && toupper(bio) != 'N');
if (toupper(out) == 'Y') {
   fprintf (fl,"\t AERATED LAGOON SYSTEM :\n");
   fprintf (fl,"\t aerated lagoon's design parameters \n");
   fprintf (fl,"\t\t effluent BOD5 requirement        =  %8.2f  mg/l \n", bod_eff);
   fprintf (fl,"\t\t k constant @ 20 °c               =  %8.2f  1/day \n", k20);
   fprintf (fl,"\t\t lagoon water temperature         =  %8.2f  °c \n", temp);
   fprintf (fl,"\t\t temperature coefficient (θ)      =  %9.3f \n", coef);
   fprintf (fl,"\t\t water depth                     =  %8.2f  m. \n", dep);
   fprintf (fl,"\t\t length/width (approx.)          =  %8.2f \n", lw);
   fprintf (fl,"\t\t embankment slope              1  :  %8.2f \n", slope);
   fprintf (fl,"\t\t free board                      =  %8.2f  m. \n", free);
   fprintf (fl,"\t\t O2-required/BOD5-removed ratio  =  %8.2f \n", ob);
   fprintf (fl,"\t\t O2 transfer capacity of aerator  =  %8.2f  ", aerate);
   fprintf (fl,"kg O2/HP-hr\n");
   fprintf (fl,"\t\t mixing energy requirement       =  %8.2f  HP/1000cu.m\n",mix);
```

```
    fprintf (fl,"\t result of aerated lagoon process design \n");
    fprintf (fl,"\t\t detention time                    =  %8.2f  day. \n", t0);
    fprintf (fl,"\t\t energy requirement                =  %8.2f  HP. \n", hp);
    fprintf (fl,"\t\t water volume                      =  %8.2f  cu.m \n", w_vol);
    if (w_width > 0) {
       fprintf (fl,"\t\t tank volume                        =  %8.2f  cu.m \n", p_vol);
       fprintf (fl,"\t\t dimensions : \n");
       fprintf (fl,"\t\t    surface width                 =  %8.2f  m. \n", s_width);
       fprintf (fl,"\t\t    surface length                =  %8.2f  m. \n",s_length);
       fprintf (fl,"\t\t    bottom width                  =  %8.2f  m. \n", b_width);
       fprintf (fl,"\t\t    bottom length                 =  %8.2f  m. \n",b_length);
       fprintf (fl,"\t\t    depth                         =  %8.2f  m. \n", depth);
    }
    fprintf (fl,"\n");
 }
 if (k == 1) {
    strcpy (profile_name[12],"Aerated lagoon");
    putc ('l',ft);
    f0 = fopen ("a1.cvl","w");
    fprintf (f0,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
    fprintf (f0,"%f %f", 0.0, 1.0);
    fclose (f0);
    f0 = fopen ("a1.eqp","w");
    fprintf (f0,"%f", hp);
    fclose (f0);
 }
 else if (k == 2) {
    strcpy (profile_name[13],"Aerated lagoon #2");
    putc ('t',ft);
    f0 = fopen ("a12.cvl","w");
    fprintf (f0,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
    fprintf (f0,"%f %f", 0.0, 1.0);
    fclose (f0);
    f0 = fopen ("a12.eqp","w");
    fprintf (f0,"%f", hp);
    fclose (f0);
 }
 else if (k == 3) {
    strcpy (profile_name[14],"Aerated lagoon #3");
    putc ('v',ft);
    f0 = fopen ("a13.cvl","w");
    fprintf (f0,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
    fprintf (f0,"%f %f", 0.0, 1.0);
    fclose (f0);
    f0 = fopen ("a13.eqp","w");
    fprintf (f0,"%f", hp);
    fclose (f0);
 }
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
extern float flow, pf, p_flow, bod, ss, tkn;
extern float mlss, mlrss;
extern float as_sludge;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
extern float trial(float w_vol, float depth, float lw, float slope, float a_width);
/*****************************************************************************/
as()
{
  int col, count;
  float val;
  float fm, yield, kd, bod_r, dep, lw, slope, free, ratio, aerate, mix;
  float qr, food, mass, age, vol_load, t0;
  float w_vol, p_vol, a_area, w_area, s_area, depth, a_width;
  float b_width, w_width, s_width, b_length, w_length, s_length;
  float bod_eff, bod_rm, o2_req, o2_hp, mix_hp, hp;
  char ans, ch;
  char s_name[26], d_name[10];
  FILE *f9;
  col = 29;
  strcpy(s_name, " Activated sludge system ");
  strcpy(d_name, "d_as.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f %f %f", &fm, &mlss, &mlrss, &yield, &kd, &bod_r, &ratio);
  fscanf (fp,"%f %f %f %f %f %f", &dep, &lw, &slope, &free, &aerate, &mix);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (21, 3); printf (" Activated sludge's design parameters ");
        gotoxy (8, 7); printf ("[1] F/M                          =  %7.2f ", fm);
        gotoxy (8, 8); printf ("[2] MLSS                         =  %7.2f   mg/l",mlss);
        gotoxy (8, 9); printf ("[3] MLRSS                        =  %7.2f   mg/l",mlrss);
        gotoxy (8, 10); printf ("[4] Yield coefficient, (Y)      =  %7.2f ", yield);
        gotoxy (8, 11); printf ("[5] Decay coefficient, (Kd)     =  %7.2f   1/day",kd);
        gotoxy (8, 12); printf ("[6] Percent of BOD removal      =  %4.0f ", bod_r);
        gotoxy (8, 13); printf ("[7] BOD_5 / BOD_u               =  %7.2f ", ratio);
        gotoxy (8, 14); printf ("[8] Water depth                 =  %7.2f   m.", dep);
        gotoxy (8, 15); printf ("[9] Length/Width (approx.)      =  %7.2f ", lw);
        gotoxy (8, 16); printf ("[A] Embankment slope        1   :  %7.2f ", slope);
        gotoxy (8, 17); printf ("[B] Free board                  =  %7.2f   m.", free);
        gotoxy (8, 18);
        printf ("[C] O2 transfer capacity of aerator   =  %7.2f   kg O2/HP-hr",aerate);
        gotoxy (8, 19); printf ("[D] Mixing energy requirement  =  %7.2f   ", mix);
```

```
                printf ("HP/1000 cu.m");
gotoxy (5, 23); printf ("ENTER = accept parameters, [1-9,A-D] for change ");
                printf ("parameter, H = HELP : ");
ch = getche();
ch = toupper(ch);
if (ch == 'H') {
   read_txt ("c_as_1.txt");
   read_txt ("c_as_2.txt");
   read_txt ("c_as_3.txt");
   read_txt ("c_aerate.txt");
}
if ((ch >= '1' && ch <= '9') || (ch >= 'A' && ch <= 'D')) {
   del_message (23);
   gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
   scanf ("%f", &val);
   count++;
   n = 0;
   switch (ch) {
      case '1' : fm = val;
                 break;
      case '2' : mlss = val;
                 break;
      case '3' : mlrss = val;
                 break;
      case '4' : yield = val;
                 break;
      case '5' : kd = val;
                 break;
      case '6' : bod_r = val;
                 break;
      case '7' : ratio = val;
                 break;
      case '8' : dep = val;
                 break;
      case '9' : lw = val;
                 break;
      case 'A' : slope = val;
                 break;
      case 'B' : free = val;
                 break;
      case 'C' : aerate = val;
                 break;
      case 'D' : mix = val;
    }
  }
} while (ch != '\r');
if (count != 0) {
  ans = save_file();
  if (toupper(ans) == 'Y') {
     fprintf (fp, "%f %f %f %f %f %f %f ", fm,mlss,mlrss,yield,kd,bod_r,ratio);
     fprintf (fp, "%f %f %f %f %f %f", dep, lw, slope, free, aerate, mix);
     fclose (fp);
  }
}
qr = (mlss / (mlrss-mlss)) * flow;
food = flow * bod / 1000;
```

```
mass = food / fm;
w_vol = mass * 1000 / mlss;
t0 = 24 * w_vol / (flow + qr);
vol_load = food / w_vol;
as_sludge = yield * bod_r / 100 * food - kd * mass;
age = mass / as_sludge;
bod_eff = (1 - bod_r / 100) * bod;
bod_rm = (bod - bod_eff) * flow / 1000;
if (fm < 0.2)
   o2_req = bod_rm / ratio - (1.42 * as_sludge) + (4.3 * tkn / 1000 * flow);
else
   o2_req = bod_rm / ratio - (1.42 * as_sludge);
o2_hp = o2_req / 24 / aerate;
mix_hp = mix * w_vol / 1000;
hp = (o2_hp > mix_hp) ? o2_hp : mix_hp;
a_area = w_vol / dep;
a_width = sqrt (a_area / lw);
w_width = trial (w_vol, dep, lw, slope, a_width);
if (w_width > 0) {
   w_length = lw * w_width;
   w_area = w_width * w_length;
   s_width = w_width + 2*slope*free;
   s_length = w_length + 2*slope*free;
   s_area = s_width * s_length;
   p_vol = w_vol + (s_area + w_area + sqrt(s_area*w_area))*free/3;
   b_width = w_width - 2*slope*dep;
   b_length = w_length - 2*slope*dep;
   depth = dep + free;
}
count = 0;
do {
   frame (x1, y1, x2, y2);
   ins_frame();
   gotoxy (19, 3); printf (" Result of ACTIVATED SLUDGE process design ");
   gotoxy (16, 6); printf ("Detention time          = %8.2f   hr. ", t0);
   gotoxy (16, 7); printf ("Sludge age              = %8.2f   day. ", age);
   gotoxy (16, 8); printf ("Volumetic loading       = %8.2f   ", vol_load);
                   printf ("kg/cu.m-d");
   gotoxy (16, 9); printf ("Effluent soluble BOD5   = %8.2f   mg/l ", bod_eff);
   gotoxy (16, 10); printf ("Excess sludge           = %8.2f   kg/d ", as_sludge);
   gotoxy (16, 11); printf ("Energy requirement      = %8.2f   HP. ", hp);
   gotoxy (16, 13); printf ("Water volume            = %8.2f   cu.m ", w_vol);
   gotoxy (16, 15); printf ("Dimensions : ");
   if (w_width > 0) {
      gotoxy (16, 14); printf ("Unit volume              = %8.2f   cu.m ", p_vol);
      gotoxy (16, 16); printf ("   Surface width         = %8.2f   m. ", s_width);
      gotoxy (16, 17); printf ("   Surface length        = %8.2f   m. ", s_length);
      gotoxy (16, 18); printf ("   Bottom width          = %8.2f   m. ", b_width);
      gotoxy (16, 19); printf ("   Bottom length         = %8.2f   m. ", b_length);
      gotoxy (16, 20); printf ("   Depth                 = %8.2f   m. ", depth);
   }
   else {
      gotoxy (10, 18); printf ("Water depth, Length/Width and Embankment slope ");
                       printf ("are mismatch.");
      gotoxy (22, 19); printf ("One or more of them must be changed.");
   }
```

