# ภาคผนวก ข

ตัวโปรแกรมภาษาซี (Turbo C) ซึ่งใช้เป็นโปรแกรมวิเคราะห์สเปกตรัมความถี่ของเสียงดนตรีไทย โดยเก็บไว้บนเครื่องไมโครคอมพิวเตอร์ ชื่อไฟล์ W1024.C

```
/*    w1024.c                                          */
/*    works with w1024.asm                             */
/*    Thank for B.Phongdara (Adviser)                  */
/*    modify fft( ) Discrete Fast Fourier Transform    */
/*    Thank for Don Cross,May 2003                     */
#include <dos.h>
#include <stdio.h>
#include <graphics.h>
#include <conio.h>
#include <string.h>
#include <math.h>
#include <fourier.h>
#include <ddcmath.h>

#define CHECKPOINTER(p)  CheckPointer(p,#p)
static void CheckPointer ( void *p, char *name )
{
   if ( p == NULL )
   {
      fprintf ( stderr, "Error in fft():  %s == NULL\n", name );
      exit(1);
   }
}

#define TRUE  1
#define FALSE 0
#define BITS_PER_WORD   (sizeof(unsigned) * 8)
```

```
int IsPowerOfTwo ( unsigned x )
{
    if ( x < 2 )
        return FALSE;
    if ( x & (x-1) )       /* Thanks to 'byang' for this cute trick!*/
        return FALSE;
    return TRUE;
}
unsigned NumberOfBitsNeeded ( unsigned PowerOfTwo )
{
    unsigned i;
    if ( PowerOfTwo < 2 )
    {
        fprintf (
            stderr,
            ">>> Error in fftmisc.c: argument %d to NumberOfBitsNeeded is too small.\n",
            PowerOfTwo );
        exit(1);
    }

    for ( i=0; ; i++ )
    {
        if ( PowerOfTwo & (1 << i) )
            return i;
    }
}

unsigned ReverseBits ( unsigned index, unsigned NumBits )
{
    unsigned i, rev;

    for ( i=rev=0; i < NumBits; i++ )
    {
        rev = (rev << 1) | (index & 1);
```

```
        index >>= 1;
    }

    return rev;
}

double Index_to_frequency ( unsigned NumSamples, unsigned Index )
{
    if ( Index >= NumSamples )
        return 0.0;
    else if ( Index <= NumSamples/2 )
        return (double)Index / (double)NumSamples;

    return -(double)(NumSamples-Index) / (double)NumSamples;
}

void port_init(int,int);
void sport(int,int);
char rport(int);
int check_stat(int);
plotline( int , int , int , int, float );
plotvector( int , int , int , int,int ,int , float );
ploty(int , int ,int ,int ,int);
void filehandle(int nb,char d);
void display1(int nb,int *p, int no);
void readdata(int p[]);
void plotdata(int nb,float *p,int *q);
void fft(
    unsigned  NumSamples,
    int       InverseTransform,
    float    *RealIn,
    float    *ImagIn,
    float    *RealOut,
    float    *ImagOut);
```

```
        int x,dx,dx2,y,maxx,maxy,x1,y1,average,a[1024],b[1024],c[1024];
        float max,min,dy,sum;
        int graphdriver=DETECT,graphmode;
        float n,time ,xt,yt,freq=1E6; /* 13 cycle time * 1.085e-6*/
/*      float time=1.41e-5 */
        char s[10],ans,move,disp,dis[15],check;
        int port,i,k,j=0,tl,th;
        char *ch,*idx,pcx[13],r,ch1,ch2;
        int xnew,xold,ynew,yold,yold1,yyold,yynew;
        unsigned size ;
        void *buff1;
        int cmode;
        float magnitude[512];
main()
{
        unsigned int m,nh,nl,t,d,p[1024],nb=1024; /* nb is no. of sampling points */
        float pa[1024],pi[1024],amp[1024],oa[1024],oi[1024],angle[1024];
        int  InverseTransform=0;
        char cs;
        clrscr();
        port = 0x00;
        port_init(port,0xe3);   /* 1000 0011 */
        detectgraph (&graphdriver,&graphmode);
        initgraph (&graphdriver,&graphmode,"");
        cmode = getgraphmode();
        maxx=getmaxx();
        maxy=getmaxy();
        max = 255.0;
        min  = 0.0;
        dx = maxx/512;
        dy = (maxy-10)/max;
        restorecrtmode();
  /* clear all a,b and c */
     for(i=0;i<nb;i++) {
```

```
            a[i] = 0 ;
            b[i] = 0 ;
            c[i] = 0 ;
        }

    do {
            clrscr();
            printf("  type  c    to establish connection \n");
            printf("\n        n    to collect new data    \n");
       printf("\n        s    to store data          \n");
            printf("\n        r    to read data from file \n");
            printf("\n        d    to display             \n");
            printf("\n        f    to FFT                 \n");
            printf("\n        q    to quit                \n");

            ch1 = getch();
            switch (ch1) {
               case 'c' : clrscr();
                          do {
