# ภาคผนวก ค

# Program Listing

```
# <<<<<<<<<<<<<< Makefile >>>>>>>>>>>>>>
instrld: main.o scre.o scrc.o scrs.o scrr.o scru.o fnrd.o fstr.o fedt.o \
    fcom.o fndx.o frep.o frph.o fthw.o
        cc main.o scre.o scrc.o scrs.o scrr.o scru.o fstr.o \
        /staff/mongkol/bin/libinx -lcurses -ltermcap -o instrld -g
main.o : main.c
        cc -c nain.c
scre.o : scre.c
        cc -c scre.c
scrc.o : scrc.c
        cc -c scrc.c
scrs.o : scrs.o
        cc -c scrs.c
scrr.o : scrr.c
        cc -c scrr.c
scru.o : scru.c
        cc -c scru.c
fnrd.o : fstr.c
        cc -c fnrd.c
fstr.o : fstr.c
        cc -c fstr.c
fedt.o : fedt.c edit.h
        cc -c fedt.c
fcom.o : fcom.c
        cc -c fcom.c
fndx.o : fndx.c
        cc -c fndx.c
frep.o : frep.c
        cc -c frep.c
frph.o : frph.c
        cc -c frph.c
fthw.o : fthw.c
        cc -c fthw.c
```

```
/*
* <<<<<<<<<<<<<<<  scr.h  >>>>>>>>>>>>>>>
*
* Escape sequence to set screen  (use for VT220 terminal)
*
*/

#define  ESC      "\033"
#define  CSI      "\033["

#define   entgrp( )   printf("%s(0", ESC)    /*   enter to graphicss mode  */
#define   extgrp( )   printf("%s(B", ESC)    /*   exit from graphics mode  */
#define   dubln( )    printf("%s#6", ESC)    /*   double-width line  */
#define   savcur( )   printf("%s7", ESC)     /*   save cursor  */
#define   rescur( )   printf("%s8", ESC)     /*   restore cursor  */
#define   clreol( )   printf("%sK", CSI)     /*   clear end of line  */
#define   clrscr( )   printf("%s2J", CSI)    /*   clear screen  */
#define   scr132( )   printf("%s3h", CSI)    /*   set screen to 132 column  */
#define   scr80( )    printf("%s3l", CSI)    /*   set screen to 80 column  */
#define   smtscl( )   printf("%s4h", CSI)    /*   smooth scrooling screen  */
#define   jmpscl( )   printf("%s4l", CSI)    /*   jump scrolling screen  */
#define   revscr( )   printf("%s5h", CSI)    /*   reverse screen */
#define   norscr( )   printf("%s?5l", CSI)   /*   normal screen  */
#define   curon( )    printf("%s?25h", CSI)  /*   text cursor on  */
#define   curoff( )   printf("%s?25l", CSI)  /*   text cursor off  */
#define   norchr( )   printf("%s0m", CSI)    /*   display normal character  */
#define   hghchr( )   printf("%s1m", CSI)    /*   display high-light charecter  */
#define   ulnchr( )   printf("%s4m", CSI)    /*   display underline character  */
#define   blkchr( )   printf("%s5m", CSI)    /*   display blinking character */
#define   revchr( )   printf("%s7m", CSI)    /*   display reverse character  */

#define   scrpos(x,y) printf("%s%d;%dH", CSI,(x),(y)   /*   line x,column y  */
```

```
/* <<<<<<<<<<<<<< edit.h >>>>>>>>>>>>>> */
short   posr0[ ] =  { 4, 4, 4},
        posc0[ ] =  {29,33,59},
        sz0[ ]   =  { 3, 3, 2},
        posr1[ ] =  { 9,12,12,12,15,18,18,18,18,19,19,19,19,20,20,20,20,21,21,21,21},
        posc1[ ] =  {18,13,32,50,23, 9,41,46,69, 9,41,46,69, 9,41,46,69, 9,41,46,69},
        sz1[ ]   =  {60, 2, 2, 2, 4, 7, 4, 4, 6, 7, 4, 4, 6, 7, 4, 4, 6, 7, 4, 4, 6},
        posr2[ ] =  { 9,12,12,13,13,14,14,15,15,16,16,17,17,18,18,19,19},
        posrt[ ] =  {13,16,19},
        posct[ ] =  {32,32,32},
        cont1[ ] [4] =  {{ 0, 1, 0, 0},{ 0, 4, 2, 1},{ 0, ,4 ,3 ,1}
                        ,{ 0, 4, 3, 2},{ 1, ,5 ,4 ,4},{ 4, 9, 6, 5}
                        ,{ 4,10, 7, 5},{ 4,11, 8, 6},{ 4,12, 8, 7}
                        ,{ 5,13,10, 9},{ 6, 14,11, 9}{ 7,15,12,10}
                        ,{ 8,16,12,11},{ 9,17,14,13},{10,18,15,13}
                        ,{11,19,16,14},{12,20,16,15},{13,17,18,17}
                        ,{14,18,19,17},{15,19,20,18},{16,20,20,19}
        cont2[ ][4]  =  {{ 0, 1, 0, 1},{ 0, 3, 2, 1},{ 0, 4, 2, 1}
                        ,{ 1, 5, 4, 3},{ 2, 6, 4, 3},{ 3, 7, 6, 5}
                        ,{ 4, 8, 6, 5},{ 5, 9, 8, 7},{ 6,10, 8, 7}
                        ,{ 7,11,10, 9},{ 8,12,10, 9},{ 9,13,12,11}
                        ,{13,15,16,15},{14,16,16,15}},
        contt[ ][4]  =  {{ 0, 1, 0, 0},{ 0, 2, 1, 1},{ 1, 2, 2, }},
        pinbh[ ] =  { 8,68,70,72},
        pinb1[ ] =  {11,  15,43,47,75,  22,51,55,81,
                          29,59,63,87,  36,67,71,93
        pinb2[ ] =  {10,  99,155,  106,158,  113,161,  120,164,
                          127,167,  134,170,  141,173,  148,176},
```

```
/* <<<<<<<<<<<<<< main.c >>>>>>>>>>>>>> */

/*
* main screen
*/

#include <curses.h>
#include "scr.h"

#define FSETUP "setup.dat"

int term,year,col132;
char prcom[50],prnm[30];

main( ) {
        FILE *f;
        char tmp[35];
        int ch;

        initscr( );
        system("reset");
        if ((f = fopen(FSETUP, "r")) != NULL) {
                fscanf(f, "%d",&term);
                fscanf(f, "%d", &year);
                fscanf(f, "%d", &col132);
                fgets(prnm,30,f);
                strcpy(prcom, "");
                strcpy(prcom,prnm);
                strcat(prcom, "report.dat");
                fclose(f);
        } else {
                term = 2;
                year = 2532;
                col132 = 0;
                strcpy(prnm, "lpr -h -Pdec");
                strcpy(prcom, "lpr -h -Pdec report.dat");
```

```
}
do {
        clrscr( );
        hghchr( );
        scrpos( 5,11);dubln( ); printf("<<<( MAIN MENU )>>>");
        norchr( );
        scrpos( 8,30); printf( "1.  EDIT DATA");
        scrpos(10,30); printf( "2.  COMPUTE TEACHING LOAD");
        scrpos(12,30); printf( "3.  SEARCH");
        scrpos(14,30); printf( "4.  REPORT");
        scrpos(16,30); printf( "5.  SETUP");
        scrpos(18,30); printf( "6.  EXIT");
        scrpos(24,58); printf( "SELECT  [6]");
        do {
                ch  =  rddg(24,66);
        }  while((ch < 0)&&(ch > 6));
        switch(ch)  {
                case  1  :  edit( );
                            break;
                case  2  :  load( );
                            break;
                case  3  :  scrh( );
                            break;
                case  4  :  rept( );
                            break;
                case  5  :  setup( );
        }
}  while  ((ch  !  =  6)&&(ch  !  =  0));
scrpos(24, 1);
system("stty -cbreak");
endwin( );
```

```
/* <<<<<<<<<<<<<<  scre.c  >>>>>>>>>>>>>> */

#include <stdio.h>
#include <curses.h>
#include "scr.h"
#define RETURN '\015'

extern short posr1[ ],posc1[ ],posr2[ ],posrc2[ ],posrt[ ],posct[ ];
int savf,savhf;

edit( )
{
        int ch;
        do {
                clrscr( );
                refresh( );
                scrpos(5,12);
                dubln( );
                hghchr( );
                printf("<<<  EDIT DATA  >>>");
                norchr( );
                scrpos( 9,27);
                printf("1.Edit Subject Data");
                scrpos(12,27);
                printf("2.Edit Teacher Data");
                scrpos(15,27);
                printf("3.EXIT");
                scrpos(22,50);
                printf("SELECT  [3]");
                do {
                        ch  =  rddg(22,58);
                } while ((ch  <  0) || (ch  >  3));
                clrscr( );
                refresh( );
                switch(ch) {
                        case  1  :  data_edit( );
                                    break;
                        case  2  :  tech_edit( );
```

```
                }
                scrpos(1,1);
        } while ((ch != 3) && (ch != 0));
}
data_edit( )
{
        FILE *fp[2],*fph[2];
        char ch,keyh[8],sect[3],key[10],buf[199],bufh[93];
        int scr;
        long ln,lnh;
        fopeninx("load", "r+",fp);
        fopeninx("hload", "r+",fph);
        page0( );
        page1( );
        ch = editp0(keyh,sect);
        while (ch == 'Y') {
        putchar('\007');
        scr = 1;
        savf = savhf = 0;
        strcpy(key, "");
        strcpy(key,keyh);
        strcat(key,sect);
        ln = freadinx(buf,key,fp);
        lnh = freadinx(bufh,keyh,fph);
        if (ln == 0L) {
                scrpos(24,48);
                printf("NEW RECORD. Exit (Y/N) ? [Y]");
                ch = rdyrn(24,75);
        } else {
                If (buf[178] == 'D') {
                        scrpos(24,44);
                        printf("DELETED RECORD. Exit (Y/N) ? [Y]");
                        ch = rdyrn(24,75);
                } else {
                        ch = 'N';
                }
```

```
}
if (ch == 'N') {
        if (ln == OL)
                sprintf (buf, "%9s%180s", key, "");
        if (lnh == OL)
                sprintf (bufh, "%7s%76s", keyh, "");
        showp11(bufh);
        showp12(buf);
        ch = gettnx( );
        while (ch != 'x') {
                if (ch == 'T') {
                        if (scr == 1) {
                                editp1(buf, bufh);
                        } else {
                                editp2(buf);
                        }
                } else {        /* ch == 'N' */
                        clrdw( );
                        if (csr == 1) {
                                page2( );
                                showp2(buf);
                                scr = 2;
                        } else {
                                page1( );
                                showp11(bufh);
                                showp12(buf);
                                chr = 1;
                        }
                }
                ch = gettnx( );
        }
        if (savf)
                if (ln == OL)
                        fwriteinx(buf, fp, "s");
                else
                        frewriteinx(buf, ln, fp);
```

```
                if (savhf)
                        if (lnh == OL)
                                fwriteinx(bufh, fph, "s");
                        else
                                frewriteinx(bufh, lnh, fph);
                }
                clrscr( );
                refresh( );
                page0( );
                page1( );
                ch = editp0(keyh,sect);
        }
        clrscr( );
        refresh( );
        fcloseinx(fph);
}
tech_edit( )
{
        FILE *ft[2];
        char buft[106], tcode[8], sterr[30];
        logn ln;
        int ch;
        fopeninx("lect", "r+", ft);
        paget( );
        strcpy(tcode, rdchr(7,9,22));
        while (strcmp(tcode, "")) {
                putchar('\007');
                if (!snum(7,tcode)) {
                        sprintf(sterr, ". . . CODE [%7s] error . . .", tcode);
                        status(sterr);
                        ch = 'Y';
                } else {
                        savf = 0;
                        ln = freadinx(buft, tcode, ft);