```
      gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                       printf ("change parameters : ");
      ch = getche();
    } while (ch != '\r' && toupper(ch) != 'C');
  } while (ch != '\r');
  bod = bod_eff;
  if (toupper(out) == 'Y') {
     fprintf (f1,"\t ACTIVATED SLUDGE SYSTEM :\n");
     fprintf (f1,"\t activated sludge's design parameters \n");
     fprintf (f1,"\t\t F/M                                    =  %8.2f \n", fm);
     fprintf (f1,"\t\t MLSS                                   =  %8.2f      mg/l \n",mlss);
     fprintf (f1,"\t\t MLRSS                                  =  %8.2f      mg/l \n",mlrss);
     fprintf (f1,"\t\t Yield coefficient, (Y)                 =  %8.2f \n", yield);
     fprintf (f1,"\t\t Decay coefficient, (Kd)                =  %8.2f      1/day \n",kd);
     fprintf (f1,"\t\t Percent of BOD removal                 =  %5.0f \n", bod_r);
     fprintf (f1,"\t\t BOD_5 / BOD_u                         =  %8.2f \n", ratio);
     fprintf (f1,"\t\t water depth                           =  %8.2f      m. \n", dep);
     fprintf (f1,"\t\t length / width (approx.)             =  %8.2f \n", lw);
     fprintf (f1,"\t\t embankment slope                 1  :  %8.2f \n", slope);
     fprintf (f1,"\t\t free board                           =  %8.2f      m. \n", free);
     fprintf (f1,"\t\t O2 transfer capacity of aerator     =  %8.2f      ", aerate);
     fprintf (f1,"kg O2/HP-hr \n");
     fprintf (f1,"\t\t mixing energy requirement           =  %8.2f      ", mix);
     fprintf (f1,"HP/1000 cu.m\n");
     fprintf (f1,"\t result of activated sludge process design \n");
     fprintf (f1,"\t\t detention time                      =  %8.2f  hr. \n", t0);
     fprintf (f1,"\t\t sludge age                          =  %8.2f  day. \n", age);
     fprintf (f1,"\t\t organic loading                     =  %8.2f  ", vol_load);
     fprintf (f1,"kg/cu.m-d \n";
     fprintf (f1,"\t\t effluent soluble BOD5               =  %8.2f  mg/l \n", bod_eff);
     fprintf (f1,"\t\t excess sludge                       =  %8.2f  kg/d \n",as_sludge);
     fprintf (f1,"\t\t energy requirement                  =  %8.2f  HP. \n", hp);
     fprintf (f1,"\t\t water volume                        =  %8.2f  cu.m \n", w_vol);
     if (w_width > 0) {
        fprintf (f1,"\t\t tank volume                          =  %8.2f  cu.m \n", p_vol);
        fprintf (f1,"\t\t dimensions : \n");
        fprintf (f1,"\t\t    surface width                    =  %8.2f  m. \n", s_width);
        fprintf (f1,"\t\t    surface length                   =  %8.2f  m. \n",s_length);
        fprintf (f1,"\t\t    bottom width                     =  %8.2f  m. \n", b_width);
        fprintf (f1,"\t\t    bottom length                    =  %8.2f  m. \n",b_length);
        fprintf (f1,"\t\t    depth                            =  %8.2f  m. \n", depth);
     }
     fprintf (f1,"\n");
  }
  strcpy (profile_name[15],"Aeration tank");
  putc('s',ft);
  f9 = fopen ("as.cvl","w");
  fprintf (f9,"%f %f %f %f %f %f ", p_vol, slope, b_width, b_length, depth, dep);
  fprintf (f9,"%f %f", 0.0, 1.0);
  fclose (f9);
  f9 = fopen ("as.eqp","w");
  fprintf (f9,"%f %f %f %f", s_width, s_length, depth, hp);
  fclose (f9);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define PI 3.141592654
extern float flow, pf, p_flow, bod, ss, tkn;
extern float mlss, mlrss;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
/*****************************************************************************/
sec_cla()
{
  int col, count;
  float val;
  float ofr, dep, free, no, percent;
  float flow1, qr, t_flow, area, volume, t_vol, hrt, depth, dia, s_load;
  float ofr_p, hrt_p, load_p;
  char ans, ch;
  char s_name[34], d_name[10];
  FILE *fi;
  col = 23;
  strcpy(s_name, " Secondary clarifier tank system ");
  strcpy(d_name, "d_sec.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f", &ofr, &dep, &free, &no, &percent);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (17, 3); printf (" Secondary clarifier tank's design parameters ");
        gotoxy (14, 7); printf ("[1]  Influent MLSS            =  %7.2f   mg/l",mlss);
        gotoxy (14, 9);
        printf ("[2]  Overflow rate (ave. Q)    =  %7.2f   cu.m/sq.m-d", ofr);
        gotoxy (14, 11); printf ("[3]  Water depth              =  %7.2f   m.", dep);
        gotoxy (14, 13); printf ("[4]  free board               =  %7.2f   m.", free);
        gotoxy (14, 15); printf ("[5]  No. of tanks             =  %4.0f ", no);
        gotoxy (14, 17); printf ("[6]  Percent of flow rate     =  %4.0f ", percent);
        gotoxy (14, 18); printf ("      for each tank");
        gotoxy (5, 23); printf ("ENTER = accept parameters,  [1-6] for change ");
                        printf ("parameter,  [H] = HELP : ");
        ch = getche();
        if (toupper(ch) == 'H')
           read_txt ("c_sec.txt");
        if (ch >= '1' && ch <= '6') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           count++;
```

```
      n = 0;
      switch (ch) {
         case '1' : mlss = val;
                    break;
         case '2' : ofr = val;
                    break;
         case '3' : dep = val;
                    break;
         case '4' : free = val;
                    break;
         case '5' : no = val;
                    break;
         case '6' : percent = val;
       }
    }
} while (ch != '\r');
if (count != 0) {
   ans = save_file();
   if (toupper(ans) == 'Y') {
      fprintf (fp, "%f %f %f %f %f", ofr, dep, free, no, percent);
      fclose (fp);
   }
}
flow1 = flow * percent / 100;
if (mlrss == 0)
   qr = 0;
else
   qr = (mlss / (mlrss - mlss)) * flow1;
t_flow = flow1 + qr;
area = t_flow / ofr;
dia = sqrt (area * 4 / PI);
volume = area * dep;
depth = dep + free;
t_vol = area * depth;
hrt = 24 * volume / flow1 ;
s_load = (mlss / 1000) * ofr / 24;
ofr_p = p_flow / (no * area);
hrt_p = 24 * volume * no / p_flow;
load_p = (mlss / 1000) * ofr_p / 24;
count = 0;
do {
   frame (x1, y1, x2, y2);
   ins_frame();
   gotoxy (15, 3); printf (" Result of SECONDARY CLARIFIER TANK process design ");
   gotoxy (14, 7); printf ("Detention time          = %7.2f hr. ", hrt);
   gotoxy (14, 8); printf ("Solids loading          = %7.2f kg/sq.m-hr ",s_load);
   gotoxy (14, 10); printf ("Detention time (peak Q) = %7.2f hr.", hrt_p);
   gotoxy (14, 11); printf ("Overflow rate (peak Q)  = %7.2f cu.m/sq.m-d",ofr_p);
   gotoxy (14, 12); printf ("Solids loading (peak Q) = %7.2f kg/sq.m-hr",load_p);
   gotoxy (14, 14); printf ("Water volume           = %7.2f cu.m ", volume);
   gotoxy (14, 15); printf ("Tank volume            = %7.2f cu.m ", t_vol);
   gotoxy (14, 16); printf ("Dimensions : ");
   gotoxy (14, 17); printf ("   Diameter            = %7.2f m. ", dia);
   gotoxy (14, 18); printf ("   Depth               = %7.2f m. ", depth);
   gotoxy (10, 23); printf("Enter = accept result,   [C] = change parameters : ");
   ch = getche();
```

```
    } while (ch != '\r' && toupper(ch) != 'C');
  } while (toupper(ch) == 'C');
  if (toupper(out) == 'Y') {
     fprintf (fl,"\t SECONDARY CLARIFIER TANK SYSTEM :\n");
     fprintf (fl,"\t secondary clarifier tank's design parameters \n");
     fprintf (fl,"\t\t influent MLSS                = %8.2f      mg/l \n", mlss);
     fprintf (fl,"\t\t overflow rate (ave. Q)       = %8.2f      cu.m/sq.m-d \n", ofr);
     fprintf (fl,"\t\t water depth                  = %8.2f      m. \n", dep);
     fprintf (fl,"\t\t free board                   = %8.2f      m. \n", free);
     fprintf (fl,"\t\t no. of tank                  = %5.0f \n", no);
     fprintf (fl,"\t\t percent of flow rate -       = %5.0f \n", percent);
     fprintf (fl,"\t\t  for each tank \n");
     fprintf (fl,"\t result of secondary clarifier tank process design \n");
     fprintf (fl,"\t\t detention time               = %8.2f      hr. \n", hrt);
     fprintf (fl,"\t\t solids loading              = %8.2f      kg/sq.m-hr\n",s_load);
     fprintf (fl,"\t\t detention time (peak Q)      = %8.2f      hr. \n", hrt_p);
     fprintf (fl,"\t\t overflow rate (peak Q)       = %8.2f      cu.m/sq.m-d\n",ofr_p);
     fprintf (fl,"\t\t solids loading (peak Q)     = %8.2f      kg/sq.m-hr\n",load_p);
     fprintf (fl,"\t\t water volume                = %8.2f      cu.m \n", volume);
     fprintf (fl,"\t\t tank volume                 = %8.2f      cu.m \n", t_vol);
     fprintf (fl,"\t\t dimensions : \n");
     fprintf (fl,"\t\t    diameter                 = %8.2f      m. \n", dia);
     fprintf (fl,"\t\t    depth                    = %8.2f      m. \n\n", depth);
  }
  strcpy (profile_name[16],"Secondary clarifier");
  putc('c',ft);
  fi = fopen ("sec_cla.cv1","w");
  fprintf (fi,"%f %f %f %f %f %f %f %f", 0.0, 0.0, 0.0, 0.0, depth, dep, dia, no);
  fclose (fi);
  fi = fopen ("sec.eqp","w");
  fprintf (fi,"%f %f %f", dia, depth, no);
  fclose (fi);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float flow, pf, p_flow, bod, ss, tkn;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
/*****************************************************************************/
chlorine()
{
  int col, count;
  float val;
  float ct, vel, dep, free, no;
  float vol, c_vol, c_length, c_width, t_width, t_length, depth;
  char ans, ch;
  char s_name[22], d_name[10];
  FILE *fi;
  col = 31;
  strcpy(s_name, " Chlorination system ");
  strcpy(d_name, "d_chlo.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f", &ct, &vel, &dep, &free, &no);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (23, 3); printf (" Chlorination's design parameters ");
        gotoxy (17, 9); printf ("[1] Contact time              =  %6.2f   min.", ct);
        gotoxy (17, 11); printf ("[2] Horizontal velocity      =  %6.2f   m/min", vel);
        gotoxy (17, 13); printf ("[3] Water depth              =  %6.2f   m.", dep);
        gotoxy (17, 15); printf ("[4] Free board               =  %6.2f   m.", free);
        gotoxy (17, 17); printf ("[5] No. of sub - channel     =  %3.0f ", no);
        gotoxy (5, 23); printf ("ENTER = accept parameters ,  [1-5] for change ");
                        printf ("parameter ,  H = HELP : ");
        ch = getche();
        if (toupper(ch) == 'H')
           read_txt ("c_chlo.txt");
        if (ch >= '1' && ch <= '5') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           count++;
           n = 0;
           switch (ch) {
              case '1' : ct = val;
                         break;
              case '2' : vel = val;
                         break;
              case '3' : dep = val;
```

```
                    break;
            case '4' : free = val;
                    break;
            case '5' : no = val;
          }
        }
      } while (ch != '\r');
      if (count != 0) {
        ans = save_file();
        if (toupper(ans) == 'Y') {
          fprintf (fp, "%f %f %f %f %f", ct, vel, dep, free, no);
          fclose (fp);
        }
      }
      vol = flow * ct / 1440;
      c_width = flow / (1440 * vel * dep);
      c_length = vel * ct;
      depth = dep + free;
      t_width = c_length / no;
      t_length = c_width * no + 0.1 * (no - 1);
      c_vol = c_width * c_length * depth;
      count = 0;
      do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (21, 3); printf (" Result of CHLORINATION process design ");
        gotoxy (20, 8); printf ("Water volume        =   %6.2f      cu.m ", vol);
        gotoxy (20, 9); printf ("Tank volume         =   %6.2f      cu.m ", c_vol);
        gotoxy (20, 11); printf ("Tank dimensions : ");
        gotoxy (20, 12); printf ("   Width           =   %6.2f      m. ", t_width);
        gotoxy (20, 13); printf ("   Length          =   %6.2f      m. ", t_length);
        gotoxy (20, 14); printf ("   Depth           =   %6.2f      m. ", depth);
        gotoxy (20, 16); printf ("Channel dimensions : ");
        gotoxy (20, 17); printf ("   Width           =   %6.2f      m. ", c_width);
        gotoxy (20, 18); printf ("   Length          =   %6.2f      m. ", c_length);
        gotoxy (20, 19); printf ("   Depth           =   %6.2f      m. ", depth);
        gotoxy (10, 23); printf ("Enter = accept result ,   [C] = back to ");
                         printf ("change parameters : ");
        ch = getche();
      } while (ch != '\r' && toupper(ch) != 'C');
    } while (ch != '\r');
    if (toupper(out) == 'Y') {
      fprintf (fl,"\t CHLORINATION SYSTEM :\n");
      fprintf (fl,"\t chlorination's design parameters \n");
      fprintf (fl,"\t\t contact time                    =  %7.2f    min. \n", ct);
      fprintf (fl,"\t\t horizontal velocity              =  %7.2f    m/min \n", vel);
      fprintf (fl,"\t\t water depth                     =  %7.2f    m. \n", dep);
      fprintf (fl,"\t\t free board                      =  %7.2f    m. \n", free);
      fprintf (fl,"\t\t no. of sub - channel            =  %4.0f \n", no);
      fprintf (fl,"\t result of chlorination process design \n");
      fprintf (fl,"\t\t water volume                    =  %7.2f    cu.m \n", vol);
      fprintf (fl,"\t\t tank volume                     =  %7.2f    cu.m \n", c_vol);
      fprintf (fl,"\t\t tank dimensions : \n");
      fprintf (fl,"\t\t    width                       =  %7.2f    m. \n", t_width);
      fprintf (fl,"\t\t    length                      =  %7.2f    m. \n", t_length);
      fprintf (fl,"\t\t    depth                       =  %7.2f    m. \n", depth);
```