                                printf(" Push RESET switch on the  SOUND SPECTRUM INTERFACE \n");
                                printf(" then hit enter to start linking with SOUND SPECTRUM INTERFACE
\n");
                                getch();

                                r=rport(port); /* these two lines do not */
                                sport(port,r); /* set conditions ,just to establish connections */

                                sport(port,255); /* condition starts here */
                                r = rport(port);
                                sport(port,r);
                                printf(" r = %c ,%d\n",r,r);
                          } while ( r != 10 );
                          clrscr();
                          printf("Put in time delays (18.4e-6 to 570e-6 sec) then Enter \n");
```

```
                scanf("%f",&time);
                tl=(int)(((time/1.085e-6)-15)/2+.5);
                time=(tl*2+15)*1.085e-6; /* this is real time for later use */
               /*  printf(" tl = %d, time = %10.20f \n",tl,time);
                getch();*/
                sport(port,tl);
                readdata(p);
                break;
case 'n' : clrscr();
                sport(port,tl);
                readdata(p);
                display1(nb,p,1);
                clrscr();
                printf("\n   Save this new data (y/n) ? \n");
                ch2 = getch();
                if (ch2 == 'y') {
                        printf("\n  store on a or b ? \n");
                        r = getch();
                        if ( r =='a') {
                          for(i=0;i<nb;i++)
                                a[i]=p[i];
                        }
                        if ( r=='b') {
                          for(i=0;i<nb;i++) {
                                b[i]=p[i];
                                c[i]=a[i]+b[i];
                          }
                        }
                }
                break;
case 'r' : clrscr();
                filehandle(nb,'r');
                printf("time = %e \n",time);
                getch();
```

```
                display1(nb,c,3);
                break;
case 's' : clrscr();
                filehandle(nb,'w');
                break;
case 'd' :
                clrscr();
                        printf("  type  r    for raw data  \n");
                        printf("\n        a    for wave A    \n");
                        printf("\n        b    for wave B    \n");
                        printf("\n        c    for wave C    \n");
                        printf("\n        t    all together  \n");
                        printf("\n        q    to quit       \n");
                  ch2 = getch();
                  switch (ch2) {
                  case 'r' : clrscr();
                             display1(nb,p,1);
                             break;
                  case 'a' : clrscr();
                             display1(nb,a,1);
                             break;
                  case 'b' : clrscr();
                             display1(nb,b,1);
                             break;
                  case 'c' : clrscr();
                             display1(nb,c,1);
                             break;
                  case 't' : clrscr();
                             display1(nb,c,3);
                             break;
                    } /* switch */
                break;
case 'f' : clrscr();
                printf("FFT a,b or c ? \n");
```

```
 ch2 = getch();
 if (ch2=='a') {
         for (i=0 ;i<nb ;i++) {
          pa[i] = a[i];
          pi[i] = 0.;
         /* printf("pa=%f a=%d pi=%f  i=%d \n",pa[i],a[i],p[i],i);
          getch();*/
        }
 /*  fft(nb,pa,pi,amp); */
     fft(nb,InverseTransform,pa,pi,oa,oi);
for (i=0;i<nb;i++)
 {
   amp[i] =  sqrt( oa[i]*oa[i] +oi[i]*oi[i] );
        /* angle[i] = atan2 ( oi[i],oa[i]);
         amp[i] = amp[i]*cos(angle[i]); negative peak*/

 }
 plotdata(nb,amp,a);
 }
 if (ch2=='b'){
         for (i=0 ;i<nb ;i++) {
          pa[i] = b[i];
          pi[i] = 0.;
        }
 /*  fft(nb,pa,pi,amp); */
 fft(nb,InverseTransform,pa,pi,oa,oi);
for (i=0;i<nb;i++)
 {
   amp[i] =  sqrt( oa[i]*oa[i] +oi[i]*oi[i] );
        /* angle[i] = atan2 ( oi[i],oa[i]);
         amp[i] = amp[i]*cos(angle[i]); negative peak */

 }
 plotdata(nb,amp,b);
```

```
                        }