                        if (ln == OL) {
                                scrpos(24,48);
```

```
                        printf("NEW RECORD. EXIT (Y/N) ? [Y]");
                        ch = rdyrn(24,75);
                        scrpos(24,48);
                        clreol( );
                } else {
                        ch = 'N';
                }
        }
        if (ch == 'N') {
                if (ln == OL)
                        sprintf(buft, "%7s%88s\n", tcode, "");
                showpt(buft);
                editpt(buft);
        }
        if (savf) {
                if (ln == OL)
                        fwriteinx(buft,ft, "s");
                else
                        frewriteinx(buft,ln,ft);
        }
        clrscr( );
        refresh( );
        paget( );
        strcpy(tcode,rdchr(7,9,22));
    }
    fcloseinx(ft);
}

page0( )
{
    int i;
    scrpos( 1,16); dubln( ); printf("SUBJECT DATA");
    scrpos( 4,15); printf("Subject-Code [   -   ]");
    scrpos( 4,50); printf("Section  [   ]");
    scrpos( 6, 3);
    for  (i = 0; i < 77; i++)
        printf("-");
```

```
}
page1( )
{
        int  i;

        scrpos( 9, 4); printf("Subject Name [%60s] "," ");
        scrpos(12, 4); printf("Credit    [ ]  :");
        scrpos(12,23); printf("Lecture  [ ]");
        scrpos(15, 4); printf("Number of Student  [    ]");
        scrpos(18, 4); printf("Day");
        scrpos(18,35): printf("Time");
        scrpos(18,63); printf("Room");
        for  (i = 0;i  <  4;i++)  {
                scrpos(18+i, 8); printf("[       ]");
                scrpos(18+i,40); printf("[            ]");
                scrpos(18+i,68); printf("[       ]");
        }
}
page2( )
{
        int  i;

        scrpos( 9, 4); printf("Subject  Type  [ ]");
        scrpos(12, 4); printf("Teacher Code");
        for  (i = 0;i  <  8;i++)  {
                scrpos(12+i,23); printf("%1d. [       ]  {     ]%%" ,i+1);
        }
}
paget( )
{
        int  i;

        scrpos( 5,15);
        dubln( );
        printf("TEACHER DATA");
        scrpos( 9, 7); printf("Teacher Code  [       ]");
        scrpos(11, 3);
        for  (i = 0;i  <  77;i++);
```

```
                printf("-");
        scrpos(13, 7); printf("Department Code          [      ]");
        scrpos(16, 7); printf("Teacher Name in Thai    [%40s]", "");
        scrpos(19, 7); printf("                  in English [%35s]", "");
}
clrdw( )
{
        int  i;
        for  (i  =  0;i  <17;i++)  {
                scrpos( 7+i,3);
                clreol( );
        }
}
showp11(hbuf)
char  hbuf[ ];
{
        int  i;
        wrstr(posr1[0], posc1[0], strcut(hbuf, 8,67));       /* Subject Name  */
        for  (i  =  0; i  <  3;i++)                          /* Credit    */
                wrstr(posr1i+1], posc1[i+1], strcut(hbuf,68+(i*2), 69+(I*2)));
}
showp12(buf)
char  buf[ ];
{
        int  i,next;
        wrstr(posr1[4],posc1[4],strcut(buf,11,14));          /*  No.  of Stud.  */
        for  (i  =  0;i  <  4;i++)  {
                next  =  i * 4;
                wrstr(posr1[5+next], posc1[5+next]
                        ,strcut(buf,15+(i*7),21(i*7)));      /*  Day  */
                wrstr(posr1[6+next], posc[6+next]
                        ,strcut(buf,43+(i*8),46+(i*8)));     /*  Time In  */
                wrstr(posr1[7+next], posc1[7+next]
                        ,strcut(buf,47+(i*8),50+(i*8)));     /*  Time Out    */
                wrstr(posr1[8+next], posc1[8+next]
```

```
                    ,strcut(buf,75+(i*6),80+(i*6)));   /* Room */
        }
}
showp2(buf)
char buf[ ];
{
        int i;
        wrstr(posr2[0], posc2[0], strcut(buf,10,10);        /* Subject Type */
        for (i = 0;i <8;i++) {
            wrstr(posr2[1+(i*2)],posc2[1+(i*2)]
                    ,strcut(buf, 99+(i*7),105+(i*7)));  /* Teacher Code */
            wrstr(posr2[2+(i*2)],posc2[2+(i*2)]
                    ,strcut(buf,155+(i*3),157+(i*3)));  /* Percent */
}
showpt(buf)
char buf[ ];
{
        wrstr(posrt[0],posct[0],strcut(buf, 8,11));
        wrstr(posrt[1],posct[1],strcut(buf,12,51));
        wrstr(posrt[2],posct[2],strcut(buf,52,86));
}
gettnx( )
{
        int ch;
        scrpos(24,40);
        hghchr( );printf("T");
        norchr( );printf("his screen,");
        hghchr( );printf("N");
        norchr( );printf("ext screen,ne");
        hghchr( ); printf("X");
        norchr( );printf("t code  [X]");
        do {
            scrpos(24,75);
            ch = toupper(getch( ));
        } while ((ch != 'T') && (ch != 'N') && (ch != 'X')
```

```
        && (ch != RETURN) && (ch != ' '));
if (ch == RETURN)
        ch = 'X';
if (ch == ' ');
        ch = 'N';
scrpos(24,40);
clreol( );
return(ch);
```

```
/* <<<<<<<<<<<<< scrc.c >>>>>>>>>>>>> */

#include <curses.h>
#include "scr.h"

load( )
{
        int choc;

        clrscr( );
        scrpos( 7,10);
        dubln( );
        printf("COMPUTE TEACHING LOAD");
        norchr( );
        scrpos(14,20);
        printf("Start to Compute Teaching Load [Y/N] ? [Y]");
        choc = rdym(14,60);
        clrscr( );
        if ( choc == 'Y' ) {
                scrpos(12, 6);
                dubln( );
                printf("<<< Compute Teaching Load >>>");
                scrpos(15,31);
                printf(". . . Please wait . . .");

                refresh( );
                comdb( );
                scrpos(15,31);
                slreol( );
                clreol( );
                putchar('\007');
                scrpos(24,55);
                printf("!READY! Press any key . . .");
                getch( );
```

```
/* <<<<<<<<<<<<< scrs.c >>>>>>>>>>>>> */
#include <stdio.h>
#include <curses.h>
#include "scr.h"

#define FREPT "report.dat"

extern int col123;
char dinf[ ][5]  =  {"1240"
                    ,"2000", "2100", "2150", "2200", "2250", "2300", "2350", "2400"
                    ,"2500", "2600", "2650", "2700", "2800", "2900"
                    ,"3000", "3200", "3220", 3240", "3260", "3280", "3300", "3320"
                            ,"3340", "3360", "3380", "3400"
                    ,"3500", "3600", "3620", "3640", "3660", "3680", "3700", "3720"
                            ,"3740", "3760", "3780", "3800", "3820", "3960"
                    ,"4000", "4100", ":4160", "4200", "4240" , "4330"
                    ,"4500", "4600", "4650", "4700"
                    ,"5000", "5100", "5150", "5200", "5250", "5300", "5350", "5410"
                    ,"5500", "5600", "5650", "5700", "5750", "5800"
                    ,"6000", "6130", "6160", "6190", "6220", "6250", "6290", "6330"
                            ,"6370"
                    ,"6500", "6600", "6650", "6700", "6750", "6800", "6850"
                    ,"7000", "7010", "7100", "7200", "7310"
                    ,"9500"
                    ,"9600", "9650"};
int  rdyf,
      numdf[ ]  =  {  1, 8, 6,12,14, 6, 4, 8, 6, 9, 7, 5, 1, 2  },
      posdf[ ]  =  {  0, 1, 9,15,27,41,47,51,59,65,74,81,86,87  };

scrh( )
{
          int  choc;
          do  {
                    clrscr( );
                    refresh( );
                    scrpos(5,13);
                    dubln( );
                    hghchr( );
                    printf("<<<  SEARCH  >>>");
```

```
                norchr( );
                scrpos( 9,18);
                printf("1.  Subjects Taught by Each Lecturer");
                scrpos(11,18);
                printf("2.  Teaching Load (Based on Standard Formula )");
                scrpos(13,18);
                printf("3.  Teaching Load (Based on New Formula )");
                scrpos(15,18);
                printf("4.  Department Code");
                scrpos(17,18);
                printf("5.  EXIT");
                scrpos(22,58);
                printf("SELECT  [5]");
                do  {
                        choc  =  rddg(22,66);
                }  while  ((choc  <  0) || (choc  >  5));
                clrscr( );
                refresh( );
                switch(choc)  {
                        case  1  :  srhsb( );
                                    break;
                        case  2  :
                        case  3  :  mksrh(choc - 1);
                                    break;
                        case  4  :  swcdp( );
                }
        }  while  ((choc  !=  5)  &&  (choc  !=  0));
}
srhsb( )
{
        int  i;
        rdyf  =  0;
        clrscr( );
        scrpos(1,1);
        refresh( );
        sleep(1);
```

```
        scrpos(7,13);
        dubln( );
        hghchr( );
        printf("<<< SEARCH >>>" );
        norchr( );
        wrmid(10, "<<< Subjects Taught by Each Lecturer >>>");
        subj_t( );
        scrpos(15,30);
        clreol( );
        putchar('\007');
        if (rdyf) {
                clrscr( );
                refresh( );
                scr123( );
                prscr( );
                scr80( );
                refresh( );
                sleep(1);
        } else {
                status("NO DATA FOR THIS CODE");
        }
}

mksrh(type)
int type;
{
        int rang,nmfac;

        clrscr( );
        hghchr( );
        scrpos(4,13);
        dubln( );
        printf("<<< SEARCH >>>");
        if (type == 1)
                wrmid(7, "Teaching Load (Based on Standard Formula )");
        else
                wrmid(7, "Teaching Load (Based on New Formula )");
        norchr( );
```

```
rang = repch( );
while (rang != 5) {
        if (rang == 3) {
                clrscr( );
                refresh( );
                nmfac = facnum( );
        } else {
                nmfac = 1;      /* set for if statment */
        }
        if (nmfac != 0) {
                nmfac--;
                rdyf = 0;
                clrscr( );
                scrpos(7,13);
                dubln( );
                hghchr( );
                printf("<<< SEARCH >>>");
                norchr( );
                if (type == 1) {
                        scrpos(10,18);
                        printf("<<< Teaching Load");
                        printf("( Based on Standard Formula ) >>>");
                } else {
                        scrpos(10,21);
                        printf("<<< Teaching Load");
                        printf("( Based on New Formula) >>>");
                }
                switch(rang) {
                        case 1 : mkrep_t(type);
                                 break;
                        case 2 : mkrep_d(type);
                                 break;
                        case 3 : mkrep_f(type,nmfac);