```
        fprintf (fl,"\t\t channel dimensions : \n");
        fprintf (fl,"\t\t    width                              =  %7.2f    m. \n", c_width);
        fprintf (fl,"\t\t    length                             =  %7.2f    m. \n", c_length);
        fprintf (fl,"\t\t    depth                              =  %7.2f    m. \n\n", depth);
    }
    strcpy (profile_name[17],"Chlorine contact tank");
    putc ('h',ft);
    fi = fopen ("chlo.cvl","w");
    fprintf (fi,"%f %f %f %f %f %f %f %f", 0.0,0.0,t_width,t_length,depth,dep,0.0,1.0);
    fclose (fi);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float flow, p_flow, bod, ss;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *ft;
extern FILE *f1;
/*****************************************************************************/
oz_uv(num)
int num;
{
  char ans, ch;
  frame (x1, y1, x2, y2);
  ins_frame();
  if (num == 1) {
     gotoxy (31, 3); printf (" Ozonation system ");
  }
  else {
     gotoxy (28, 3); printf (" UV disinfection system ");
  }
  gotoxy (14, 8); printf ("Average flow              =  %8.2f          cu.m/d", flow);
  gotoxy (14, 9); printf ("Peak flow                 =  %8.2f          cu.m/d", p_flow);
  gotoxy (14, 10); printf ("BOD5                      =  %8.2f           mg/l", bod);
  gotoxy (14, 15); printf ("Consult supplier for specific type model and number ");
  gotoxy (31, 16); printf ("for the equipment");
  gotoxy (27, 23); printf ("Press any key to continue ");
  getche();
  if (toupper(out) == 'Y') {
     if (num == 1) {
        fprintf (f1,"\t OZONATION SYSTEM : \n");
        fprintf (f1,"\t Ozonation's preliminary information \n");
     }
     else {
        fprintf (f1,"\t UV DISINFECTION SYSTEM :\n");
        fprintf (f1,"\t UV disinfection's preliminary information \n");
     }
     fprintf (f1,"\t\t Average flow                 =  %8.2f      cu.m/d \n", flow);
     fprintf (f1,"\t\t Peak flow                    =  %8.2f      cu.m/d \n", p_flow);
     fprintf (f1,"\t\t BOD5                         =  %8.2f      mg/l \n", bod);
     fprintf (f1,"\t\t Consult supplier for specific type model and number ");
     fprintf (f1,"for the equipment\n\n");
  }
  if (num == 1)
     putc ('o',ft);
  else
     putc ('u',ft);
  }
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define PI 3.141592654
extern float flow, pf, p_flow, bod, ss, tkn;
extern float pri_sludge, as_sludge, sludge_vol, solid;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern char profile_name[19][25];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
/*****************************************************************************/
an_digest()
{
  int col, count;
  float tmp;
  float dsr, sgr, ratio, t, vsr, dsd, sgd, dep, free, no, percent;
  float t_sludge, d_sludge, vf, vs, fs, rvs, ts, vd, vol, s_area, dia, depth, t_vol;
  char ans, ch;
  char s_name[36], d_name[11];
  FILE *fi;
  col = 22;
  strcpy(s_name, " Anaerobic sludge-digestion system ");
  strcpy(d_name, "d_an_d.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f %f", &dsr, &sgr, &ratio, &t, &vsr, &dsd);
  fscanf (fp,"%f %f %f %f %f", &sgd, &dep, &free, &no, &percent);
  fclose (fp);
  count = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (16, 3); printf (" Anaerobic sludge-digestion's design parameters ");
        gotoxy (13, 5); printf ("     1° sludge          = %8.2f    kg/d",pri_sludge);
        gotoxy (13, 6); printf ("     2° sludge          = %8.2f    kg/d", as_sludge);
        gotoxy (13, 8); printf ("     Raw sludge : ");
        gotoxy (13, 9); printf ("[1]    %% dry solids              =  %7.2f   ", dsr);
        gotoxy (13, 10); printf ("[2]    Sludge gravity            =  %7.2f   ", sgr);
        gotoxy (13, 11); printf ("[3]  VSS/SS ratio                =  %7.2f ", ratio);
        gotoxy (13, 12); printf ("[4]  Digested time               =  %7.2f days", t);
        gotoxy (13, 13); printf ("[5]  %% volatile solids reduction =  %4.0f   ", vsr);
        gotoxy (13, 14); printf ("     Digested sludge : ");
        gotoxy (13, 15); printf ("[6]    %% dry solids             =  %7.2f   ", dsd);
        gotoxy (13, 16); printf ("[7]    Sludge gravity            =  %7.2f   ", sgd);
        gotoxy (13, 17); printf ("[8]  Sludge depth                =  %7.2f  m.", dep);
        gotoxy (13, 18); printf ("[9]  free board                  =  %7.2f  m.",free);
        gotoxy (13, 19); printf ("[A]  No. of tanks                =  %4.0f   ", no);
        gotoxy (13, 20); printf ("[B]  %% raw sludge for each tank  =  %4.0f",percent);
        gotoxy (5, 23); printf ("ENTER = accept parameters, [1-9,A-B] for change ");
                        printf ("parameter, [H] = HELP : ");
        ch = getche();
```

```
ch = toupper(ch);
if (ch == 'H') {
   read_txt ("c_sludg1.txt");
   read_txt ("c_sludg2.txt");
   read_txt ("c_an_d.txt");
}
if ((ch >= '1' && ch <= '9') || ch == 'A' || ch == 'B') {
   del_message (23);
   gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
   scanf ("%f", &tmp);
   count++;
   n = 0;
   switch (ch) {
      case '1' : dsr = tmp;  break;
      case '2' : sgr = tmp;  break;
      case '3' : ratio = tmp; break;
      case '4' : t = tmp; break;
      case '5' : vsr = tmp; break;
      case '6' : dsd = tmp; break;
      case '7' : sgd = tmp; break;
      case '8' : dep = tmp; break;
      case '9' : free = tmp; break;
      case 'A' : no = tmp; break;
      case 'B' : percent = tmp;
    }
  }
} while (ch != '\r');
if (count != 0) {
   ans = save_file();
   if (toupper(ans) == 'Y') {
      fprintf (fp,"%f %f %f %f %f %f ", dsr, sgr, ratio, t, vsr, dsd);
      fprintf (fp,"%f %f %f %f %f", sgd, dep, free, no, percent);
      fclose (fp);
   }
}
t_sludge = pri_sludge + as_sludge;
d_sludge = t_sludge * percent / 100;
vf = d_sludge / (sgr * 1000 * dsr/100);
vs = d_sludge * ratio;
fs = (1 - ratio) * d_sludge;
rvs = (1 - vsr/100) * vs;
ts = fs + rvs;
vd = ts / (sgd * 1000 * dsd/100);
vol = (vf - 2.0 / 3.0 * (vf - vd)) * t;
s_area = vol / dep;
dia = sqrt (s_area * 4 / PI);
depth = dep + free;
t_vol = s_area * depth;
count = 0;
do {
   frame (x1, y1, x2, y2);
   ins_frame();
   gotoxy (14, 3); printf (" Result of ANAEROBIC SLUDGE-DIGESTION process design ");
   gotoxy (14, 10); printf ("Fresh sludge added          = %8.2f cu.m/d", vf);
   gotoxy (14, 11); printf ("Digested sludge withdraw    = %8.2f cu.m/d", vd);
   gotoxy (14, 13); printf ("Digested sludge volume      = %8.2f cu.m ", vol);
```

```
        gotoxy (14, 14); printf ("Tank volume                    =  %8.2f cu.m ", t_vol);
        gotoxy (14, 15); printf ("Dimensions : ");
        gotoxy (14, 16); printf ("   Diameter                    =  %8.2f m. ", dia);
        gotoxy (14, 17); printf ("   Depth                       =  %8.2f m. ", depth);
        gotoxy (10, 23); printf("Enter = accept result,   [C] = change parameters : ");
        ch = getche();
      } while (ch != '\r' && toupper(ch) != 'C');
   } while (toupper(ch) == 'C');
   sludge_vol = vd;
   solid = dsd;
   if (toupper(out) == 'Y') {
      fprintf (fl,"\t ANAEROBIC SLUDGE-DIGESTION SYSTEM :\n");
      fprintf (fl,"\t anaerobic sludge-digestion's design parameters \n");
      fprintf (fl,"\t\t Raw sludge : \n");
      fprintf (fl,"\t\t   %% dry solids                    =  %8.2f \n", dsr);
      fprintf (fl,"\t\t   Sludge gravity                  =  %8.2f  \n", sgr);
      fprintf (fl,"\t\t VSS/SS ratio                       =  %8.2f  \n", ratio);
      fprintf (fl,"\t\t Digested time                      =  %8.2f      days\n", t);
      fprintf (fl,"\t\t %% volatile solids reduction       =  %8.2f \n", vsr);
      fprintf (fl,"\t\t Digested sludge : \n");
      fprintf (fl,"\t\t   %% dry solids                    =  %8.2f \n", dsd);
      fprintf (fl,"\t\t   Sludge gravity                  =  %8.2f  \n", sgd);
      fprintf (fl,"\t\t Sludge depth                       =  %8.2f      m.\n", dep);
      fprintf (fl,"\t\t Free board                         =  %8.2f      m.\n", free);
      fprintf (fl,"\t\t No. of tanks                       =  %5.0f  \n", no);
      fprintf (fl,"\t\t %% raw sludge for each tank        =  %5.0f \n", percent);
      fprintf (fl,"\t result of anaerobic sludge-digestion process design \n");
      fprintf (fl,"\t\t Fresh sludge added                 =  %8.2f      cu.m/d \n", vf);
      fprintf (fl,"\t\t Digested sludge withdraw          =  %8.2f      cu.m/d \n", vd);
      fprintf (fl,"\t\t Digested sludge volume            =  %8.2f      cu.m \n", vol);
      fprintf (fl,"\t\t Tank volume                       =  %8.2f      cu.m \n",t_vol);
      fprintf (fl,"\t\t Dimensions : \n");
      fprintf (fl,"\t\t    Diameter                      =  %8.2f      m. \n", dia);
      fprintf (fl,"\t\t    Depth                         =  %8.2f      m. \n\n",depth);
   }
   strcpy (profile_name[18],"Anaerobic digestion tank");
   putc ('d',ft);
   fi = fopen ("an_d.cvl","w");
   fprintf (fi,"%f %f %f %f %f %f %f %f", 0.0, 0.0, 0.0, 0.0, depth, dep, dia, no);
   fclose (fi);
   fi = fopen ("an_d.eqp","w");
   fprintf (fi,"%f %f %f", dia, depth, no);
   fclose (fi);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <conio.h>
extern float flow, pf, p_flow, bod, ss, tkn;
extern float pri_sludge, as_sludge, sludge_vol;
extern int n;
extern int x1, y1, x2, y2;
extern char answer, out, bio;
extern char r_name[12];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
/*****************************************************************************/
dry_bed()
{
  int col, count, i = 1;
  float val;
  float dsr, sgr, t0, dep, sand, free, lw, no;
  float d_sludge, area, width, length, depth, vol, unit, t_vol;
  char ans, ch;
  char s_name[28], d_name[10];
  FILE *fi;
  col = 26;
  strcpy(s_name, " Sludge-drying beds system ");
  strcpy(d_name, "d_dry.dat");
  read_default (col, s_name, d_name);
  fscanf (fp,"%f %f %f %f %f %f %f %f", &dsr, &sgr, &t0, &dep, &sand, &free, &lw, &no);
  fclose (fp);
  count = 0;
  if (sludge_vol == 0)
     i = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (20, 3); printf (" Sludge-drying beds' design parameters ");
        if (i == 0) {
           gotoxy (13, 8); printf ("     Raw sludge : ");
           gotoxy (13, 9); printf ("[1]     %% dry solids       = %7.2f   ", dsr);
           gotoxy (13, 10); printf ("[2]     Sludge gravity     = %7.2f   ", sgr);
           gotoxy (13, 12); printf ("[3]  drying time           = %7.2f    days", t0);
           gotoxy (13, 13); printf ("[4]  Sludge depth          = %7.2f    m.", dep);
           gotoxy (13, 14); printf ("[5]  Media depth           = %7.2f    m.", sand);
           gotoxy (13, 15); printf ("[6]  free board            = %7.2f    m.", free);
           gotoxy (13, 16); printf ("[7]  Length/Width ratio    = %7.2f    ", lw);
           gotoxy (13, 17); printf("[8] No. of tanks per sludge/day = %4.0f units",no);
           gotoxy (5, 23); printf ("ENTER = accept parameters,  [1-8] for change ");
                           printf ("parameter,  [H] = HELP : ");
           ch = getche();
           ch = toupper(ch);
           if (ch == 'H') {
              read_txt ("c_sludg1.txt");
              read_txt ("c_sludg2.txt");
           }
           if (ch >= '1' && ch <= '8') {
```

