```
/*
This reads the Stommel-Luyten file format - data has been scaled and
converted to binary integers

build with
 cc sl.c ~jgofs/src/outer.o -lm -o sl
*/

#include <stdio.h>
#include <string.h>
#include <fcntl.h>
#include <math.h>
double atof();

/* #define DEBUG */

/* number of variables */
#define NVAR 10

/* number of integers per record & number of bytes*/
#define NREAD 9
#define NREADBYTE 18

/* number of group variables (level=0) */
#define NGROUP 5

/* rest of variables are level=1 */

/* flags for controlling reading sequence */
#define OK0 -999
#define ERR1 -9999

static char *names[NVAR]={"station","lat","lon","year","month",
         "press","temp","sal","o2","sigth"};

/* unit number for file */
static int fl;

/* oldstn:saves old station number, vrd is input record */
static short oldstn= -1,vrd[NREAD];

/* comments to be returned */
static char comments[]="Data prepared by Stommel and Luyten";

/* error reporting */
err(s,t)
char *s,*t;
{
printf("&x error - %s %s\n",s,t);
exit(1);
}

/*
     open subroutine
*/
int ioopen_(s,nparams,ntotal)
char *s[];
int *nparams,*ntotal;
{
/* open file named in first parameter */
if((fl=open(s[0],O_RDONLY)) < 0)err("header file name ",s[0]);
/* clear parameter so that the outer routine doesn't deal with it */
s[0][0] = 0;
/* return number of variables */
*ntotal=NVAR;
/* set up for next group read */
vrd[0]=oldstn;
/* return max hierarchical level =1 */
return 1;
}

/*
     return name of variable number *vn
*/
ioname_(vn,s)
int *vn;
char *s;
{
strcpy(s,names[*vn]);
}

/*
      return level of variable number *vn
*/
int iovarlevel_(vn)
int *vn;
{
if(*vn<NGROUP) return 0;
return 1;
}

/*
      output attributes of variable number *vn
*/
int ioattrout_(vn,str)
int *vn;
char *str;
{
/* none to return */
return 0;
}

/*
    return value of variable *vn as a real number
*/
iovalreal_(vn,f)
int *vn;
float *f;
{
int i;
i= *vn;
if(i<0){*f= -9999.0;return;};
switch(i){
  case 0:*f=vrd[0];break;
  case 1:
  case 2:*f=vrd[i]*0.01;break;
  case 3:*f=floor(0.01*vrd[3]);break;
  case 4:*f=vrd[3]-100*floor(0.01*vrd[3]);break;
  case 5:*f=vrd[4];break;
  case 6:*f=0.001*vrd[5];break;
  case 7:*f=0.001*vrd[6]+15.0;break;
  case 8:*f=0.01*vrd[7];break;
  case 9:*f=0.001*vrd[8];break;
  };
}

/*
      return value of variable *vn as a string
```

```
*/
iovalstr_(vn,tmp)
int *vn;
char *tmp;
{
char *s;
int i;
i= *vn;
if(i<0){strcpy(tmp,"nd");return;};
switch(i){
  case 0:sprintf(tmp,"%4d",vrd[0]);break;
  case 1:
  case 2:sprintf(tmp,"%6.2f",vrd[i]*0.01);break;
  case 3:sprintf(tmp,"%3d",vrd[3]/100);break;
  case 4:sprintf(tmp,"%2d",vrd[3]-100*(vrd[3]/100));break;
  case 5:sprintf(tmp,"%4d",vrd[4]);break;
  case 6:sprintf(tmp,"%6.3f",0.001*vrd[5]);break;
  case 7:sprintf(tmp,"%6.3f",0.001*vrd[6]+15.0);break;
  case 8:sprintf(tmp,"%5.2f",0.01*vrd[7]);break;
  case 9:sprintf(tmp,"%6.3f",0.001*vrd[8]);break;
  };
s=tmp+strspn(tmp," ");
strcpy(tmp,s);
}

/*
      get next record at requested level
*/
int ioreadrec_(level)
int *level;
{
if(fl<0)return 0;
switch(*level){
  case 0:
/* level 0: if we just read a record with a new station number in the
   level 1 code, return ok at this level without reading more */
    if(oldstn == ERR1){
      oldstn= OK0;
      return 1;
      };
/*  if previous read at level 0 was ok, record the station
    number */
    if(oldstn == OK0){
      oldstn=vrd[0];
      };
/* read to new station number */
    while(vrd[0] == oldstn)
      if(read(fl,vrd,NREADBYTE) <= 0){close(fl);fl= -1;return 0;};
/* set that we read it ok at this level */
    oldstn= OK0;
    return 1;
  case 1:
/* level 1: if we just completed a group properly, save the station
   number */
    if(oldstn == OK0){
      oldstn=vrd[0];
      return 1;
      };
/* read at level 1 */
    if(read(fl,vrd,NREADBYTE) <= 0){close(fl);fl= -1;return 0;};
    if(vrd[0]==oldstn)return 1;
/* station number has changed: save that we changed at this level
   and report that we finished a group at level 1 */
    oldstn= ERR1;
    return 0;
  };
}

/*
    return comments
*/
int iocommout_(str)
char *str;
{
if (*comments) return 0;
strcpy(str,comments);
*comments=0;
return 1;
}

/*
    close file
*/
ioclose_()
{
if(fl>=0)close(fl);
}
```