                                 break;
                        case 4 : mkrep_u(type);
                                 break;
                }
```

```
                    scrpos(15,30);
                    clreol( );
                    putchar('\007');
                    if (rdyf) {
                            clrscr( );
                            refresh( );
                            scr123( );
                            prscr( );
                            scr80( );
                            refresh( );
                    } else {
                            status("NO DATA FOR THIS CODE");
                    }
            }
            clrscr( );
            hghchr( );
            scrpos(4,13);
            dubln( );
            printf("<<< SEARCH >>>");
            if (type == 1)
                    wrmid(7, "Teaching Load (Based on Standard Formula )");
            else
                    wrmid(7, "Teaching Load (Based on New Formula )");
            norchr( );
            rang = repch( );
    }
}

prscr( )
{
    FILE *fr;
    unsigned char buf[300];
    int i = 0;

    i = 0;
    fr = fopen(FREPT, "r");
    while (fgets(buf, 300, fr) != NULL) {
            if (i++ < 9)
```

```
                    If (col132)
                        printf("%33s", "");
                    else
                        printf("%28s", "");
                printf("%s", thvax(buf));
            }
            fclose(fr);
            printf("%104sPress any to conftinue. . . ","");
            getch( );
}
swcdp( )
{
            FILE *f[2];
            int nmfac,ch,i,j;
            char bufd[90], *thstr[4];
            fopeninx("dept", "r",f);
            thstr[0] = (char *) malloc(200);
            thstr[1] = (char *) malloc(200);
            thstr[2] = (char *) malloc(200);
            thstr[3] = (char *) malloc(200);
            do {
                clrscr( );
                refresh( );
                ch = facnum( );
                if (ch != 0) {
                    clrscr( );
                    wrmid(1, "<<< DEPARTMENT CODE >>>");
                    scrpos(3,1);
                    refresh( );
                    nmfac = ch - 1;
                    for (i = 0;i < numdf[nmfac]; i++) {
                        if ((i != 0) && ((i % 5) == 0)) {
                            scrpos(24,1);
                            revchr( );
                            printf("--MORE--");
                            getch( );
```

```
                        norchr( );
                        clrscr( );
                        wrmid(1, "<<< DEPARTMENT NAME >>>");
                        scrpos(3,1);
                        refresh( );
                    }
                    freadinx(bufd,dinf[posdf[nmfac] + i],f);
                    thaich(strct(bufd,10,61),thstr);
                    for (j = 0;j < 4;j++) {
                        if (j == 2)
                            printf("          %4s -"
                                ,dinf[posdf[nmfac]+i]);
                        else
                            printf("          ");
                        printf("%s", thvax(thstr[j]));
                        if (j != 3)
                            printf("\n");
                    }
                }
                scrpos(24,46);
                printf("Press any key to continue . . . ");
                getch( );
            }
        } while (ch != 0);
        free(thstr[0]);
        free(thstr[1]);
        free(thstr[2]);
        free(thstr[3]);
        fcloseinx(f);
}
```

```
/* <<<<<<<<<<<<< scrr.c >>>>>>>>>>>>> */
#include <stdio.h>
#include <curses.h>
#include "scr.h"

extern char prcom[ ];
int rdyf;

rept( )
{
        int choc;
        do {
                clrscr( );
                scrpos(5,13);
                dubln( );
                hghchr( );
                printf("<<< REPORT >>>");
                norchr( );
                scrpos(9,18);
                printf("1. Subjects Taught by Each Lecturer");
                scrpos(12,18);
                printf("2. Teaching Load (Based on Standard Formula )");
                scrpos(15,18);
                printf("3. Teaching Load (Based on New Formula )");
                scrpos(18,18);
                printf("4. EXIT");
                scrpos(22,50);
                printf("SELECT [4]");
                do {
                        choc = rddg(22,58);
                } while ((choc < 0) || (choc > 4));
                clrscr( );
                switch(choc) {
                        case 1 : repsb( );
                                 break;
                        case 2 :
                        case 3 : mkrep(choc - 1);
                }
```

```
        } while ((choc != 4) && (choc != 0));
}
repsb( )
{
        int i;
        rdyf = 0;
        clrscr( );
        scrpos(1,1);
        refresh( );
        sleep(1);
        scrpos(7,13);
        dubln( );
        hghchr( );
        printf("<<< REPORT >>>");
        norchr( );
        wrmid(10, "<<< Subjects Taught by Each Lecturer >>>");
        subj_t( );
        scrpos(15,30);
        clreol( );
        putchar('\007');
        if (rdyf) {
                system(prcom);
                scrpos(23,55);
                printf("!READY! Press any key . . .");
                getch( );
        } else {
                status("NO DATA FOR THIS CODE");
        }
}
mkrep(type)
int type;
{
        int rang,nmfac;
        clrscr( );
        hghchr( );
        scrpos(4,13);
```

```
dubln( );
printf("<<< REPORT >>>");
if (type == 1)
        wrmid(7, "Teaching Load  (Based on Standard Formula )");
else
        wrmid(7, "Teaching Load  (Based on New Formula )");
norchr( );
rang = repch( );
while (rang != 5) {
        if (rang == 3) {
                clrscr( );
                refresh();
                nmfac = facnum( );
        } else {
                nmfac = 1;
        }
        if (nmfac != 0) {
                nmfac--;
                rdyf =0;
                clrscr( );
                scrpos(7,13);
                dubln( );
                hghchr( );
                printf("<<< REPORT >>>");
                norchr( );
                if (type == 1) {
                        scrpos(10,18);
                        printf("<<< Teaching Load ");
                        printf("( Based on Standard Formula ) >>>");
                } else {
                        scrpos(10,21);
                        printf("<<< Teaching Load ");
                        printf("( Based on New Formula ) >>>");
                }
                switch(rang) {
                        case 1 : mkrep_t(type);
                                 break;
```

```
                case 2 : mkrep_d(type);
                         break;
                case 3 : mkrep_f(type,nmfac);
                         break;
                case 4 : mkrep_u(type);
                         break;
        }
        scrpos(15,30);
        clreol( );
        putchar('\007');
        if (rdyf) {
                system(prcom);
                scrpos(23,55);
                printf("!READY! Press any key . . .");
                getch( );
        } else {
                status("NO DATA FOR THIS CODE");
        }
}
clrscr( );
hghchr( );
scrpos(4,13);
dubln( );
printf("<<< REPORT >>>");
if (type == 1)
        wrmid(7, "Teaching Load (Based on Standard Formula )"):
else
        wrmid(7, Teaching Load (Based on New Formula )");
norchr( );
rang = repch( ):
}
```

```
/* <<<<<<<<<<<<<< scru.c >>>>>>>>>>>>>> */

#include <stdio.h>

#include <curses.h>

#include "scr.h"

#define FSETUP "setup.dat"

extern int termyear,col132;

extern char prcom[ ],prnm[ ];

setup( )

{

        FILE *f;

        char tmp[35];

        int ch;

        clrscr( );

        scrpos( 5,13);dubln( );hghchr( );

        printf("<<< SETUP >>>");

        norchr( );

        scrpos( 8,15); printf("1. TERM ( 1, 2 or 3)                    [%1d", term);

        scrpos(10,15); printf("2. YEAR                                  [%4d]", year);

        scrpos(12,15); printf("3. PRINTER COMMAND                       [%-25.25s]", prnm);

        scrpos(14,15); printf("4. PAPER COLUMN (1=80,2=132)             [%1d]", col132+1);

        scrpos(16,15); printf("5. SAVE SETUP DATA (Y/N) ?               [Y]");

        scrpos(18,15); printf("6. EXIT");

        do {

                scrpos(22,58); printf("Select [6]");

                refresh( );

                do {

                        ch = rddg(22,66);

                } while (ch > 6);

                scrpos(22,51);clreol( );

                switch(ch) {

                        case 1 : do {

                                        ch = rddg(8,46);

                                } while ((ch != 0) && ((ch < 1) || (ch > 3)));

                                if (ch != 0)

                                        term = ch;

                                scrpos(8,46);
```

```
                    printf("%1d", term);
                    break;
            case 2 : strcpy(tmp,rdchr(4,10,46));
                    if (atoi(tmp) > 1900)
                        year = atoi(tmp);
                    scrpos(10,46);
                    printf("%4d", year);
                    break;
            case 3 : strcpy(tmp,rdchr(25,12,46));
                    if (strcmp(tmp, "")) {
                        strcpy(prnm,tmp);
                        strcpy(prcom,prnm);
                        strcat(prco, "report.dat");
                    }
                    scrpos(12,46);
                    printf("%25s", prnm);
                    break;
            case 4 : do {
                        ch = rddg(14,46);
                    } while (ch > 2);
                    if (ch != 0);
                        col132 = ch - 1;
                    scrpos(14,46);
                    printf("%1d", col132 + 1);
                    break;
            case 5 : ch = rdyrn(16,46);
                    if (ch == 'Y') {
                        f = fopen(FSETUP, "w");
                        fprintf(f, "%d\n", term);
                        fprintf(f, "%d\n", year);
                        fprintf(f, "%-30s\n", prnm);
                        fprintf(f, "%d\n", col132);
                        fclose(f);
                    }
        }
    } while ((ch != 0) && (ch != 6));
}
```

```
/* <<<<<<<<<<<<< fnrd.c >>>>>>>>>>>>> */
/*
* this module use to read value
*/
#include <ctype.h>
#include <curses.h>
#include "scr.h"
#define todigit(ch)  ( (ch) - '0' )
#define RETURN       '\015'
#define BCKSPC       '\177'

int fnf;        /* external variable */