```
            del_message (23);
            gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
            scanf ("%f", &val);
            count++;
            n = 0;
            switch (ch) {
               case '1' : dsr = val;  break;
               case '2' : sgr = val;  break;
               case '3' : t0 = val;   break;
               case '4' : dep = val;  break;
               case '5' : sand = val; break;
               case '6' : free = val; break;
               case '7' : lw = val;   break;
               case '8' : no = val;
            }
         }
      }
      else {
         gotoxy (13, 8); printf ("[1]  drying time           =  %7.2f    days", t0);
         gotoxy (13, 10); printf ("[2]  Sludge depth          =  %7.2f    m.", dep);
         gotoxy (13, 12); printf ("[3]  Media depth           =  %7.2f    m.", sand);
         gotoxy (13, 14); printf ("[4]  free board            =  %7.2f    m.", free);
         gotoxy (13, 16); printf ("[5]  Length/Width ratio    =  %7.2f    ", lw);
         gotoxy (13, 18); printf("[6]  No. of tanks per sludge/day = %4.0f unit",no);
         gotoxy (11, 23);printf("ENTER = accept parameters, [1-6] change parameter");
         ch = getche();
         ch = toupper(ch);
         if (ch >= '1' && ch <= '6') {
            del_message (23);
            gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
            scanf ("%f", &val);
            count++;
            n = 0;
            switch (ch) {
               case '1' : t0 = val;  break;
               case '2' : dep = val; break;
               case '3' : sand = val; break;
               case '4' : free = val; break;
               case '5' : lw = val;  break;
               case '6' : no = val;
            }
         }
      }
   } while (ch != '\r');
   if (count != 0) {
      ans = save_file();
      if (toupper(ans) == 'Y') {
         fprintf (fp,"%f %f %f %f %f %f %f %f", dsr, sgr, t0, dep, sand, free,lw,no);
         fclose (fp);
      }
   }
   if (i == 0)
      sludge_vol = (pri_sludge + as_sludge) / (sgr * 1000 * dsr/100);
   d_sludge = sludge_vol / no;
   area = d_sludge / dep;
   width = sqrt (area / lw);
```

```
    length = width * lw;
    depth = dep + sand + free;
    vol = area * depth;
    unit = (t0 + 1) * no;
    t_vol = vol * unit;
    count = 0;
    do {
       frame (x1, y1, x2, y2);
       ins_frame();
       gotoxy (18, 3); printf (" Result of SLUDGE-DRYING BEDS process design ");
       gotoxy (14, 9); printf ("Sludge volume            = %8.2f    ", d_sludge);
                       printf ("cu.m/unit");
       gotoxy (14, 10); printf ("Tank volume             = %8.2f    cu.m/unit",vol);
       gotoxy (14, 11); printf ("Dimensions : ");
       gotoxy (14, 12); printf ("   Width                = %8.2f    m. ", width);
       gotoxy (14, 13); printf ("   Length               = %8.2f    m. ", length);
       gotoxy (14, 14); printf ("   Depth                = %8.2f    m. ", depth);
       gotoxy (14, 16); printf ("No. of sludge-drying bed = %5.0f units",unit);
       gotoxy (14, 17); printf ("Total volume            = %8.2f    cu.m ", t_vol);
       gotoxy (10, 23); printf("Enter = accept result,   [C] = change parameters : ");
       ch = getche();
    } while (ch != '\r' && toupper(ch) != 'C');
  } while (toupper(ch) == 'C');
  if (toupper(out) == 'Y') {
     fprintf (fl,"\t SLUDGE-DRYING BEDS SYSTEM : \n");
     fprintf (fl,"\t sludge-drying beds' design parameters \n");
     if (i == 0) {
        fprintf (fl,"\t\t Raw sludge : \n");
        fprintf (fl,"\t\t   %% dry solids                  = %8.2f \n", dsr);
        fprintf (fl,"\t\t   Sludge gravity                = %8.2f  \n", sgr);
     }
     fprintf (fl,"\t\t Drying time                       = %8.2f      days\n", t0);
     fprintf (fl,"\t\t Sludge depth                      = %8.2f      m.\n", dep);
     fprintf (fl,"\t\t Media depth                      = %8.2f      m.\n", sand);
     fprintf (fl,"\t\t Free board                       = %8.2f      m.\n", free);
     fprintf (fl,"\t\t Length/Width ratio               = %8.2f \n", lw);
     fprintf (fl,"\t\t No. of tanks per sludge/day       = %5.0f       units \n", no);
     fprintf (fl,"\t result of sludge-drying beds process design \n");
     fprintf (fl,"\t\t Sludge volume                  = %8.2f  cu.m/unit\n",d_sludge);
     fprintf (fl,"\t\t Tank volume                     = %8.2f  cu.m/unit \n", vol);
     fprintf (fl,"\t\t Dimensions : \n");
     fprintf (fl,"\t\t    Width                       = %8.2f      m. \n", width);
     fprintf (fl,"\t\t    Length                      = %8.2f      m. \n", length);
     fprintf (fl,"\t\t    Depth                       = %8.2f      m. \n", depth);
     fprintf (fl,"\t\t No. of sludge-drying beds       = %5.0f      units \n",unit);
     fprintf (fl,"\t\t Total volume                    = %8.2f   cu.m \n\n", t_vol);
  }
  putc ('b',ft);
  fi = fopen ("dry_bed.cvl","w");
  fprintf (fi,"%f %f %f %f %f %f %f %f", 0.0,0.0,length,width*unit,depth,-1.0,0.0,1.0);
  fclose (fi);
  fi = fopen ("bed.eqp","w");
  fprintf (fi,"%f %f %f %f", width, length, sand, unit);
  fclose (fi);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float pri_sludge, as_sludge, sludge_vol, solid;
extern int n;
extern int x1, y1, x2, y2;
extern char out;
extern char r_name[12];
extern FILE *ft;
extern FILE *fp;
extern FILE *fl;
/*****************************************************************************/
dewater()
{
  int col, count, i = 1;
  float val;
  float dsr, sgr;
  char ans, ch;
  char s_name[28], d_name[10];
  if (sludge_vol == 0) {
     i = 0;
     col = 27;
     strcpy(s_name, " Dewatering machine system ");
     strcpy(d_name, "dewater.dat");
     read_default (col, s_name, d_name);
     fscanf (fp,"%f %f", &dsr, &sgr);
     fclose (fp);
     count = 0;
     do {
        do {
           frame (x1, y1, x2, y2);
           ins_frame();
           gotoxy (20, 3); printf (" dewatering machine's design parameters ");
           gotoxy (17, 8); printf (" 1' sludge          =  %7.2f    kg/d", pri_sludge);
           gotoxy (17, 9); printf (" 2' sludge          =  %7.2f    kg/d", as_sludge);
           gotoxy (13, 12); printf ("     Raw sludge : ");
           gotoxy (13, 13); printf ("[1]     %% dry solids    =  %7.2f   ", dsr);
           gotoxy (13, 14); printf ("[2]     Sludge gravity  =  %7.2f   ", sgr);
           gotoxy (5, 23); printf ("ENTER = accept parameters,  [1,2] for change ");
                           printf ("parameter,  [H] = HELP : ");
           ch = getche();
           ch = toupper(ch);
           if (ch == 'H') {
              read_txt ("c_sludg1.txt");
              read_txt ("c_sludg2.txt");
           }
           if (ch >= '1' && ch <= '2') {
              del_message (23);
              gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
              scanf ("%f", &val);
              count++;
              n = 0;
              switch (ch) {
                 case '1' : dsr = val;
                            break;
                 case '2' : sgr = val;
```

```
            }
          }
        } while (ch != '\r');
        if (count != 0) {
           ans = save_file();
           if (toupper(ans) == 'Y') {
              fprintf (fp,"%f %f", dsr, sgr);
              fclose (fp);
           }
        }
        sludge_vol = (pri_sludge + as_sludge) / (sgr * 1000 * dsr/100);
        count = 0;
        do {
           frame (x1, y1, x2, y2);
           ins_frame();
           gotoxy (16, 3); printf (" DEWATERING MACHINE's preliminary informations ");
           gotoxy (14, 8); printf ("Sludge volume              =  %8.2f", sludge_vol);
                           printf ("cu.m/d");
           gotoxy (14, 13); printf ("Consult supplier for specific type model and number ");
           gotoxy (31, 14); printf ("for the equipment");
           gotoxy (10, 23); printf ("Enter = accept result,   [C] = back to change ");
                            printf ("parameters : ");
           ch = getche();
        } while (ch != '\r' && toupper(ch) != 'C');
     } while (toupper(ch) == 'C');
  }
  else {
     frame (x1, y1, x2, y2);
     ins_frame();
     gotoxy (16, 3); printf (" DEWATERING MACHINE's preliminary informations ");
     gotoxy (14, 8); printf ("Sludge volume          =  %8.2f    cu.m/d", sludge_vol);
     gotoxy (14, 9); printf ("%% dry solids           =  %8.2f   ", solid);
     gotoxy (14, 13); printf ("Consult supplier for specific type model and number ");
     gotoxy (31, 14); printf ("for the equipment");
     gotoxy (27, 23); printf ("Press any key to continue ");
     getche();
  }
  if (toupper(out) == 'Y') {
     fprintf (fl,"\t DEWATERING MACHINE SYSTEM : \n");
     if (i == 0) {
        fprintf (fl,"\t dewatering machine's design parameters \n");
        fprintf (fl,"\t\t Raw sludge : \n");
        fprintf (fl,"\t\t   %% dry solids                    =  %8.2f \n", dsr);
        fprintf (fl,"\t\t   Sludge gravity                  =  %8.2f  \n", sgr);
     }
     fprintf (fl,"\t dewatering machine's preliminary informations \n");
     fprintf (fl,"\t\t Sludge volume            =  %8.2f      cu.m/d \n", sludge_vol);
     if (i != 0)
        fprintf (fl,"\t\t %% dry solids                      =  %8.2f \n", solid);
     fprintf (fl,"\t\t Consult supplier for specific type model and number ");
     fprintf (fl,"for the equipment\n\n");
  }
  putc ('r',ft);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float elev[19];
extern float wall, base, degree, rc, excav, pond, struc, earth;
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern char profile_name[19][25];
extern FILE *f1;
extern FILE *ft;
/*****************************************************************************/
cost()
{
  int i, j, z;
  float civil_c;
  char ch, sure;
  z = 1;
  do {
     frame (x1, y1, x2, y2);
     if (z == 1) {
        gotoxy (10, 12);
        printf ("press [E] to exit,  others key to COST ESTIMATE processes : ");
        ch = getch();
        if (toupper(ch) == 'E') {
           del_message(12);
           gotoxy (25, 12); printf ("Are you sure to exit ? <Y/N> : ");
           sure = getch();
           if (toupper(sure) == 'Y')
              exit(1);
        }
        del_message(12);
     }
     ins_frame();
     gotoxy (31, 3); printf (" Hydraulic profile ");
     gotoxy (15, 6); printf ("Ground elevation              =   0.00          m.");
     gotoxy (9, 23); printf ("input water level of each tank. EXAM :- 0, 0.5, -1.35 ");
     j = 0;
     for (i=18; i>=1; i--) {
        if (strlen (profile_name[i]) > 2) {
           j++;
           gotoxy (15, 7+j); printf ("%s", profile_name[i]);
           gotoxy (45, 7+j); printf ("=                 m.");
        }
     }
     j = 0;
     for (i=18; i>=1; i--) {
        if (strlen (profile_name[i]) > 2) {
           j++;
           gotoxy (49, 7+j); scanf ("%f", &elev[i]);
        }
     }
     del_message(23);
     gotoxy (13, 23); printf ("press [R] to re-input data, others key to continue : ");
     ch = getch();
     z++;
```