rdupp( )
{
        return(toupper(getch( )));
}

rddg(ln,cl)
int ln,cl;
{
        int ch;

        do {
                scrpos(ln,cl);
                ch = getch( );
        } while ((ch != RETURN) && (! isdigit(ch)));
        refresh( );
        return( (ch == RETURN) ? 0 : todigit(ch) );
}

rdalp( )
{
        int ch;

        ch = getch( );
        switch(ch) {
                case '\033' : ch = getch( );     /* beginning with ESC */
                              if (ch == '[') {/* function key */
                                      ch = getch( );
                                      switch (ch) {
```

```
                                        case 'A' : fnf = 0; break;
                                        case 'B' : fnf = 1; break;
                                        case 'C' : fnf = 2; break;
                                        case 'D' : fnf = 3; break;
                                }
                                ch = RETURN;
                        }
                        break;
                case '\004' : fnf = 4;          /* CTRL_D */
                        ch = RETURN;
                        break;
                case '\025' : fnf = 5;          /* CTRL_U */
                        ch = RETURN;
                        break;
                case '\030' : fnf = 6;          /* CTRL_X */
                        ch = RETURN;
                        break;
        }
        if ((ch < 32) && (ch != RETURN) && (fnf < 4)) {
                if (ch == '\001')
                        putchar('\016');
                hghchr( );
                putchar(ch + 'A' - 1);
                norchr( );
                if (ch == '\001')
                        putchar('\001');
        }
        return(ch);
}
char *rdchr(sz,ln,cl)
int sz,ln,cl;
{
        int i,j,thf = 0;
        char *str;

        fnf = -1;
        str = (char *) malloc(200);
```

```
i = 0;
scrpos(ln,cl);
str[0] = rdalp( );
if (str[0] == '\001')
        thf = 1;
if (str[0] != RETURN) {
        if (sz != 1) {
                if (str[0] == BCKSPC) {
                        printf("%*s",sz, " ");
                        j = 0;
                } esle {
                        printf("%*s",sz-1, " ");
                        j = 1;
                }
                while (j++ < sz)
                        printf("\b");
        }
        while((str[i] != RETURN)&&(i < sz-1)) {
                if ((i > 0) || (str[i] != BCKSPC))
                        i++;
                str[i] = rdalp( );
                if (str[i] = '\001') {
                        thf = 1;
                } else {
                        if (str[i] == '\016')
                                thf = 0;
                }
                if ((str[i] == BCKSPC)&&(i >0)) {
                        printf("\b \b");
                        str[i] = '\000';
                        i -=2;;
                } else {
                        if (str[i] == BCKSPC)
                                printf("\b[");
                }
                if (i != 0) {
                        if (str[i] != RETURN)
```

```
                        i++;
                    for (j=i;j<sz;j++)
                        str[j] = ' ';
                    if (thf)
                        str[sz-1] = '\016';
                    str[sz] = '\000';
                } esle {
                    if (thf)
                        str[0] = ' ';
                    if (str[i] != RETURN)
                        str[++i] = '\000';
                }
            }
            if (thf)
                putchar('\016');
            return((i==0)? " " : str);
}

char rdyrn(ln,cl)
int ln,cl;
{
    int ch;
    do {
        scrpos(ln,cl);
        ch = getch( );
    } while ((ch != 'n')&&(ch != 'y')&&(ch != 'N')&&(ch != 'Y')
        &&(ch != RETURN));
    if (ch == RETURN) ch = 'Y';
    return(islower(ch)? toupper(ch):ch);
}

status(str)
char str[ ];
{
    hghchr( );
    blkchr( );
    wrmid(22,str);
    scrpos(23,33);
```

```
        norchr( );
        printf("Press any key");
        getch( );
        scrpos(22,1); clreol( );
        scrpos(23,1); clreol( );
}
```

```
/* <<<<<<<<<<<< fstr.c >>>>>>>>>>> */

/*
* this module use to cut string from begin-col to end-col
*/
#include <ctype.h>
#include "scr.h"

#define CTRL_A  1
#define CTRL_N  14

char *strcut(buf,bgc,edc)
char buf[ ];
int bgc,edc;
{
        int i,sz;
        char tmps[512];

        sz = edc - bgc + 1;
        for (i = 0;i < sz;i++)
                tmps[i] = buf[bgc+i-1];
        tmps[sz] = '\000';
        return (tmps);
}

wrstr(ln,cl,str)
int ln,cl;
char str[ ];
{
        int i,sz;

        scrpos(ln,cl);
        sz = strlen(str);
        for (i = 0;i < sz;i++) {
                if (str[i] == '\016')
                        putchar(str[i]);
                if (str[i] < 32) {
                        hghchr( );
                        putchar(str[i] + 'A' - 1);
                        norchr( );
                }
```

```
                if (str[i] != '\016')
                        putchar(str[i]);
        }
}
isnum(num,str)
int num;
char str[ ];
{
        int i,flag = 1;
        for (i = 0;(i < num) && (flag);i++) {
                if (! isdigit(str[i])) {
                        flag = 0;
                }
        }
        return(flag);
}
wrmid(ln,str)
int ln;
char str[ ];
{
        int cl;
        cl = (80 - strlen(str)) / 2;
        scrpos(ln,cl);
        printf("%s", str);
}
/*
* This module use to change thai code from 8 bit to 7 bit
*/
unsigned char *thvax(str)
unsigned char str[ ];
{
        unsigned char rstr[300];
        int i,j,thf;
        i = j = thf = 0;
        while (str[i] != '\000') {
                if (str[i] > 128) {
```

```
                if (!thf) {
                        thf = 1;
                        rstr[j++] = CTRL_A;
                }
                if (str[i] != '\t')
                        rstr[j++] = str[i] - 128;
                else
                        rstr[j++] = str[i];
        } else {
                if (thf) {
                        thf = 0;
                        rstr[j++] = CTRL_N;
                }
                rstr[j++] = str[i];
        }
        i++;
}
rstr[j] = '\000';
return(rstr);
```

```
mkrep_f(tyep,num)
int type,num;
{
        FILE *fc[2],*fd[2],*fr;
        int headf = 0,inf = 1;
        char buf[BUFCM],codef[5],strtmp[50],coded[5],ocoded[5];
        float tval[10],dval[10],fval[10];

        scrpos(12,14);
        sprintf(strtmp, "FACULTY : %s",fname[num]);
        wrmid(12,strtmp);
        if (fopenndx("compu2", "r",fc) != NULL) {
                fopeninx("dept", "r",fd);
                wrmid(15, . . . Please wait . . . ");
                refresh( );
                fr = fopen(FREPT, "w");
                strcpy(codef,fackey[num]);
                fstndx(fc,atoi(codef),1);
                strcpy(ocoded, "");
                while ((inf) && (fgets(buf,BUFCM,fc[0]) != NULL)) {
                        strcpy(coded,strcut(buf,1,4));
                        if (num == facul(atoi(coded)) ) {
                                if (!headf) {
                                        headrep(fr,type);
                                        headf = 1;
                                        facrep(fr,fd,codef);
                                        setary(fval);
                                        setary(dval);
                                }
                                if (strcmp(ocoded,coded)) {
                                        if (strcmp(ocoded, ""))
                                                depsum(fr,fd,1,ocoded,dval);
                                        strcpy(ocoded,coded);
                                        setary(dval);
                                }
                                bodyval(buf,type,tval);
                                addary(dval,tval);
```

```
            }
            if ((flag == 'Y') && ((scf != 1) || (sef != 1))) {
                    strcpy(tsec,rdchr(sz0[2],posr0[2],posc0[2]));
                    if (((strcmp(tsec, " ")) || (sef == 0)) && (fnf != 3)) {
                            if (atoi(tsec) == 0) {
                                    strcpy(tsec, "01");
                            } else {
                                    tmp = atoi(tsec);
                                    if (! isnum(2,tsec)) {
                                            tsec[0] = (tmp/10) + '0';
                                            tsec[1] = (tmp%10) + '0';
                                            tsec[2] = '\000';
                                    }
                            }
                            sef = 1;
                            strcpy(sec,tsec);
                    }
                    wrstr(posr0[2],posc0[2],sec);
            }
    } while ((flag != 'N') && ((fnf == 3) || (scf == 0))
                    && ((scf != 1) || (sef != 1)));
    return(flag);
}

editp1(buf,bufh)
char buf[ ],bufh[ ];
{
    short i,bgc,enc,c;

    for (i = 0;i < 21;i++)
            strcpy(chgbuf1[i], " ");
    hghchr( );
    wrmid(24, "[ ^U - Update, ^D - Delete, ^X - Exit ]");
    norchr( );
    i = 0;
    do {
            strcpy(chgbuf1[i],rdchr(sz1[i],posr1pi],posc1[i]));
            if ((fnf >= 0) && (fnf <= 3))
```

```
                i = cont1[i][fnf];
        else
                i = ((i >= 20) ? 20 : i+1);
    } while ((fnf < 4) || (fnf > 6));
    if (fnf == 4) {
        savf = 1;
        buf[178] = 'D';
    } else {
        if (fnf == 5) {
            if (buf[178] == 'D') {
                savf = 1;
                buf[178] = ' ';
            }
            for (i = 0;i < 21;i++) {
                if (strcmp(chgbuf1[i], " )) {
                    if (i < 4) {
                        savhg = 1;
                        bgc = pinbh[i] - 1;
                        enc = pinbh[i] + sz1[i] - 2;
                        for (c = bgc;c <= enc;c++)
                            bufh[c] = chgbuf1[i][c-bgc];
                    } else {
                        savf = 1;
                        bgc = pinb1[i-4] - 1;
                        enc = pinb1[i-4] + sz1[i] - 2;
                        for (c = bgc;c <= enc;c++)
                            buf[c] = chgbuf1[i][c-bgc];
                    }
                }
            }
        }
    }
    scrpos(24,19);
    clreol( );
}
editp2(buf)
```

```
char  buf[ ];
{
        chort  i,bgc,enc,c;
        char  tmp[61];
        for  (i  =  0;i  <  17;i++)
                strcpy(chgbuf2[i], " ");
        ghgchr( );
        wrmid(24, "[ ^U - Update , ^D - Delete , ^X - eXit ]");
        norchr( );
        i  =  0;
        do  {
                strcpy(chgbuf2[i],rdchr(sz2[i],posr2[i],posc2[i]));
                if  ((i != 0 ) && ((i % 2)  ==  0)&&
                        (strcmp(chgbuf2[i], " "))&&(atoi(chgbuf2[i])  !=  0)) {
                                sprintf(tmp, "%03d",atoi(chgbuf2[i]));
                                strcpy(chgbuf2[i],tmp);
                                wrstr(posr2[i],posc2[i],tmp);
                }
                if  ((fnf  >=  0)  &&  (fnf  <=  3))
                        i  =  conft2[i][fnf];
                else
                        i  =  ((i  >=  16)  ?  16  :  i+1);
        }  while  ((fnf  <  4)  ||  (fnf  >  6));
        if  (fnf  ==  4)  {
                savf  =  1;
                buf[178]  =  'D';
        }  else  {
                if  (fnf  ==  5)  {
                        if  (buf[178]  ==  'D')  {
                                savf  =  1;
                                buf[178]  =  ' ';
                        }
                        for  (i  =0;i  <  17;i++)  {
                                if  (strcmp(chgbuf2[i], " "))  {
                                        savf  =  1;
                                        bgc  =  pinb2[i]  -  1;
```

```
                                        enc = pinb2[i] + sz2[i] - 2:
                                        for (c = bgc;c <= enc;c++)
                                                buf[c] = chgbuf2[i][c-bgc];
                                        if (i == 0)
                                                buf[bgc] = toupper(buf[bgc]);
                                }
                        }
                }
        }
        scrpos(24,19);
        clreol( );
}

editpt(buf)
char buf[ ];
{
        chort i,bgc,enc,c:
        for (i = 0;i < 17;i++)
                strcpy((chgbuft[i], " ");
        hghchr( );
        wrmid(24, "[ ^U - Update, ^X - eXit ]");
        norchr( );
        i = 0;
        do {
                strcpy(chgbuft[i],rdchr(szt[i],posrt[i],posct[i]));
                if ((fnf >= 0) && (fnf <= 3))
                        i = contt[i][fnf];
                else
                        i = ((i >= 2) ? 2 : I+1);
        } while ((fnf < 4) || (fnf > 6));
        if (fnf == 5) {
                for (i = 0;i < 3;i++) {
                        if (strcmp(chgbuft[i], " ")) {
                                savf = 1;
                                bgc = pinbt[i] - 1;
                                enc = pinbt[i] + szt[i] - 2;
                                for (c = bgc;c <= enc;c++)
```

```
                        buf[c] = chgbuft[i][c=bgc];
                    if (i == 0)
                        buf[bgc] = toupper(buf[bgc]);
                }
            }
        }
        scrpos(24,19);
        clreol( );
}
```

```
/* <<<<<<<<<<<<< fcom.c >>>>>>>>>>>>> */

#include <stdio.h>
#define  BLK7  "      "
#define  BLK10  "      \n"
#define  RECSZ  42   /* record size of file "compu"  */
#define  RPTSZ  69   /* record size of file "compu2"  */
#define  MAXBF  3000
#define  LGSTD  70
#define  SDSTD  35
#define  MXCRD  2.0

comdb( )
{
      FILE  *fi,*fo,*fl[2];
      int  i,flgi,nrec;
      char buf[199],bufc[52],tech[8],hour[5];
      char buf1[106];
      char  *sbuf;       /*  Sort buffer  */

      fi  =  fopen("load", "r");
      fopeninx("lect", "r",fl);
      fo  =  fopen("compu" , "w");
      nrec  =  0;
      sbuf  =  (char *)  calloc(MAXBF,RECSZ);
      strcpy(sbuf, "");
      while(fgets(buf,199,fi)  !=  NULL)  {
            if  (buf[178]  !=  'D')  {
                  flgi  =  1;
                  for  (I  =  0;(I  <8)&&(flgi);i++)  {
                        strcpy(tech,strcut(buf,99+(i*7),105+(i*7)));
                        if  (strcmp(tech,BLK7))  {
                              if (I  ==  0)
                                    sprintf(hour, "%04d"
                                          ,comtm(strcut(buf,15,74)));
                              strcpy(bufc, "");   /* clear new record  */
                              if  (freadinx(bufl,tech,fl)  !=  OL)
                                    strcpy(bufc,strcut(buf1,8,11));
```

```
                                        else
                                                strcpy(bufc, "0000");
                                        strcat(bufc,tech);
                                        strcat(bufc,strcut(buf,1,14));
                                        strcat(bufc
                                                ,strcut(buf,155+(i*3),157+(i*3)));
                                        strcat(bufc,hour);
                                        strcat(bufc,BLK10);
                                        strcat(sbuf,bufc);
                                        nrec++;
                                } else {
                                        flgi = 0;
                                }
                        }
                }
        }
        qsort(sbuf,nrec,RECSZ,strcmp);
        fputs(sbuf,fo);
        fcloseinx(fl);
        fclose(fi);
        fclose(fo);
        mkndx("compu",RECSZ,7,4);
        mkfrpt(sbuf,nrec);
        mkndx("compu2",RPTSZ,4,0);
}
mkfrpt(buf,num)         /*  make main report file  */
char  *buf;
int num;
{
        FILE  *fr,*fph[2];
        char stype,tcod[12],otcod[12],ssub[8],ossub[8];
        char stnm[5],perc[4],hour[5],bufh[93];
        int i,j,dcrf;
        float lecu,labu,lecg,labg,ward,semi,prju,prjg,thss,crdn,crds;
        float nums,pert,hurt;
        float cbth,clec,hlab,tmp,bons;
```

```
fr = fopen("compu2" , "w");
fopeninx("hload" , "r",fph);
lecu = labu = lecg = labg = ward = semi = 0.0;
prju = prjg = thss = crdn = crds = 0.0;
strcpy(otcod, "");
for (i = 0;i <num;i++) {
        for (j = 0;j < 11;j++)   /* office code + teacher code */
                tcod[j] = *(buf + (i * RECSZ) + j);
        tcod[11] = '\000';
        for (j = 0;j < 7;j++)    /* subject + section */
                ssub[j] = *(buf + (i * RECSZ) + 11 + j);
        ssub[7] = '\000';
        stype = *(buf + (i * RECSZ) + 20);
        for (j = 0;j < 4;j++)    /* number of student */
                stnm[j] = *(buf + (i * RECSZ) + 21 + j);
        for (j = 0;j < 3;j++)    /* teaching percent */
                perc[j] = *(buf + (i * RECSZ) + 25 + j);
        for (j = 0;j < 4;j++)    /* teaching hour */
                hour[j] = *(buf + (i * REXSZ) + 28 + j);
        nums = (float) atoi(stnm);
        pert = (float) atoi(perc) / 100.0;
        hurt = (float) atoi(hour) / 100.0;
        if (strcmp(tcod,otcod)) {      /* if tcod is changed */
                if (strcmp(otcod, "")) {
                        fprintf(fr, "%11s%05d%05d%05d%05d%05d%05d",
                                otcod,
                                (int)(lecu*1000),(int)(labu*1000);
                                (int)(lecg*1000),(int)(labg*10000);
                                (int)(ward*1000),(int)(semi*1000);
                        fprintf(fr, "%02d%02d%02d%",
                                (int)prju,(int)prjg,(int)thss);
                        fprintf(fr, "%06d%06d
                                (int)(crdn*1000),(int)(crds*1000));
                }
                strcpy(otcod,tcod);
                strcpy(ossub, "");
```

```
        lecu + labu = lecg = labg = ward = semi = 0.0;
        prju = prjg = thss = crdn = crds = 0.0;
}
if (strcmp(ssub,ossub)) {
        strcpy(ossub,ssub);
        dcrf = 0;
        freadinx(bufh,ssub,fph);
        cbth = (float) atoi(strcut(bufh,68,69));
        clec = (float) atoi(strcut(bufh,70,71));
        hlab = ((hurt != 0) ? (hurt-clec);
                        (float) atoi(strcut(bufh,72,73)));
} else {
        dcfr = 1;
}
if (nums >= LGSTD) {
        bons = (nums - LGSTD) / SDSTD * 0.5;
        if (bons > MXCRD;
} else {
        bons = 0.0;
}
switch (stype) {
        case 'U' : lecu += clec * pert;
                   labu += (hlab / 2.0) * pert;
                   crdn += (clec + (hlab /2.0)) * pert;
                   crds += ((clec + bons) + (hlab / 2.0)) * pert
                           * ((dcrf) ? (5.0 / 6.0) : 1.0);
                   break;
        case 'G' : clec =  clec * 1.5;
                   lecg += clec * pert;
                   labg += (hlab / 2.0) * pert;
                   crdn += (clec + (hlab /2.0)) * pert;
                   crds += ((clec + bons) + (hlab / 2.0)) * pert
                           * ((dcrf) ? (5.0 / 6.0) : 1.0);
                   break;
        case 'W' : ward += (hurt / 3.0) * pert;
                   crdn += (hurt / 3.0) * pert;
```

```
                crds += (hurt / 3.0) * pert;
                break;
case 'S' : semi += cbth * pert;
                crdn += cbth * pert;
                crds += cbth * pert;
                break;
case 'P' : pert = pert * 100.0;
                prju += pert;
                crdn += cbth + ((pert - 1.0) / 2.0);
                crds += pert * 2;
                break:
case 'T' : pert = pert * 100.0;
                thss += pert;
                if (cbth < 11) {
                        switch ((int)pert) {
                                case 0 : tmp = 0.0;
                                        break;
                                case 1 : tmp = 4,0;
                                        break;
                                case 2 : tmp = 5,5;
                                        break;
                                default : tmp = 6.0;
                        }
                } else {
                        if (cbth > 15) {
                                switch ((int)pert) {
                                        case 0 : tmp = 0.0;
                                                break;
                                        case 1 : tmp = 2.5;
                                                break;
                                        case 2 : tmp = 3.0;
                                                break;
                                        default : tmp = 3.5;
                                }
                        } else {
                                switch ((int)pert) {
```

```
                                        case 0 : tmp = 0.0;
                                                 break;
                                        case 1 : tmp = 3.0;
                                                 break;
                                        case 0 : tmp = 4.0;
                                                 break;