```
} while (toupper(ch) == 'R');
if (toupper(out) == 'Y') {
   fprintf (fl,"\n\t\t\t\t   ------------------- \n");
   fprintf (fl,"\t\t\t\t    Hydraulic Profile \n");
   fprintf (fl,"\t\t\t\t   ------------------- \n\n");
   fprintf (fl,"\t\t Ground elevation                    =   %6.2f       m.\n", 0.0);
   for (i=18; i>=1; i--) {
      if (strlen (profile_name[i]) > 2) {
         fprintf (fl,"\t\t %-34s", profile_name[i]);
         fprintf (fl,"=   %6.2f       m.\n", elev[i]);
      }
   }
   fprintf (fl,"\n\n");
}
ft = fopen ("file.tmp","r");
if (toupper(out) == 'Y') {
   fprintf (fl,"\t\t\t\t ---------------------- \n");
   fprintf (fl,"\t\t\t\t  Civil cost estimation \n");
   fprintf (fl,"\t\t\t\t ---------------------- \n\n");
}
while ((ch = getc(ft)) != EOF) {
   switch (ch) {
      case 'g' : civil("GRIT CHAMBER SYSTEM :", elev[1], "grit.cvl");  break;
      case 'e' : civil("EQUALIZATION TANK SYSTEM :", elev[2], "equal.cvl");  break;
      case 'p' : civil("PRIMARY SEDIMENTATION SYSTEM ",elev[3],"pri_sed.cvl"); break;
      case 'a' : civil("ANAEROBIC POND SYSTEM :", elev[4], "an_p.cvl");  break;
      case 'i' : civil("ANAEROBIC POND SYSTEM #2 :", elev[5], "an_p2.cvl");  break;
      case 'j' : civil("ANAEROBIC POND SYSTEM #3 :", elev[6], "an_p3.cvl");  break;
      case 'f' : civil("FACUTATIVE POND SYSTEM :", elev[7], "fac_p.cvl");  break;
      case 'k' : civil("FACUTATIVE POND SYSTEM #2 :", elev[8], "fac_p2.cvl");  break;
      case 'n' : civil("FACUTATIVE POND SYSTEM #3 :", elev[9], "fac_p3.cvl");  break;
      case 'm' : civil("MATURATION POND SYSTEM :", elev[10], "mat_p.cvl");  break;
      case 'q' : civil("MATURATION POND SYSTEM #2 :", elev[11], "mat_p2.cvl"); break;
      case 'l' : civil("AERATED LAGOON SYSTEM :", elev[12], "al.cvl");  break;
      case 't' : civil("AERATED LAGOON SYSTEM #2 :", elev[13], "al2.cvl");  break;
      case 'v' : civil("AERATED LAGOON SYSTEM #3 :", elev[14], "al3.cvl");  break;
      case 's' : civil("ACTIVATED SLUDGE SYSTEM :", elev[15], "as.cvl");  break;
      case 'c' : civil("SECONDARY CLARIFIER SYSTEM :",elev[16],"sec_cla.cvl"); break;
      case 'h' : civil("CHLORINATION SYSTEM :", elev[17], "chlo.cvl");  break;
      case 'd' : civil("ANAEROBIC DIGESTION SYSTEM :", elev[18], "an_d.cvl");  break;
      case 'b' : civil("SLUDGE-DRYING BEDS SYSTEM :", 0.0, "dry_bed.cvl");
    }
 }
 fclose(ft);
 civil_c = struc + earth;
 del_message(6); del_message(12); del_message(13); del_message(14); del_message(23);
 highvideo(); gotoxy (5, 6); cprintf ("Summary :"); normvideo();
 gotoxy (12, 10); printf ("Total structure cost          =  %9.0f     Baht.", struc);
 gotoxy (12, 12); printf ("Total earth work cost         =  %9.0f     Baht.", earth);
 gotoxy (49, 14); printf ("----------");
 gotoxy (12, 16); printf ("TOTAL CIVIL COST              =  %9.0f     BAHT.", civil_c);
 gotoxy (27, 23); printf ("Press any key to continue "); getch();
 if (toupper(out) == 'Y') {
    fprintf (fl,"\t\t TOTAL structure cost          =  %9.0f     baht.\n", struc);
    fprintf (fl,"\t\t TOTAL earth work cost         =  %9.0f     baht.\n", earth);
    fprintf (fl,"\t\t TOTAL CIVIL COST             =  %9.0f     BAHT.\n\n\n",civil_c);
```

```
}
if ((ft = fopen ("file.tmp","r")) == NULL) {
   printf ("error in reading file 'file.tmp' ");
   fclose(fl);
   exit(1);
}
if (toupper(out) == 'Y') {
   fprintf (fl,"\t\t\t\t -------------------------- \n");
   fprintf (fl,"\t\t\t\t  Equipment cost estimation \n");
   fprintf (fl,"\t\t\t\t -------------------------- \n\n");
}
while ((ch = getc(ft)) != EOF) {
   if (ferror (ft)) {
      printf ("error reading 'file.tmp' ");
      fclose(fl);
      exit(1);
   }
   switch (ch) {
      case 'g' : grit_et("GRIT CHAMBER SYSTEM :",'g');  break;
      case 'e' : grit_et("EQUALIZATION TANK SYSTEM :", 'e');  break;
      case 'p' : e_pri("PRIMARY SEDIMENTATION TANK SYSTEM :"); break;
      case 'c' : e_sec("SECONDARY CLARIFIER SYSTEM :");  break;
      case 's' : as_and("ACTIVATED SLUDGE SYSTEM :", 's');  break;
      case 'd' : as_and("ANAEROBIC SLUDGE-DIGESTION SYSTEM :", 'd');  break;
      case 'b' : e_bed("SLUDGE-DRYING BEDS SYSTEM :");  break;
      case 'h' : e_chlo("CHLORINATION SYSTEM :");  break;
      case 'l' : equip("AERATED LAGOON SYSTEM :", 'l');  break;
      case 't' : equip("AERATED LAGOON SYSTEM #2 :", 't');  break;
      case 'v' : equip("AERATED LAGOON SYSTEM #3 :", 'v');  break;
      case 'o' : equip("OZONATION SYSTEM :", 'o');  break;
      case 'u' : equip("UV DISINFECTION SYSTEM :", 'u');  break;
      case 'r' : equip("DEWATERING MACHINE SYSTEM :", 'r');
    }
 }
 fclose(ft);
 del_message(6); del_message(12); del_message(13); del_message(14); del_message(23);
 highvideo(); gotoxy (5, 6); cprintf ("Summary :"); normvideo();
 gotoxy (11, 13); printf ("TOTAL EQUIPMENT COST          =  %9.0f     BAHT.", equip_c);
 gotoxy (27, 23); printf ("Press any key to continue "); getch();
 if (toupper(out) == 'Y')
    fprintf (fl,"\t\t TOTAL EQUIPMENT COST          =  %9.0f     BAHT.\n\n\n", equip_c);
 grand(civil_c);
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define PI 3.141592654
extern int x1, y1, x2, y2;
extern char out;
extern FILE *f1;
extern float wall, base, degree, rc, excav, pond, struc, earth;
/*****************************************************************************/
civil (sys_name, elev, dat_name)
float elev;
char sys_name[], dat_name[];
{
  float val;
  char ch;
  float vol, slope, width, length, depth, w_depth, dia, no;
  float t_width, t_length, exc_depth, top_expand, b_area, s_area, exc_vol, exc_cost;
  float wall_conc, base_conc, conc, t_conc, conc_cost, wall_d, base_d, t_vol;
  FILE *fcivil;
  fcivil = fopen (dat_name,"r");
  fscanf (fcivil,"%f %f %f %f ", &vol, &slope, &width, &length);
  fscanf (fcivil,"%f %f %f %f", &depth, &w_depth, &dia, &no);
  fclose(fcivil);
  if (slope == 0) {
     do {
        do {
           frame (x1, y1, x2, y2);
           ins_frame();
           gotoxy (29, 3); printf (" CIVIL COST ESTIMATION ");
           highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
           if (dia == 0) {
              gotoxy (6, 8); printf ("    Tank width          = %7.2f   m.", width);
              gotoxy (6, 9); printf ("    Tank length         = %7.2f   m.",length);
           }
           else {
              gotoxy (6, 9); printf ("    Tank diameter       = %7.2f   m.", dia);
           }
           gotoxy (6, 10); printf ("    Tank depth          = %7.2f   m.",depth);
           gotoxy (6, 13); printf ("[1] WALL's thickness    = %7.0f   cm.", wall);
           gotoxy (6, 14); printf ("[2] FLOOR's thickness   = %7.0f   cm.",base);
           gotoxy (6, 15); printf("[3] Excavation's slope   = %7.0f   deg.",degree);
           gotoxy (6, 17); printf ("    unit cost :");
           gotoxy (6, 18); printf ("[4] Structure work      = %7.0f   ", rc);
                           printf ("baht/cu.m CONC.");
           gotoxy (6, 19); printf ("[5] Soil excavation w/back fill = %7.0f   ",excav);
                           printf ("baht/cu.m");
           gotoxy (10, 23);
           printf ("ENTER = accept parameters,  [1-5] for change parameter : ");
           ch = getche();
           if (ch >= '1' && ch <= '5') {
              del_message (23);
              gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
              scanf ("%f", &val);
              switch(ch) {
                 case '1' : wall = val;  break;
```

```
            case '2' : base = val;  break;
            case '3' : degree = val; break;
            case '4' : rc = val;   break;
            case '5' : excav = val;
          }
        }
      } while (ch != '\r');
      wall_d = wall/100;
      base_d = base/100;
      exc_depth = w_depth - elev + base_d;
      if (exc_depth < 0)
        exc_depth = 0;
      top_expand = 2 * exc_depth / tan (PI/180*degree);
      if (dia == 0) {
        wall_conc = (2*width + 2*length + 4*wall_d) * depth * wall_d;
        base_conc = (width + 2*wall_d + 0.2) * (length + 2*wall_d + 0.2) * base_d;
        t_width = width + 2*wall_d + 1;
        t_length = length + 2*wall_d + 1;
        b_area = t_width * t_length;
        s_area = (t_width + top_expand) * (t_length + top_expand);
      }
      else {
        wall_conc = PI * (dia + wall_d) * depth * wall_d;
        base_conc = PI / 4 * pow ((dia + 2*wall_d + 0.2), 2) * base_d;
        b_area = pow ((dia + 2*wall_d + 1), 2);
        s_area = pow ((dia + 2*wall_d + 1 + top_expand), 2);
      }
      conc = wall_conc + base_conc;
      t_conc = no * conc;
      exc_vol = (b_area + s_area + sqrt(b_area * s_area)) * exc_depth / 3;
      t_vol = no * exc_vol;
      conc_cost = t_conc * rc;
      exc_cost = t_vol * excav;
      do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (29, 3); printf (" CIVIL COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
        gotoxy (14,12); printf ("Structure cost       = %9.0f    baht.",conc_cost);
        gotoxy (14,14); printf ("Earth work cost      = %9.0f    baht.",exc_cost);
        gotoxy (10,23); printf ("Enter = accept result, [C] = change parameters");
        ch = getche();
      } while (ch != '\r' && toupper(ch) != 'C');
    } while (toupper(ch) == 'C');
    if (toupper(out) == 'Y') {
      fprintf (fl,"\t %s \n", sys_name);
      fprintf (fl,"\t\t Concrete WALL's thickness    = %9.0f   cm.\n", wall);
      fprintf (fl,"\t\t Concrete FLOOR's thickness   = %9.0f   cm.\n", base);
      fprintf (fl,"\t\t Soil excavation's slope      = %9.0f   degree.\n", degree);
      fprintf(fl,"\t\t Structure work             = %9.0f   baht/cu.m CONC.\n",rc);
      fprintf (fl,"\t\t Soil excavation w/back fill  = %9.0f  baht/cu.m\n", excav);
      fprintf (fl,"\t\t civil costs :\n");
      fprintf (fl,"\t\t Structure cost              = %9.0f  baht.\n", conc_cost);
      fprintf (fl,"\t\t Earth work cost             = %9.0f  baht.\n\n",exc_cost);
    }
}
```