                                        default : tmp = 4.5;
                                    }
                                }
                            }
                            crdn += tmp;
                            crds += tmp;
                    }
            }
            if (num != 0) {
                    fprintf(fr, "%11s%05d%05d%05d%05d%05d%05d",otcod,
                            (int)(lecu*1000),(int(labu*1000);
                            (int)(lecg*1000),(int(labg*1000);
                            (int)(ward*1000),(int(semi*1000));
                    fprintf(fr, "%02d%02d%02d",
                            (int)prju,(int)prjg,(int)thss);
                    fprintf(fr, "%06d%06d
                            (int)(crdn*1000),(int)(crds*1000);
            }
            fcloseinx(fph);
            fclose(fr);
}
comt(str)
char str[ ];
{
            int  i,flag=i,nday=0;
            char day[8];
            for (i = 0;(i < 4) && (flag);i++) {
                    strcpy(day,strcut(str,1+(i*7),7+(i*7)));
                    if (strcmp(day,BLK7)) {
                            nday += fndy(day) * fntm(strcut(str,29+(i*8),36+(i*8)));
```

```
                } else {
                        flag = 0;
                }
        }
        return(nday);
}
fndy(sdy)
char sdy[ ];
{
        int i,flag=1,cou=0;
        for (i = 0;(i < 8) && (flag);i++) {
                switch (sdy[i]) {
                case 'S' :
                case 'M' :
                case 'T' :
                case 'W' :
                case 'F' : cou++;
                           break;
                case ' ' : flag = 0;
                           break;
                }
        }
        return(cou);
}
fntm(stm)
char stm[ ];
{
        int ou,sttm,sptm;
        sttm = htoi(strcut(stm,1,4));
        sptm = htoi(strcut(stm,5,8));
        cou  = sptm - sttm;
        return( ((cou < 0) ? ) : cou) );
}
/*
* This module use to change hour&minute to decimal radio
```

```
        * i.e. 13.30 to 1350 or 11.15 to 1125
*/
htoi(str)
char str[ ];
{
        int val;

        val = atoi(strcut(str,1,2))*100
                + (int) (atoi(strcut(str,3,4)) / 3.0 * 5.0);
        return(val);
}
```

```
/* <<<<<<<<<<<<<< fndx.c >>>>>>>>>>>>>> */

#include <stdio.h>

mkndx(name,rsz,ksz,pos)
char name[ ];
int rsz,ksz,pos;
{
        FILE *fi,*fx;
        char ndxnam[20],buf[200],key[20],okey[20];
        int ndx;

        strcpy(ndxnam,name);
        strcat(ndxnam, ".ndx");
        fi = fopen(name, "r");
        fx = fopen(ndxnam, "w");
        strcpy(okey, "");
        ndx = 0;
        while (fgets(buf,rsz+10,fi) != NULL) {
                strcpy(dey,strcut(buf,1+pos,ksz+pos));
                if (strcmp(key,okey)) {
                        strcpy(okey,key);
                        fprnitf(fx, "%*s %06d\n",ksz,okey,ndx);
                }
                ndx += rsz;
        }
        fclose(fi);
        fclose(fx);
}

fopenndx(name,type,fp)
char name[ ],type[ ];
FILE *fp[ ];
{
        char nmndx[30];

        strcpy(nmndx,name);
        strcat(nmndx, ".ndx");
        fp[0] = fopen(name,type);
        fp[1] = fopen(nmndx,type);
```

```
        return((fp[0] == NULL) || (fp[1] == NULL)) ? NULL : 1);
}
fstndx(fp,code,flg)          /* fseekndx */
FILE *fp[ ];
int code,flg;
{
        int rdcd,ndx = -1,nmfac;

        nmfac = facul(code);
        do {
                fscanf(fp[1], "%d%d",2rdcd,&ndx);
                if (flg)
                        if (nmfac == facul(rdce))
        } while ((!feof(fp[1])) && (rdcd != code));
        if (feof(fp[1]))
                ndx = -1;
        }
        fseek(fp[0],(long ndx,0);
        return((ndx == -1) ? 0 : 1);
}
facul(num)
int num;
{
        int code;
        if (num < 2000)
                code = 1;
        else if (num < 2500)    /* engineer */
                code = 1;
        else if (num < 3000)    /* educate */
                code = 2;
        else if (num < 3500)    /* science */
                code = 3;
        else if (num < 4000)    /* medical */
                code = 4;
        else if (num < 4500)    /* human nd social */
                code = 5;
        else if (num < 5000)    /* management science */
```

```
        code = 6;
else if (num < 5500)    /* natural resources */
        code = 7;
else if (num < 6000)    /* famarcy */
        code = 8;
else if (num < 6500)    /* nurses */
        code = 9;
else if (num < 7000)    /* dentist */
        code = 10;
else if (num < 7500)    /* science and technology */
        code = 11;
else if ((num >= 9500) && (num < 9600))        /* graduate */
        code = 12;
else if (num >= 9600)   /* Phuket college */
        code = 13;
else
        code = 0;
return(code);
```

```
/* <<<<<<<<<<<<< frep.c >>>>>>>>>>>>> */

#include <stdio.h>
#include <curses.h>
#include "scr.h"

#define BUFCM  79              /* compu2 file */
#define COMBF  52              /* compu file */
#define FREPT  "report.dat"            /* report file-name */
#define ON   1
#define OFF  0
#define COND_CH(FP,STATUS)          fprintf((FP), "%c",((STATUS) ? 15 : 18))
#define ENLG_CH(FP,STATUS)          fprintf((FP), "\033W%C",(STATUS))
#define NORMFEED(FP)

extern int col132;
extern char fackey[ ][5];
extern char fname[ ][30];
int rdyf;

subj_t( )
{
        FILE *fc[2],*fh[2],*fr;
        int cmpf = 1;
        char bufc[COMBF],bufh[93];
        char codet[8],codes[8],sterr[30];

        scrpos(12,31);
        printf("Teacher [       ]");
        strcpy(codet,rdchr(7,12,40));
        if (! isnum(7,codet)) {
                sprintf(sterr, ". . . CODE [%7s] error > . .",codet);
                status(sterr);
                return(10);
        }
        scrpos(12,12);
        clreol( );
        dubln( );
        printf("Teacher : %7s",codet);
        if (fopenndx("compu", "r",fc) != NULL) {
```

```
wrmid(15, ". . . Please wait > . . .");
refresh( );
fopeninx("hload", "r",fh);
fr = fopen(FREPT, "w");
if (fstndx(fc,atoi(codet),0)) {
        rdyf = 1;
        hdsbrp(fr,codet);
        while ((cmpf >= 0)&&(fgets(bufc,COMBF,fc[0]) != NULL)) {
                cmpf = strcmp(codet,strcut(bufc,5,11));
                if (cmpf == 0) {
                        strcpy(codes, "");
                        strcpy(codes,strcut(bufc,12,18));
                        fprintf(fr, "    %7s",codes);
                        if (freadinx(bufh,codes,fh) == OL)
                                sprintf(bufh, "%7s%75s\n",
                                        codes,, " ");
                        fprintf(fr, "    %60s",
                                strcut(bufh,8,67));
                        fprintf(fr, "    %2s",
                                strcut(nufc,19,20));
                        fprintf(fr, "    %2d(%2d-%2d)",
                                atoi(strcut(bufh,68,69));
                                atoi(strcut(bufh,70,71));
                                atoi(strcut(bufh,72,73)));
                        fprintf(fr, "      %1s",
                                strcut(bufc,21,21));
                        fprintf(fr, "                %3d",
                                atoi(strcut(bufc,26,28)));
                        fprintf(fr,                 %4d\n",
                                atoi(strcut(bufc,22,25)));
                }
        }
        lnrep(fr);
        if(!col132)
                COND_CH(fr,OFF);
        fclose(fr);
```

```
                    fcloseinx(fh);
                    fclosendx(fc);
                }
        }
}
mkrep_t(type)
int type;
{
        FILE *fc[2],*ft[2],*fd[2],*fr;
        int flag = 0,cmpf = 1;
        char buf[BUFCM],codet[8],coded[5],buft[106],sterr[30];
        float tval[10];

        scrpos(12,31l);
        printf("Teacher [          ]");
        strcpy(codet,rdchr(7,12,40));
        if (! isnum(7,codet)) {
                sprintf(sterr, ". . . CODE [%7s] error . . .",codet);
                status(sterr);
                return(1);
        }
        scrpos(12,12);
        clreol( );
        dubln( );
        printf("Teacher : %7s",codet);
        if (fopenndx("compu2", "r",fc) != NULL) {
                fopeninx("lect", "r",ft);
                fopeninx("dept", "r",fd);
                wrmid(15, ". . . Please wait . . .");
                refresh( );
                fr = fopen(FREPT, "w");
                freadinx(buft,codet,ft);
                flag = fstndx(fc,atoi(strcut(buft,8,11)),0);
                while ((cmpf > 0) && (flag)) {
                        if (fgets(buf,BUFCM,fc[0]) != NULL) {
                                cmpf = strcmp(codet,strcut(buf,5,11));
                        } else {
```

```
                        flag = 0;
                }
        }
        if (cmpf == 0) {
                strcpy(coded,strcut(buf,1,4));
                headrep(fr,type);
                facrep(fr,fd,coded);
                deprep(fr,fd.coded);
                vodyrep(fr,ft,buf,type,tval);
        }
        lnrep(fr);
        if (!col132)
                COND_CH(fr,OFF);
        NORMFEED(fr);
        fclose(fr);
        fcloseinx(fd);
        fcloseinx(ft);
        fclosendx(fc);
    }
}
mkrep_d(type)
int type;
{
    FILE *fc[2],*ft[2],*fd[2],*fr;
    int headf = 0,cmpf = 1;
    char buf[BUFCM],coded[5],sterr[30];
    float tval[10],dval[10];

    scrpos(12,32);
    printf("Department [      ]");
    strcpy(coded,rdchr(4,12,44));
    if (! isnum(4,coded)) {
        sprintf(sterr, ". . . CODE [%3s] error . . .",coded);
        status(sterr);
        return(1);
    }
    scrpos(12,12);
```

```
        clreol( );
        dubln( );
        printf("Department : %4s",coded);
        if (fopenndx("compu2", "r",fc) != NULL) {
                fopeninx("lect", "r",ft);
                fopeninx("dept", "r",fd);
                wrmid(15, ". . . Please wait . . .");
                refresh( );
                fr = fopen(FREPT, "w");
                fstndx(fc,atoi(coded),0);
                while ((cmpf >= 0) && (fgets(buf,BUFCM,fc[0]) != NULL)) {
                        cmpf = strcmp(coded,strcut(buf,1,4));
                        if (cmpf == 0) {
                                if (!headf) {
                                        headrep(fr,type);
                                        headf = 1;
                                        facrep(fr,fd,coded);
                                        deprep(fr,fd,coded);
                                        setary(dval);
                                }
                                bodyrep(fr,ft,buf,type,tval);
                                addary(dval,tval);
                        }
                }
                lnrep(fr);
                sumrep(fr,1,dval);
                lnrep(fr);
                if (!Col132)
                        COND_CH(fr,OFF);
                NORMFEED(fr);
                fclose(fr);
                fcloseinx(fd);
                fcloseinx(ft);
                fclosendx(fc);
        }
}
```

```
mkrep_f(tyep,num)
int type,num;
{
        FILE *fc[2],*fd[2],*fr;
        int headf = 0,inf = 1;
        char buf{BUFCM],codef[5],strtmp[50],coded[5],ocoded[5];
        float tval[10],dval[10],fval[10];

        scrpos(12,14);
        sprintf(strtmp, "FACULTY : %s",fname[num]);
        wrmid(12,strtmp);
        if (fopenndx("compu2", "r",fc) != NULL) {
                fopeninx("dept", "r",fd);
                wrmid(15, . . . Please wait . . .");
                refresh( );
                fr = fopen(FREPT, "w");
                strcpy(codef,fackey[num]);
                fstndx(fc,atoi(codef),1);
                strcpy(ocoded, "");
                while ((inf) && (fgets(buf,BUFCM,fc[0]) != NULL)) {
                        strcpy(coded,strcut(buf,1,4));
                        if (num == facul(atoi(coded)) ) {
                                if (!headf) {
                                        headrep(fr,type);
                                        headf = 1;
                                        facrep(fr,fd,codef);
                                        setary(fval);
                                        setary(dval);
                                }
                                if (strcmp(ocoded,coded)) {
                                        if (strcmp(ocoded, ""))
                                                depsum(fr,fd,1,ocoded,dval);
                                        strcpy(ocoded,coded);
                                        setary(dval);
                                }
                                bodyval(buf,type,tval);
                                addary(dval,tval);
```

```
                        addary(fval,tval);
                    } else {
                        inf = 0;
                    }
                }
                depsum(fr,fd,1,ocoded,dval);
                lnrep(fr);
                sumrep(fr,2,fval),;
                lnrep(fr);
                if (!col132)
                    COND_CH(fr,OFF);
                NORMFEED(fr);
                fclose(fr);
                fcloseinx(fd);
                fclosendx(fc);
        }
}
mkrep_u(type)
int type;
{
        FILE *fc,*fd[2],*fr;
        int headf = 0,nmfac,onmfac;
        char buf[BUFCM],coded[5],ocoded[5];
        float tval[10],fval[10],uval[10];
        scrpos(12,15);
        dubln( );
        printf("UNIVERSITY");
        if ((fc = fopen("compu2", "r")) != NULL) {
                fopeninx("dept", "r",fd);
                wrmidl(15, ". . . Please wait . . .");
                refresh( );
                fr = fopen(FREPT, "w");
                strcpy(ocoded, "");
                onmfac = -1;
                while(fgets(buf,BUFCM,fc) != NULL) {
                        strcpy(coded,strcut(buf,1,4));
```

```
                nmfac = facul(atoi(coded));
                if (!headf) {
                        headrep(fr,type);
                        headf = 1;
                        setary(uval);
                }
                if (strcmp(ocoded,coded)) {
                        if (nmfac != onmfac) {
                                if (onmfac != -1)
                                        depsum(fr,fd,2,fackey[onmfac],fval);
                                onmfac = nmfac;
                                setary(fval);
                        }
                        strcpy(ocoded,coded);
                }
                bodyval(buf,type,tval);
                addary(fval,tval);
                addary(uval,tval);
        }
        depsum(fr,fd,2,fackey[onmfac],fval);
        lnrep(fr);
        sumrep(fr,3,uval);
        lnrep(fr);
        if (!col132)
                COND_CH(fr,OFF);
        NORMFEED(fr);
        fclose(fr);
        fcloseinx(fd);
        fclose(fc);
    }
}
repch( )
{
    int choc;
    scrpos(10,33);
    printf("1. Teacher");
```

```
        scrpos(12,33);
        printf("2. Department");
        scrpos(14,33);
        printf("3. Faculty");
        scrpos(16,33);
        printf("4. University");
        scrpos(18,33);
        printf("5. EXIT");
        scrpos(22,50);
        printf("SELECT [5]");
        do {
                choc = rddg(22,58);
        } while ((choc < 0) || (choc > 5));
        return((choc == 0) ? 5 : choc);
}
```

```
/* <<<<<<<<<<<< frph.c >>>>>>>>>>>> */
#include <stdio.h>
#include "scr.h"

#define  CRTl_A  1
#define  CRTl_n  14
#define  ON      1
#define  OFF     0
#define  RETURN  '\015'
#define  COND_CH(FP,STATUS)        fprintf((FP), "%c",((STATUS) ? 15 : 18))
#define  ENLG_CH(FP,STATUS)fprintf((FP), "\033W%c", (STATUS))
#define  EMPH_ON(FP)          fprintf((FP), "\033E")
#define  EMPH_OFF(FP)              fprintf((FP), "\033F")
#define  NORMFEED(FP)              fprintf((FP), "\0332")

#define  HEADLN1     "รายงานแบบที่ 1"
#define  HEADLN2     "สรุปภาระงานสอนของอาจารย์"
#define  HEADLN3     "ประจำภาคการศึกษาที่ 2  ปีการศึกษา 2532"
#define  HEADLN4  \
        "จำนวนชั่วโมง                    จำนวนเรื่อง"
l \001          (mR9G9 'CwhM'          \016\
#define  HEADLN5  "ชื่อ       บรรยาย          ปฏิบัติการ
วอร์ด          สัมมนา          โครงงาน          วิทยานิพนธ์ภาระงาน"
#define  HEADLN6  \
        "ตรี          โท      ตรี      โท          ฝึกงาน          ตรี      โท"
#define  HEADSB1  "ข้อมูลวิชาที่สอนของแต่ละอาจารย์"
#define  HEADSB2  "รหัสอาจารย์"
#define  HEADSB3  "ชื่ออาจารย์ "
#define  HEADSB4  "รหัสวิชา                              ชื่อวิชา
                    ตอน     หน่วยกิต          ประเภทวิชา
    ร้อยละการสอน       จำนวนนักศึกษา"
#define  NONAME   "ไม่พบชื่ออาจารย์     รหัส"
#define  NODEPT   "ไม่พบชื่อภาควิชา     รหัส"
#define  NOFACU   "ไม่พบชื่อคณะ         รหัส"
#define  SUMDEP   "รวมทั้งภาควิชา"
#define  SUMFAC   "รวมทั้งคณะ"
#define  SUMUNI   "รวมทั้งมหาวิทยาลัย"
extern  int  Col132,term,year,rdyf;
```

```
char fackey[ ][15] = {"0000", "2000", "2500", "3000", "3500", "4000", "4500", "5000", "5500",
                      "6000", "6500", "7000", "9500", "9600"};
cahr fname[ ][30]  = {  "CODE OUT OF RANGE",
                        "ENGINEERING",
                        "EDUCATION",
                        "SCIENCE",
                        "MEDICINE",
                        "HUMANITIIES AND SOCIAL SCIENCE",
                        "MANAGEMENT SCIENCE",
                        "NATURAL RESOURCES",
                        "PHARMACY",
                        "NURSING",
                        "DENTISTRY",
                        "SCIENCE AND TECHNOLOGY",
                        "GRADUATE SCHOOL",
                        "PHUKET COMMUNITY COLLEGE"  };
hdsbrp(fp,codet)
FILE  *fp;
char  codet[ ];
{
        FILE  *ft[2],*fd[2];
        char  *thstr[4],*tmp,buft[106],coded[5];

        thstr[0] = (char *) malloc(200);
        thstr[1] = (char *) malloc(200);
        thstr[2] = (char *) malloc(200);
        thstr[3] = (char *) malloc(200);
        tmp = (char *) malloc(200);
        fopeninx("lect", "r",ft);
        fopeninx("dept", "r",fd);
        fprintf(fp, "\0333\001\n\n\n\n");
        if (col132)
                ENLG_CH(fp,ON);
        thaich(HEADSB1,thstr);
        thwstr(fp,((col132) ? 21 : 27),thstr);
        if (col132)
                ENLG_CH(fp,OFF);
```

```
        else
                COND_CH(fp,ON);
        thaich(HEADLN3,thstr);
        thstr[2][17] = term + '0';
        sprintf(tmp, "%.28s %4d",thstr[2],year);
        strcpy(thstr[2],tmp);
        thwstr(fp,49,thstr);
        thaich(HEADSB2,thstr);
        sprintf(tmp, "%.9s %7s",thstr[2],codet);
        strcpy(thstr[2],tmp);
        thwstr(fp,4,thstr);
        if (freadinx(buft,codet,ft) == OL)
                sprintf(buf, "%7s%4s%-26s%7s-%50s\n"
                                                ,codet, "000",NONAME,codet, " ");
        strcpy(coded,strcut(buft,8,11));
        strcpy(mp,HEADSB3);
        strcat(tmp,strcut)(buft,12,51));
        thaich(tmp,thstr);
        thwstr(fp,4,thstr);
        deprep(fp,fd,coded);
        facrep(fp,fd,coded);
        lnrep(fp);
        thaich(HEADSB4,thstr);
        thwstr(fp,5,thstr);
        lnrep(fp);
        NORMFEED(fp);
        fprintf(fp, "\n");
        free(thstr[0]);
        free(thstr[1]);
        free(thstr[2]);
        free(thstr[3]);
        free(tmp);
        fcloseinx(fd);
        fcloseinx(ft);
}

headrep(fp,type)
```

```
FILE *fp;
int type;
{
        char *thstr[4],*tmp;
        int i;

        rdyf = 1;
        thstr[0] = (char *) malloc(200);
        thstr[1] = (char *) malloc(200);
        thstr[2] = (char *) malloc(200);
        thstr[3] = (char *) malloc(200);
        tmp = (char *) malloc(200);
        if (col132)
                ENLG_CH(fp,ON);
        thaich(HEADLN1,thstr);
        thstr[2][12] = type + '0';
        thwstr(fp,((col132) ? 27 : 32),thstr);
        thaich(HEADLN2,thstr);
        thwstr(fp,((col132) ? 22 : 27),thstr);
        if (col132)
                ENLG_CH(fp,OFF);
        else
                COND_CH(fp,ON);
        thaich(HEADLN3,thstr);
        thstr[2][17] = term + '0';
        sprintf(tmp, "%.28s %4d",thstr[2],year);
        strcpy(thstr[2],tmp);
        thwstr(fp,49,thstr);
        lnrep(fp);
        thaich(HEADLN4,thstr);
        thwstr(fp,44,thstr);
        fprintf(fp, "%44s", " ");
        fputc('|',fp);
        for (i = 0;i < 53;i++)
                fputc('-',fp);
        fputc('|',fp);
        fputc('\n',fp);
```

```
        thaich(HEADLN5,thstr);
        thwstr(fp,19,thstr);
        thaich(HEADln6,thstr);
        sprintf(tmp, "\0333%c",1);
        strcat(thstr[0],tmp);
        thwstr(fp,44,thstr);
        lnrep(fp);
        free(thstr[0]);
        free(thstr[1]);
        free(thstr[2]);
        free(thstr[3]);
        free(tmp);
}

deprep(fp,fd,coded)
FILE  *fp,*fd[2];
char  coded[ ];
{
        char  bfd[90],*thstr[4];

        thstr[0]  =  (char  *)  malloc(200);
        thstr[1]  =  (char  *)  malloc(200);
        thstr[2]  =  (char  *)  malloc(200);
        thstr[3]  =  (char  *)  malloc(200);
        if  (freadinx(bufd,coded,fd)  ==  OL)
                sprintf(bufd, "%4s       %-20s%4s-%40s\n ",coded,NODEPT,coded,"");
        thaich(strcut(bufd,10,61),thstr);
        EMPH_ON(fp);
        thwstr(fp,4,thstr);