```
  else {
     do {
        do {
           frame (x1, y1, x2, y2);
           ins_frame();
           gotoxy (29, 3); printf (" CIVIL COST ESTIMATION ");
           highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
           gotoxy (11, 13); printf ("Pond excavation work            = %7.0f", pond);
                            printf ("         baht/cu.m");
           gotoxy (9, 23); printf ("ENTER = accept parameters,  [C] for change ");
                           printf ("parameter's value : ");
           ch = getche();
           if (toupper(ch) == 'C') {
              del_message (23);
              gotoxy (27, 23); printf ("New parameter's value  =  ");
              scanf ("%f", &val);
              pond = val;
           }
        } while (ch != '\r');
        conc_cost = 0;
        exc_cost = vol * pond;
        do {
           frame (x1, y1, x2, y2);
           ins_frame();
           gotoxy (29, 3); printf (" CIVIL COST ESTIMATION ");
           highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
           gotoxy(14,13); printf ("Earth work cost     =   %9.0f     baht.", exc_cost);
           gotoxy(10,23); printf("Enter = accept result, [C] = change parameter : ");
           ch = getche();
        } while (ch != '\r' && toupper(ch) != 'C');
     } while (toupper(ch) == 'C');
     if (toupper(out) == 'Y') {
        fprintf (fl,"\t %s \n", sys_name);
        fprintf (fl,"\t\t Pond excavation work        =  %9.0f   baht/cu.m\n", pond);
        fprintf (fl,"\t\t Earth work cost            =  %9.0f   baht.\n\n", exc_cost);
     }
  }
  struc += conc_cost;
  earth += exc_cost;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *f1;
/*****************************************************************************/
e_a (sys_name)
char sys_name[];
{
  float value;
  char ch;
  float dia, depth, no;
  float brid, skim, scrap, weir, cont, excs, pump, box, oth, equip;
  FILE *fequip;
  fequip = fopen ("sec.eqp","r");
  fscanf (fequip,"%f %f %f", &dia, &depth, &no);
  fclose(fequip);
  brid = skim = scrap = weir = cont = excs = pump = box = oth = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
        gotoxy (12, 7); printf ("    Tank diameter           = %7.2f  m.", dia);
        gotoxy (12, 8); printf ("    Tank depth              = %7.2f  m.", depth);
        gotoxy (12, 9); printf ("    No. of tanks            = %4.0f  ", no);
        gotoxy (12, 11); printf ("    Cost per unit tank :- ");
        gotoxy (12, 12); printf ("[1] Access bridge           = %7.0f  baht.", brid);
        gotoxy (12, 13); printf ("[2] Scum skimmer            = %7.0f  baht.", skim);
        gotoxy (12, 14); printf ("[3] Sludge scrapers         = %7.0f  baht.",scrap);
        gotoxy (12, 15); printf ("[4] Adjustable weir         = %7.0f  baht.", weir);
        gotoxy (12, 16); printf ("[5] Sludge level control    = %7.0f  baht.", cont);
        gotoxy (12, 17); printf ("[6] Excess sludge control   = %7.0f  baht.", excs);
        gotoxy (12, 18); printf ("[7] Sludge pump             = %7.0f  baht.", pump);
        gotoxy (12, 19); printf ("[8] Split box               = %7.0f  baht.", box);
        gotoxy (12, 20); printf ("[9] Others                  = %7.0f  baht.", oth);
        gotoxy (10, 23);printf("ENTER = accept parameters, [1-9] change parameter :");
        ch = getche();
        if (ch >= '1' && ch <= '9') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &value);
           switch(ch) {
              case '1' : brid = value;
                         break;
              case '2' : skim = value;
                         break;
              case '3' : scrap = value;
                         break;
              case '4' : weir = value;
                         break;
              case '5' : cont = value;
                         break;
```

```
            case '6' : excs = value;
                       break;
            case '7' : pump = value;
                       break;
            case '8' : box = value;
                       break;
            case '9' : oth = value;
          }
        }
      } while (ch != '\r');
      equip = no * (brid + weir + skim + scrap + cont + excs + pump + box + oth);
      do {
         frame (x1, y1, x2, y2);
         ins_frame();
         gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
         highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
         gotoxy(13,13); printf ("Total equipment cost      =   %9.0f    baht.", equip);
         gotoxy(10,23); printf ("Enter = accept result,  [C] =  change parameters : ");
         ch = getche();
      } while (ch != '\r' && toupper(ch) != 'C');
   } while (toupper(ch) == 'C');
   if (toupper(out) == 'Y') {
      fprintf (fl,"\t %s \n", sys_name);
      fprintf (fl,"\t cost per unit tank :-\n");
      if (brid != 0)
         fprintf (fl,"\t\t Access bridge / guard rail    = %9.0f     baht.\n", brid);
      if (skim != 0)
         fprintf (fl,"\t\t Scum skimmer                  = %9.0f     baht.\n", skim);
      if (scrap != 0)
         fprintf (fl,"\t\t Sludge scrapers               = %9.0f     baht.\n", scrap);
      if (weir != 0)
         fprintf (fl,"\t\t Adjustable weir               = %9.0f     baht.\n", weir);
      if (cont != 0)
         fprintf (fl,"\t\t Sludge level control         = %9.0f     baht.\n", cont);
      if (excs != 0)
         fprintf (fl,"\t\t Excess sludge control        = %9.0f     baht.\n", excs);
      if (pump != 0)
         fprintf (fl,"\t\t Sludge pump                  = %9.0f     baht.\n", pump);
      if (box != 0)
         fprintf (fl,"\t\t Split box                    = %9.0f     baht.\n", pump);
      if (oth != 0)
         fprintf (fl,"\t\t Others                       = %9.0f     baht.\n", oth);
      fprintf (fl,"\t summary :- \n");
      fprintf (fl,"\t\t Total equipment cost          = %9.0f     baht.\n\n", equip);
   }
   equip_c += equip;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *f1;
/*****************************************************************************/
e_b (sys_name)
char sys_name[];
{
  float val;
  char ch;
  float ma, oth, equip;
  FILE *fequip;
  ma = oth = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
        gotoxy (13,12); printf ("[1] Ozone generation machine  = %7.0f   baht.",ma);
        gotoxy (13,14); printf ("[2] Others                    = %7.0f   baht.", oth);
        gotoxy (10, 23); printf("ENTER = accept parameters, [1,2] change parameter:");
        ch = getche();
        if (ch >= '1' && ch <= '2') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           switch(ch) {
              case '1' : ma = val;
                         break;
              case '2' : oth = val;
           }
        }
     } while (ch != '\r');
     equip = ma + oth;
     if (oth != 0) {
        del_message(8); del_message(12); del_message(14); del_message(23);
        do {
          gotoxy(13,13); printf("Total equipment cost    =   %9.0f   baht.", equip);
          gotoxy(10,23); printf("Enter = accept result, [C] = change parameters : ");
          ch = getche();
        } while (ch != '\r' && toupper(ch) != 'C');
     }
  } while (ch != '\r');
  if (toupper(out) == 'Y') {
     fprintf (f1,"\t %s\n", sys_name);
     fprintf (f1,"\t\t Ozone generation machine        = %9.0f     baht.\n", ma);
     fprintf (f1,"\t\t Others                         = %9.0f     baht.\n", oth);
     fprintf (f1,"\t\t Total equipment cost           = %9.0f     baht.\n", equip);
     fprintf (f1,"\n");
  }
  equip_c += equip;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *f1;
/*****************************************************************************/
e_c (sys_name, code)
char sys_name[], code;
{
  float val;
  char ch;
  float mb, mm, ma, oth, equip;
  FILE *fequip;
  if (code == '1') {
     fequip = fopen ("a1.eqp","r");
     fscanf (fequip,"%f", &mb);
     fclose(fequip);
  }
  if (code == 't') {
     fequip = fopen ("a12.eqp","r");
     fscanf (fequip,"%f", &mb);
     fclose(fequip);
  }
  if (code == 'v') {
     fequip = fopen ("a13.eqp","r");
     fscanf (fequip,"%f", &mb);
     fclose(fequip);
  }
  mm = ma = oth = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
        if (code == '1' || code == 't' || code == 'v') {
           gotoxy (13, 8); printf ("    Energy requirement        = %7.2f  hp.", mb);
           gotoxy (13,11); printf ("[1] Aeration & Mixing system  = %7.0f  baht.", ma);
           gotoxy (13,13); printf ("[2] Distribution box          = %7.0f  baht.", mm);
        }
        if (code == 'h') {
           gotoxy (13,11); printf ("[1] Chlorine feeder           = %7.0f  baht.", ma);
           gotoxy (13,13); printf ("[2] Chlorine safety equipment = %7.0f  baht.", mm);
        }
        gotoxy (13,15); printf ("[3] Others                    = %7.0f  baht.", oth);
        gotoxy (10, 23); printf("ENTER = accept parameters, [1-3] change parameter :");
        ch = getche();
        if (ch >= '1' && ch <= '3') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           switch(ch) {
              case '1' : ma = val;
                         break;
```

```
                case '2' : mm = val;
                           break;
                case '3' : oth = val;
              }
            }
          } while (ch != '\r');
          equip = ma + mm + oth;
          del_message(8); del_message(11); del_message(13); del_message(15);del_message(23);
          do {
            gotoxy(13,13); printf("Total equipment cost      =   %9.0f     baht.", equip);
            gotoxy(10,23); printf("Enter = accept result,   [C] =  change parameters : ");
            ch = getche();
          } while (ch != '\r' && toupper(ch) != 'C');
        } while (ch != '\r');
        if (toupper(out) == 'Y') {
          fprintf (fl,"\t %s\n", sys_name);
          if (code == 'l' || code == 't' || code == 'v') {
            if (ma != 0)
              fprintf (fl,"\t\t Aeration & Mixing system       =  %9.0f     baht.\n", ma);
            if (mm != 0)
              fprintf (fl,"\t\t Chlorine safety equipment      =  %9.0f     baht.\n", mm);
          }
          if (code == 'h') {
            if (ma != 0)
              fprintf (fl,"\t\t Chlorine feeder system         =  %9.0f     baht.\n", ma);
            if (mm != 0)
              fprintf (fl,"\t\t Chlorine safety equipment      =  %9.0f     baht.\n", mm);
          }
          if (oth != 0)
            fprintf (fl,"\t\t Others                          =  %9.0f     baht.\n", oth);
          fprintf (fl,"\t\t Total equipment cost          =  %9.0f     baht.\n\n", equip);
        }
        equip_c += equip;
      }
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *f1;
/*****************************************************************************/
e_d(sys, code)
char sys[], code;
{
  float value;
  char chr;
  float width, length, depth, no;
  float cover, mix, heat, oth, equip;
  FILE *fequip;
  if (code == 'b') {
     fequip = fopen ("bed.eqp","r");
     fscanf (fequip,"%f %f %f %f", &width, &length, &depth, &no);
     fclose(fequip);
  }
  cover = mix = heat = oth = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys); normvideo();
        if (code == 'b') {
           gotoxy (13, 8); printf ("    Width                    = %7.2f  m.", width);
           gotoxy (13, 9); printf ("    Length                   = %7.2f  m.",length);
           gotoxy (13, 10); printf ("    Media depth              = %7.2f  m.",depth);
           gotoxy (13, 11); printf ("    No. of tanks             = %4.0f  ", no);
           gotoxy (13, 14); printf ("    Cost per unit tank :- ");
           gotoxy (13, 15); printf ("[1] Underdrain system     = %7.0f  baht.",cover);
           gotoxy (13, 16); printf ("[2] Media                 = %7.0f  baht.", mix);
           gotoxy (13, 17); printf ("[3] Roof                  = %7.0f  baht.", heat);
        }
        if (code == 'u') {
           no = 1;
           gotoxy (13, 12); printf ("[1] UV generation machine = %7.0f  baht.",cover);
           gotoxy (13, 13); printf ("[2] UV lamp spares        = %7.0f  baht.", mix);
           gotoxy (13, 14); printf ("[3] UV dosage chambers    = %7.0f  baht.", heat);
        }
        gotoxy (13, 15); printf ("[4] Others                = %7.0f  baht.", oth);
        gotoxy (10, 23);printf("ENTER = accept parameters, [1-4] change parameter :");
        chr = getche();
        if (chr >= '1' && chr <= '4') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", chr);
           scanf ("%f", &value);
           switch(chr) {
              case '1' : cover = value;
                         break;
              case '2' : mix = value;
                         break;
```