        EMPH_OFF(fp);
        free(thstr[0];
        free(thstr[1];
        free(thstr[2];
        free(thstr[3];
}

facrep(fp,fd,code)FILE  *fp,*fd[2];
char  code[ ];
```

```
{
        char  bufd[90],*thstr[4],*tmp;
        int  fcnm;
        thstr[0]  =  (char  *)  malloc(200);
        thstr[1]  =  (char  *)  malloc(200);
        thstr[2]  =  (char  *)  malloc(200);
        thstr[3]  =  (char  *)  malloc(200);
        tmp  =  (char  *)  malloc(200);
        fcnm  =  facnl(atoi(code));
        if  (fcnm  !=  0)
                freadinx(bufd,fackey[fcnm],fd);
        else
                sprintf(bufd, "%4s     %-22s%43s\n",code,NOFACU,code, " ");
        thaich(strcut(bufd,10,61),thstr);
        EMPH_ON(fp);
        thwstr(fp,4,thstr);
        EMPH_OFF(fp);
        free(thstr[0]);
        free(thstr[1]);
        free(thstr[2]);
        free(thstr[3]);
        free(tmp);
}
bodyrep(fp,ft,buf,type,rval)
FILE  *fp,*ft[ ];
char  buf[ ];
int  type;
float  rval[ ];
{
        char  *rpbuf[4],*wrbuf,lect[106],tcod[8],tmp[50];
        float  dat;
        long  ln;
        int  i,j  =  0;
        rpbuf[0]  =  (char  *)  molloc(200);
        rpbuf[1]  =  (char  *)  molloc(200);
        rpbuf[2]  =  (char  *)  molloc(200);
```

```
rpbuf[3] = (char *) molloc(200);
wrbuf    = (char *) molloc(400);
strcpy(tcod,strcut(buf,5,11));
if (freadinx(lect,tcod,ft) == OL) {
        sprintf(lect, "%7s%4s%-26s%7s-%50s\n"
                                        ,tcod, "0000",NONAME,tcod, " ");
}
thaich(strcut(lect,12,51),rpbuf);
sprintf(wrbuf, "%-40.40s",rpbuf[2]);
for (i = 0;i < 6;i++) {
        dat = (float atoi(strcut(buf,12+(i*5),16+(i*5))) / 1000.0;
        if (dat == 0.0)
                sprintf(tmp, "      - ");
        else
                sprintf(tmp, "    %5.2f",dat);
        strcat(wrbuf,tmp);
        rval[j++] = dat;
}
for (I = 0;I < 3;I++) [
        dat = (float) atoi (strcut(ubf,42+(i*2),43+(i*2)));
        if (dat == 0.0)
                sprintf(tmp, "      - ");
        else
                sprintf(tmp, "    %2.0f",dat);
        strcat(wrbuf,tmp);
        rval[j++] = dat;
}
if (type == 1)
        dat = (float atoi(strcut(buf,48,53)) / 1000.0;
else
        dat = (float atoi(strcut(buf,54,59)) / 1000.0;
if (dat == 0.0)
        sprintf(tmp, "      - ");
else
        sprintf(tmp, "    %6.2f",dat);
strcat(wrbuf,tmp);
```

```
        rval[j] = dat;
        strcpy(rpbuf[2],wrbuf);
        thwstr(fp,6,rpbuf);
        free(rpbuf[0]);
        free(rpbuf[1]);
        free(rpbuf[2]);
        free(rpbuf[3]);
        free(wrbuf);
}
bodyval(buf,type,rval)
char  buf[ ];
int  type;
float  rval[ ];
{
        int  i,,j  =  0;
        for  (i  =  0;i  <   6;i++)  {
                rval[j++]  =  (float)atoi(strcut(buf,12+(i*5),16+(i*5)))   /   1000.0;
        }
        for  (i  =  0;i  <  3;i++)  {
                rval[j++]  =  (float)  atoi(strcut(buf,42+(i*2),43+(i*2)));
        }
        if  (type  ==  1)
                rval[j]  =  (float)  atoi(strcut(buf,48,53))   /   1000.0;
        else
                rval[j]  =  (float)  atoi(strcut(buf,54,59))   /   1000.0;
}
depsum(fp,fd,sby,coded,val)
FILE  *fp,*fd[2];
int  sby;
char  coded[ ];
float  val[ ];
{
        int  i,j  =  0;
        char  bufd[90],*thstr[4],*tmp,*wrbuf;
        thstr[0]  =  (char  *)  malloc(200);
        thstr[1]  =  (char  *)  malloc(200);
```

```
thstr[2] = (char *) malloc(200);
thstr[3] = (char *) malloc(200);
wrbuf = (char * malloc(400);
tmp = (char *) malloc(200);
if (freadinx(bufd,coded,fd) == OL)
        if (sby == 1)
                sprintf(bufd, "%4s%-22s%4s-%43s\n"
                                        ,coded,NODEPT,coded, " ");
        else
                sprintf(bufd, "%4s%-22s%4s-%43s\n"
                                        ,coded,NOFACU,coded, " ");
thaich(strcut(bufd,10,61),thstr);
sprintf(wrbuf, "%-35.35s",thstr[2]);
for (i = 0;i < 6;i++) {
        if (val[j] == 0.0)
                sprintf(tmp, "      - ");
        else
                sprintf(tmp, "    %6.2f",val[j]);
        j++;
        strcat(wrbuf,tmp);
}
for (i = 0;i < 3;i++) {
        if (val[j] == 0.0)
                sprintf(tmp, "      -");
        else
                sprintf(tmp, "    %3.0f",val[j]):
        j++;
        strcat(wrbuf,tmp);
}
if (val[j] == 0.0)
        sprintf(tmp, "      - ");
else
        sprintf(tmp, "    %7.2f,val[j]);
strcat(wrbuf,tmp);
strcpy(thstr[2],wrbuf);
thwstr(fp,8,thstr);
```

```
        free(tmp);
        free(thstr[0]);
        free(thstr[1]);
        free(thstr[2]);
        free(thstr[3]);
}

sumrep(fp,sby,val)
FILE *fp;
int sby;
float val[ ];
{
        char *rpbuf[4],*wrbuf,tmp[50];
        int i,j = 0;

        rpbuf[0] = (char *) malloc(200);
        rpbuf[1] = (char *) malloc(200);
        rpbuf[2] = (char *) malloc(200);
        rpbuf[3] = (char *) malloc(200);
        wrbuf    = (char *) malloc(400);
        strcpy(tmp, " ");
        switch(sby) {
                case 1 : strcpy(tmp,SUMDEP);
                         break;
                case 2 : strcpy(tmp,SUMFAC);
                         break;
                case 3 : strcpy(tmp,SUMUNI);
        }
        thaich(tmp,rpbuf);
        sprintf(wrbuf, "%-30s",rpbuf[2]);
        for (i = 0;i < 6;i++) {
                if (val[j] == 0.0)
                        sprintf(tmp, "       - ");
                else
                        sprintf(tmp, "    %6.2f",val[j]);
                j++;
                strcat(wrbuf,tmp);
        }
```

```
        for (i =0;i <3;i++) {
                if (val[j] == 0.0)
                        sprintf(tmp, "          -");
                else
                        sprintf(tmp, "      %3.0f",val[j]);
                j++;
                strcat(wrbuf,tmp);
        }
        if (val[j] == 0.0)
                sprintf(tmp, "          -   ");
        else
                sprintf(tmp, "      %7.2f",val[j]);
        strcat(wrbuf,tmp);
        strcpy(rpbuf[2],wrbuf);
        EMPH_ON(fp);
        thwstr(fp,16,rpbuf);
        EMPH_OFF(fp);
        free(rpbuf[0]);
        free(rpbuf[1]);
        free(rpbuf[2]);
        free(rpbuf[3]);
        free(wrbuf);
}

lnrep(fp)
FILE *fp;
{
        int i;
        fprintf(fp, "          ");
        for (i = 0;i < 126;i++)
                fputc('-',fp);
        fputc('\n',fp);
}

thwstr(fp,blk,str)
FILE *fp;
int blk;
char *str[ ];
```

```
{
        int  i;
        for  (i  =  0;i  <  4;i++)
                fprintf(fp, "%*s%s\n",blk, " ",str[i]);
}
facnum( )
{
        int  i,ch;
        hghchr( );
        wrmid92, "FACULYCODE");
        norchr( );
        for  (i  =  0;i  <14;i++)  {
                scrpos(5=i,25);
                printf("%c.  %s",i+ 'A',fname[i]);
        }
        scrpos(19,25);
        printf("X.  EXIT");
        scrpos(24,55);
        printf("Enter faculty code  [X]");
        do  {
                scrpos(24,75);
                ch  =  rdupp( );
        }  while  ((ch  !+  'X')  &&  ((ch  <  'A')  ||  (ch  >  'N'))  &&  (ch  !=  RETURN));
        return(((ch  ==  'X')  ||  (ch  ==  RETURN))  ?  )  :  ch  -  'A'  +  1);
}
setary(val)
float val[ ];
{
        int  i;
        for  (i  =  0;i  <10;i++)
                val[i]  =  0.0;
}
addary(valt,vals)
float  valt[ ],vals[ ];
{
        int  i;
```

```
        for (i = 0;i < 10;i++)
                valt[i] += vals[i];
}
/* <<<<<<<<<<<< fthw.c >>>>>>>>>>>> */
#define CTRL_A 01
#define CTRL_N 14
#include <stdio.h>
thaich(str,stt)
char *str,*stt[ ];
{
        int size,i,jm,jx,fth,f12;
        char ch;
        jx = jm = 0;
        fth = f12 = 0;
        size = strlen(str);
        for (i = 0;i <size;i++) {
                ch = *(str + I);
                if (fth) {
                        if (ch == CTRL_N) {
                                fth = 0;
                        } else {
                                ch += 128;
                                switch (inline(ch - 128)) {
                                case 'U' : if (f12 {
                                                if ((ch - 128) == 'm') {
                                                        stt[0][jx-1]=stt[1][jx-1];
                                                        stt[1][jx-1] = ch;
                                                } else {
                                                        stt[0][jx-1] = ch;
                                                }
                                        } else }
                                                stt[1]jjx-1] = ch;
                                                f12 = 1;
                                        }
                                        break;
                                case 'M' : if ((ch - 128) != '\t') {
```

```
                                stt[0][jx]  = stt[1][jx]
                                            = stt[3][jx] = ' ';
                                stt[2][jx]  = ch;
                            } else {
                                stt[0][jx]  = stt[1][jx]
                                            = stt[2][jm]
                                            = stt[3][jx] = '\t';
                            }
                            jm++;
                            jx++;
                            f12 = 0;
                            break;
                    case 'L' : stt[3][jx-1] = ch;
                    }
                }
        } else {
                if (ch == CTRL_A) {
                        fth = 1;
                } else {
                        if (ch != '\t') {
                                stt[0][jx]=stt[1][jx]=stt[3][jx] = ' ';
                        } else {
                                stt[0][jx]=stt[1][jx]=stt[3][jx] = '\t';
                        }
                        jx++;
                        stt[2][jm++] = ch;
                }
        }
}
        /* put thai linefeed code & mull charecter */
stt[0][jx] = stt[1][jx] = stt[2][jm++] = stt[3][jx] = '\033';
jx++;
stt[0][jx] = stt[1][jx] = stt[2][jm++] = stt[3][jx] = '3';
jx++;
stt[0][jx] = 15;stt[1][jx] = 24;stt[2][jm++] = 22;stt[3][jx] = 10;
jx++;
```

```
        stt[0][jx] = stt[1][jx] = stt[2][jm] = stt[3][jx] = '\000';
}

inline(ch)
char ch;
{
        int i,ret;
        char nomidd[19];

        strcpy(nomidd, "YZXi[ ]ghjklm\QWTUV");
        for (i = 0;(i <18) && (ch != nomidd[i]);i++);
        if (i < 3) {
                ret = 'L';
        } else {
                if (i < 18) {
                        ret = 'U';
                } else {
                        re = 'M';
                }
        }
        return(ret);
}
```