```
                case '3' : heat = value;
                           break;
                case '4' : oth = value;
              }
            }
          } while (chr != '\r');
          equip = no * (cover + mix + heat + oth);
          do {
             frame (x1, y1, x2, y2);
             ins_frame();
             gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
             highvideo(); gotoxy (5, 6); cprintf ("%s", sys); normvideo();
             gotoxy(13,13); printf ("Total equipment cost      =   %9.0f   baht.", equip);
             gotoxy(10,23); printf ("Enter = accept result,  [C] = change parameters : ");
             chr = getche();
          } while (chr != '\r' && toupper(chr) != 'C');
        } while (toupper(chr) == 'C');
        if (toupper(out) == 'Y') {
          fprintf (fl,"\t %s \n", sys);
          if (code == 'u') {
             if (cover != 0)
                fprintf (fl,"\t\t UV generation machine       =  %9.0f     baht.\n", cover);
             if (mix != 0)
                fprintf (fl,"\t\t UV lamp spares              =  %9.0f     baht.\n", mix);
             if (heat != 0)
                fprintf (fl,"\t\t UV dosage chambers          =  %9.0f     baht.\n", heat);
          }
          if (code == 'b') {
             fprintf (fl,"\t cost per unit tank :-\n");
             if (cover != 0)
                fprintf (fl,"\t\t Underdrain system          =  %9.0f     baht.\n", cover);
             if (mix != 0)
                fprintf (fl,"\t\t Media                      =  %9.0f     baht.\n", mix);
             if (heat != 0)
                fprintf (fl,"\t\t Roof                       =  %9.0f     baht.\n", heat);
          }
          if (oth != 0)
             fprintf (fl,"\t\t Others                     =  %9.0f     baht.\n", oth);
          fprintf (fl,"\t summary :- \n");
          fprintf (fl,"\t\t Total equipment cost          %9.0f     baht.\n\n", equip);
        }
        equip_c += equip;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *f1;
/*****************************************************************************/
e_e(sys, code)
char sys[], code;
{
  float value;
  char ch;
  float width, length, depth, no, en;
  float slui, air, scre, scrap, oth, equip;
  FILE *fequip;
  if (code == 'g') {
     fequip = fopen ("grit.eqp","r");
     fscanf (fequip,"%f %f %f %f", &width, &length, &depth, &no);
     fclose(fequip);
  }
  if (code == 'e') {
     fequip = fopen ("equal.eqp","r");
     fscanf (fequip,"%f %f %f %f", &width, &length, &depth, &en);
     fclose(fequip);
  }
  air = slui = scre = scrap = oth = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys); normvideo();
        if (code == 'r') {
        no = 1;
        gotoxy (13, 12); printf ("[1] Sludge dewatering machine  = %7.0f baht.",slui);
        gotoxy (13, 13); printf ("[2] Chemical dosing equipments = %7.0f baht.", air);
        gotoxy (13, 14); printf ("[3] Cake transfer system       = %7.0f baht.", scre);
        gotoxy (13, 15); printf ("[4] Cake hauling system        = %7.0f baht.",scrap);
        gotoxy (13, 16); printf ("[5] Others                     = %7.0f baht.", oth);
        }
        if (code == 'g') {
        gotoxy (13, 8); printf ("    Width                      = %7.2f m.", width);
        gotoxy (13, 9); printf ("    Length                     = %7.2f m.",length);
        gotoxy (13, 10); printf ("    Depth                      = %7.2f m.",depth);
        gotoxy (13, 11); printf ("    No. of tanks               = %4.0f ", no);
        gotoxy (13, 13); printf ("    Cost per unit tank :- ");
        gotoxy (13, 14); printf ("[1] Sluice gates               = %7.0f baht.", slui);
        gotoxy (13, 15); printf ("[2] Air lift pumps             = %7.0f baht.", air);
        gotoxy (13, 16); printf ("[3] Screw pumps                = %7.0f baht.", scre);
        gotoxy (13, 17); printf ("[4] Grit scrapers              = %7.0f baht.",scrap);
        gotoxy (13, 18); printf ("[5] Others                     = %7.0f baht.", oth);
        }
        if (code == 'e') {
        no = 1;
        gotoxy (13, 8); printf ("    Width                      = %7.2f m.", width);
```

```
      gotoxy (13, 9); printf ("    Length                          = %7.2f m.",length);
      gotoxy (13, 10); printf ("    Depth                           = %7.2f m.",depth);
      gotoxy (13, 11); printf ("    Water volume                    = %7.2f cu.m", en);
      gotoxy (13, 14); printf ("[1] Aeration & Mixing system        = %7.0f baht.", slui);
      gotoxy (13, 15); printf ("[2] Scum skimmers                   = %7.0f baht.", air);
      gotoxy (13, 16); printf ("[3] Flow-measuring devices          = %7.0f baht.", scre);
      gotoxy (13, 17); printf ("[4] Pumps                           = %7.0f baht.",scrap);
      gotoxy (13, 18); printf ("[5] Others                          = %7.0f baht.", oth);
      }
      gotoxy (10, 23);printf("ENTER = accept parameters, [1-5] change parameter :");
      ch = getche();
      if (ch >= '1' && ch <= '5') {
         del_message (23);
         gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
         scanf ("%f", &value);
         switch(ch) {
            case '1' : slui = value;
                       break;
            case '2' : air = value;
                       break;
            case '3' : scre = value;
                       break;
            case '4' : scrap = value;
                       break;
            case '5' : oth = value;
         }
      }
   } while (ch != '\r');
   equip = no * (slui + air + scre + scrap + oth);
   do {
      frame (x1, y1, x2, y2);
      ins_frame();
      gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
      highvideo(); gotoxy (5, 6); cprintf ("%s", sys); normvideo();
      gotoxy(13,13); printf ("Total equipment cost     =   %9.0f    baht.", equip);
      gotoxy(10,23); printf ("Enter = accept result,  [C] = change parameters : ");
      ch = getche();
   } while (ch != '\r' && toupper(ch) != 'C');
} while (toupper(ch) == 'C');
if (toupper(out) == 'Y') {
   fprintf (fl,"\t %s \n", sys);
   if (code == 'r') {
      if (slui != 0)
         fprintf (fl,"\t\t Sludge dewatering machine     =  %9.0f  baht.\n", slui);
      if (air != 0)
         fprintf (fl,"\t\t Chemical dosing equipments    =  %9.0f  baht.\n", air);
      if (scre != 0)
         fprintf (fl,"\t\t Cake transfer system         =  %9.0f  baht.\n", scre);
      if (scrap != 0)
         fprintf (fl,"\t\t Cake hauling system          =  %9.0f  baht.\n", scrap);
   }
   if (code == 'g') {
      fprintf (fl,"\t cost per unit tank :-\n");
      if (slui != 0)
         fprintf (fl,"\t\t Sluice gates                 =  %9.0f  baht.\n", slui);
      if (air != 0)
```

```
            fprintf (fl,"\t\t Air lift pumps                = %9.0f  baht.\n", air);
         if (scre != 0)
            fprintf (fl,"\t\t Screw pumps                   = %9.0f  baht.\n", scre);
         if (scrap != 0)
            fprintf (fl,"\t\t Grit scrapers                 = %9.0f  baht.\n", scrap);
      }
      if (code == 'e') {
         if (slui != 0)
            fprintf (fl,"\t\t Aeration & Mixing system      = %9.0f  baht.\n", slui);
         if (air != 0)
            fprintf (fl,"\t\t Scum skimmers                 = %9.0f  baht.\n", air);
         if (scre != 0)
            fprintf (fl,"\t\t Flow-measuring devices        = %9.0f  baht.\n", scre);
         if (scrap != 0)
            fprintf (fl,"\t\t Pumps                         = %9.0f  baht.\n", scrap);
      }
      if (oth != 0)
         fprintf (fl,"\t\t Others                       = %9.0f  baht.\n", oth);
      fprintf (fl,"\t summary :- \n");
      fprintf (fl,"\t\t Total equipment cost          = %9.0f  baht.\n\n", equip);
   }
   equip_c += equip;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *f1;
/*****************************************************************************/
e_f (sys_name)
char sys_name[];
{
  float value;
  char ch;
  float width, length, depth, en;
  float brid, skim, scrap, weir, cont, pump, oth, equip;
  FILE *fequip;
  fequip = fopen ("as.eqp","r");
  fscanf (fequip,"%f %f %f %f", &width, &length, &depth, &en);
  fclose(fequip);
  brid = skim = scrap = weir = cont = pump = oth = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
        gotoxy (12, 8); printf ("    Width                          = %7.2f m.", width);
        gotoxy (12, 9); printf ("    Length                         = %7.2f m.",length);
        gotoxy (12, 10); printf ("    Depth                          = %7.2f m.", depth);
        gotoxy (12, 11); printf ("    Energy requirement             = %7.2f HP.", en);
        gotoxy (12, 14); printf ("[1] Access bridge / Guard rail = %7.0f baht.", brid);
        gotoxy (12, 15); printf ("[2] Aeration & Mixing system   = %7.0f baht.", skim);
        gotoxy (12, 16); printf ("[3] Defoaming system           = %7.0f baht.",scrap);
        gotoxy (12, 17); printf ("[4] split box                  = %7.0f baht.", weir);
        gotoxy (12, 18); printf ("[5] D.O. controller / meter    = %7.0f baht.", cont);
        gotoxy (12, 19); printf ("[6] Pump                       = %7.0f baht.", pump);
        gotoxy (12, 20); printf ("[7] Others                     = %7.0f baht.", oth);
        gotoxy (10, 23);printf("ENTER = accept parameters,  [1-7] change parameter :");
        ch = getche();
        if (ch >= '1' && ch <= '7') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &value);
           switch(ch) {
              case '1' : brid = value;
                         break;
              case '2' : skim = value;
                         break;
              case '3' : scrap = value;
                         break;
              case '4' : weir = value;
                         break;
              case '5' : cont = value;
                         break;
              case '6' : pump = value;
                         break;
```

```
            case '7' : oth = value;
          }
        }
      } while (ch != '\r');
      equip = brid + weir + skim + scrap + cont + pump + oth;
      do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
        gotoxy(13,13); printf ("Total equipment cost        =   %9.0f  baht.", equip);
        gotoxy(10,23); printf ("Enter = accept result,   [C] = change parameters : ");
        ch = getche();
      } while (ch != '\r' && toupper(ch) != 'C');
    } while (toupper(ch) == 'C');
    if (toupper(out) == 'Y') {
      fprintf (fl,"\t %s \n", sys_name);
      if (brid != 0)
        fprintf (fl,"\t\t Access bridge / Guard rail     =  %9.0f     baht.\n", brid);
      if (skim != 0)
        fprintf (fl,"\t\t Aeration & Mixing system       =  %9.0f     baht.\n", skim);
      if (scrap != 0)
        fprintf (fl,"\t\t Defoaming system               =  %9.0f     baht.\n", scrap);
      if (weir != 0)
        fprintf (fl,"\t\t Split box                      =  %9.0f     baht.\n", weir);
      if (cont != 0)
        fprintf (fl,"\t\t D.O. controller / meter        =  %9.0f     baht.\n", cont);
      if (pump != 0)
        fprintf (fl,"\t\t Pump                           =  %9.0f     baht.\n", pump);
      if (oth != 0)
        fprintf (fl,"\t\t Others                         =  %9.0f     baht.\n", oth);
      fprintf (fl,"\t summary :- \n");
      fprintf (fl,"\t\t Total equipment cost          =  %9.0f     baht.\n\n", equip);
    }
    equip_c += equip;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *fl;
/*****************************************************************************/
e_g (sys_name, code)
char sys_name[], code;
{
  float value;
  char ch;
  float width, length, dia, depth, no;
  float brid, skim, scrap, weir, cont, pump, box, oth, equip;
  FILE *fequip;
  if (code == 'p') {
     fequip = fopen.("pri.eqp","r");
     fscanf (fequip,"%f %f %f %f %f", &width, &length, &dia, &depth, &no);
     fclose(fequip);
  }
  if (code == 'd') {
     fequip = fopen ("an_d.eqp","r");
     fscanf (fequip,"%f %f %f", &dia, &depth, &no);
     fclose(fequip);
  }
  brid = skim = scrap = weir = cont = pump = box = oth = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
        highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
        if (dia == 0) {
           gotoxy (12, 7); printf ("    Tank width           = %7.2f    m.", width);
           gotoxy (12, 8); printf ("    Tank length          = %7.2f    m.",length);
        }
        else {
           gotoxy (12, 8); printf ("    Tank diameter        = %7.2f    m.", dia);
        }
        gotoxy (12, 9); printf ("    Tank depth           = %7.2f    m.", depth);
        gotoxy (12, 10); printf ("    No. of tanks         = %4.0f      ", no);
        gotoxy (12, 12); printf ("    Cost per unit tank :- ");
        if (code == 'p') {
           gotoxy (12, 13); printf ("[1] Access bridge        = %7.0f    baht.", brid);
           gotoxy (12, 14); printf ("[2] Scum skimmers        = %7.0f    baht.", skim);
           gotoxy (12, 15); printf ("[3] Sludge scrapers      = %7.0f    baht.",scrap);
           gotoxy (12, 16); printf ("[4] Adjustable weir      = %7.0f    baht.", weir);
           gotoxy (12, 17); printf ("[5] Sludge level control = %7.0f    baht.", cont);
           gotoxy (12, 18); printf ("[6] Sludge pump          = %7.0f    baht.", pump);
           gotoxy (12, 19); printf ("[7] Distribution box     = %7.0f    baht.", box);
        }
        if (code == 'd') {
           gotoxy (12, 13); printf ("[1] Digester covers      = %7.0f    baht.", brid);
           gotoxy (12, 14); printf ("[2] Mixing system        = %7.0f    baht.", skim);
           gotoxy (12, 15); printf ("[3] Heater system        = %7.0f    baht.",scrap);
```

```
            gotoxy (12, 16); printf ("[4] Alkalinity control   = %7.0f    baht.", weir);
            gotoxy (12, 17); printf ("[5] Gas meter            = %7.0f    baht.", cont);
            gotoxy (12, 18); printf ("[6] Gas burner           = %7.0f    baht.", pump);
            gotoxy (12, 19); printf ("[7] Split box            = %7.0f    baht.", box);
         }
         gotoxy (12, 20); printf ("[8] Others                = %7.0f    baht.", oth);
         gotoxy (10, 23);printf("ENTER = accept parameters,  [1-8] change parameter :");
         ch = getche();
         if (ch >= '1' && ch <= '8') {
            del_message (23);
            gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
            scanf ("%f", &value);
            switch(ch) {
               case '1' : brid = value;
                          break;
               case '2' : skim = value;
                          break;
               case '3' : scrap = value;
                          break;
               case '4' : weir = value;
                          break;
               case '5' : cont = value;
                          break;
               case '6' : pump = value;
                          break;
               case '7' : box = value;
                          break;
               case '8' : oth = value;
             }
          }
      } while (ch != '\r');
      equip = no * (brid + weir + skim + scrap + cont + pump + box + oth);
      do {
         frame (x1, y1, x2, y2);
         ins_frame();
         gotoxy (27, 3); printf (" EQUIPMENT COST ESTIMATION ");
         highvideo(); gotoxy (5, 6); cprintf ("%s", sys_name); normvideo();
         gotoxy(13,13); printf ("Total equipment cost      =   %9.0f    baht.", equip);
         gotoxy(10,23); printf ("Enter = accept result,   [C] = change parameters : ");
         ch = getche();
      } while (ch != '\r' && toupper(ch) != 'C');
   } while (toupper(ch) == 'C');
   if (toupper(out) == 'Y') {
      fprintf (fl,"\t %s \n", sys_name);
      fprintf (fl,"\t cost per unit tank :-\n");
      if (code == 'p') {
         if (brid != 0)
            fprintf (fl,"\t\t Access bridge / Guard rail    = %9.0f   baht.\n", brid);
         if (skim != 0)
            fprintf (fl,"\t\t Scum skimmers                = %9.0f   baht.\n", skim);
         if (scrap != 0)
            fprintf (fl,"\t\t Sludge scrapers              = %9.0f   baht.\n", scrap);
         if (weir != 0)
            fprintf (fl,"\t\t Adjustable weir              = %9.0f   baht.\n", weir);
         if (cont != 0)
            fprintf (fl,"\t\t Sludge level control        = %9.0f   baht.\n", cont);
```

```
        if (pump != 0)
           fprintf (fl,"\t\t Sludge pump                    = %9.0f   baht.\n", pump);
        if (box != 0)
           fprintf (fl,"\t\t Split box                      = %9.0f   baht.\n", box);
      }
      if (code == 'd') {
        if (brid != 0)
           fprintf (fl,"\t\t Digester cover                 = %9.0f   baht.\n", brid);
        if (skim != 0)
           fprintf (fl,"\t\t Mixing system                  = %9.0f   baht.\n", skim);
        if (scrap != 0)
           fprintf (fl,"\t\t Heater system                  = %9.0f   baht.\n", scrap);
        if (weir != 0)
           fprintf (fl,"\t\t Alkalinity control            = %9.0f   baht.\n", weir);
        if (cont != 0)
           fprintf (fl,"\t\t Gas meter                      = %9.0f   baht.\n", cont);
        if (pump != 0)
           fprintf (fl,"\t\t Gas burner                     = %9.0f   baht.\n", pump);
        if (box != 0)
           fprintf (fl,"\t\t Split box                      = %9.0f   baht.\n", box);
      }
      if (oth != 0)
         fprintf (fl,"\t\t Others                          = %9.0f   baht.\n", oth);
      fprintf (fl,"\t summary :- \n");
      fprintf (fl,"\t\t Total equipment cost          = %9.0f   baht.\n\n", equip);
   }
   equip_c += equip;
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
extern float equip_c;
extern int x1, y1, x2, y2;
extern char out;
extern FILE *fl;
/*****************************************************************************/
grand (civil_c)
float civil_c;
{
  float val;
  float arch, land, fenc, pipe, elec, faci, misc, over, tax, pro;
  float grand;
  char ch;
  arch = land = fenc = pipe = elec = faci = misc = over = tax = pro = 0;
  do {
     do {
        frame (x1, y1, x2, y2);
        ins_frame();
        gotoxy (25, 3); printf (" PRELIMINARY COST ESTIMATION ");
        gotoxy (13, 7); printf ("    Civil work                = %9.0f    baht.",civil_c);
        gotoxy (13, 8); printf ("    Equipment cost            = %9.0f    baht.",equip_c);
        gotoxy (13, 10); printf ("[1] Architectural work      = %9.0f    baht.", arch);
        gotoxy (13, 11); printf ("[2] Landscaping work        = %9.0f    baht.", land);
        gotoxy (13, 12); printf ("[3] Fence work              = %9.0f    baht.", fenc);
        gotoxy (13, 13); printf ("[4] Piping work             = %9.0f    baht.", pipe);
        gotoxy (13, 14); printf ("[5] Electric power          = %9.0f    baht.", elec);
        gotoxy (13, 15); printf ("[6] Facilities work         = %9.0f    baht.", faci);
        gotoxy (13, 16); printf ("[7] Miscellaneous cost      = %9.0f    baht.", misc);
        gotoxy (13, 17); printf ("[8] Overhead & Expenses     = %9.0f    baht.", over);
        gotoxy (13, 18); printf ("[9] Tax & Duty              = %9.0f    baht.", tax);
        gotoxy (13, 19); printf ("[0] Profit                  = %9.0f    baht.", pro);
        gotoxy (10, 23);printf("ENTER = accept parameters,[1-0] for change parameter");
        ch = getche();
        if (ch >= '0' && ch <= '9') {
           del_message (23);
           gotoxy (15, 23); printf ("Input new value of parameter no.[%c] : ", ch);
           scanf ("%f", &val);
           switch(ch) {
              case '1' : arch = val;  break;
              case '2' : land = val;  break;
              case '3' : fenc = val;  break;
              case '4' : pipe = val;  break;
              case '5' : elec = val;  break;
              case '6' : faci = val;  break;
              case '7' : misc = val;  break;
              case '8' : over = val;  break;
              case '9' : tax = val;   break;
              case '0' : pro = val;
           }
        }
     } while (ch != '\r');
     grand = civil_c+equip_c+arch+land+fenc+pipe+elec+faci+misc+over+tax+pro;
     do {
        frame (x1, y1, x2, y2);
```

```
      ins_frame();
      gotoxy (25, 3); printf (" PRELIMINARY COST ESTIMATION ");
      gotoxy(12,13); printf ("GRAND TOTAL COST        =   %10.0f      baht.", grand);
      gotoxy(10,23); printf("Enter = accept result,[C] = back to change parameters");
      ch = getche();
   } while (ch != '\r' && toupper(ch) != 'C');
} while (toupper(ch) == 'C');
if (toupper(out) == 'Y') {
   fprintf (fl,"\t\t\t      ---------------------------- \n");
   fprintf (fl,"\t\t\t       Preliminary cost estimation \n");
   fprintf (fl,"\t\t\t      ---------------------------- \n\n");
   fprintf (fl,"\t\t Civil work                          =  %9.0f     baht.\n", civil_c);
   fprintf (fl,"\t\t Equipment cost                      =  %9.0f     baht.\n", equip_c);
   if (arch != 0)
      fprintf (fl,"\t\t Architectural work                =  %9.0f     baht.\n", arch);
   if (land != 0)
      fprintf (fl,"\t\t Landscaping work                  =  %9.0f     baht.\n", land);
   if (fenc != 0)
      fprintf (fl,"\t\t Fence work                        =  %9.0f     baht.\n", fenc);
   if (pipe != 0)
      fprintf (fl,"\t\t Piping work                       =  %9.0f     baht.\n", pipe);
   if (elec != 0)
      fprintf (fl,"\t\t Electric power                    =  %9.0f     baht.\n", elec);
   if (faci != 0)
      fprintf (fl,"\t\t Facilities work                   =  %9.0f     baht.\n", faci);
   if (misc != 0)
      fprintf (fl,"\t\t Miscellaneous cost                =  %9.0f     baht.\n", misc);
   if (over != 0)
      fprintf (fl,"\t\t Overhead & Expenses              =  %9.0f     baht.\n", over);
   if (tax != 0)
      fprintf (fl,"\t\t Tax & Duty                       =  %9.0f     baht.\n", tax);
   if (pro != 0)
      fprintf (fl,"\t\t Profit                           =  %9.0f     baht.\n", pro);
   fprintf (fl,"\n");
   fprintf (fl,"\t\t GRAND TOTAL                        =  %9.0f     BAHT.\n", grand);
  }
}
```

ประวัติผู้เขียน

นายสมศักดิ์ สุรชัยพิทักษ์ เกิดวันที่ 28 มิถุนายน พ.ศ. 2508 สำเร็จการศึกษาปริญญาตรีวิศวกรรมศาสตรบัณฑิต (เกียรตินิยมอันดับสอง) สาขาวิศวกรรมโยธา คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าธนบุรี ในปีการศึกษา 2530 และเข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต ที่จุฬาลงกรณ์มหาวิทยาลัย เมื่อ พ.ศ. 2533