                        if (ch2=='c') {
                                for (i=0 ;i<nb ;i++) {
                                 pa[i] = c[i];
                                 pi[i] = 0.;
                                }
                       /* fft(nb,pa,pi,amp);*/
                         fft(nb,InverseTransform,pa,pi,oa,oi);
                       for (i=0;i<nb;i++)
                        {
                          amp[i] =  sqrt( oa[i]*oa[i] +oi[i]*oi[i] );
                               /* angle[i] = atan2 ( oi[i],oa[i]);
                                amp[i] = amp[i]*cos(angle[i]); negative peak */

                        }
                        plotdata(nb,amp,c);
                        }
                        break;
              }   /* switch */
           } while (ch1 != 'q');
    closegraph();
 }

void fft(
   unsigned  NumSamples,
   int       InverseTransform,
   float    *RealIn,
   float    *ImagIn,
   float    *RealOut,
   float    *ImagOut)
/*    ,
   float    *magnitude)*/

   {
```

```
unsigned NumBits;    /* Number of bits needed to store indices */
unsigned i, j, k, n;
unsigned BlockSize, BlockEnd;

double angle_numerator = 2.0 * DDC_PI;
double tr, ti;     /* temp real, temp imaginary */
float magnitude[512];

if ( !IsPowerOfTwo(NumSamples) )
{
   fprintf (
      stderr,
      "Error in fft():  NumSamples=%u is not power of two\n",
      NumSamples );

   exit(1);
}

if ( InverseTransform )
   angle_numerator = -angle_numerator;

CHECKPOINTER ( RealIn );
CHECKPOINTER ( RealOut );
CHECKPOINTER ( ImagOut );

NumBits = NumberOfBitsNeeded ( NumSamples );

/*
**   Do simultaneous data copy and bit-reversal ordering into outputs...
*/

for ( i=0; i < NumSamples; i++ )
{
   j = ReverseBits ( i, NumBits );
```

```
    RealOut[j] = RealIn[i];
    ImagOut[j] = (ImagIn == NULL) ? 0.0 : ImagIn[i];
}

/*
**   Do the FFT itself...
*/

BlockEnd = 1;
for ( BlockSize = 2; BlockSize <= NumSamples; BlockSize <<= 1 )
{
    double delta_angle = angle_numerator / (double)BlockSize;
    double sm2 = sin ( -2 * delta_angle );
    double sm1 = sin ( -delta_angle );
    double cm2 = cos ( -2 * delta_angle );
    double cm1 = cos ( -delta_angle );
    double w = 2 * cm1;
    double ar[3], ai[3];
    double temp;

    for ( i=0; i < NumSamples; i += BlockSize )
    {
        ar[2] = cm2;
        ar[1] = cm1;

        ai[2] = sm2;
        ai[1] = sm1;

        for ( j=i, n=0; n < BlockEnd; j++, n++ )
        {
            ar[0] = w*ar[1] - ar[2];
            ar[2] = ar[1];
            ar[1] = ar[0];
```

```
            ai[0] = w*ai[1] - ai[2];
            ai[2] = ai[1];
            ai[1] = ai[0];

            k = j + BlockEnd;
            tr = ar[0]*RealOut[k] - ai[0]*ImagOut[k];
            ti = ar[0]*ImagOut[k] + ai[0]*RealOut[k];

            RealOut[k] = RealOut[j] - tr;
            ImagOut[k] = ImagOut[j] - ti;

            RealOut[j] += tr;
            ImagOut[j] += ti;
        }
    }

    BlockEnd = BlockSize;
}

/*
**   Need to normalize if inverse transform...
*/

if ( InverseTransform )
{
    double denom = (double)NumSamples;

    for ( i=0; i < NumSamples; i++ )
    {
        RealOut[i] /= denom;
        ImagOut[i] /= denom;
    }
}
/*   for              (i=0 ;i<NumSamples ;i++) {
```

```
                    float       magnitude[i] = sqrt( RealOut[i]*RealOut[i] + ImagOut[i]*ImagOut[i] );

                                                }  */
}
/*--- end of file fourierf.c ---*/
 void readdata (int p[])
 {
     int i,k;
     char r;
              printf("\n\n Please wait for seconds !!\n");
              i=0;
                do
                      {
                        r = rport(port);
                        sport(port,r);
                        k=r;
                        if(r<0)
                          k=256+r;
                        p[i] = k;
                        i = i+1;
                      }
                while ( i <1024 );
 }

void display1(int nb,int p[],int no)
{
  float max=255.0;
  float dy,sumb,maxc,sum;
  /* printf("no=%d ",no);getch();*/
  do {
     if (no == 3){
              sumb=0.;maxc=0.;
              for(i=0;i<nb;i++) {
                      sumb = sumb +b[i];
```

```
                maxc = ((c[i]>maxc) ? c[i] : maxc);
        }
        average = sumb/1024;
        dy = ((maxy-10)/max)/3;
        setgraphmode(cmode);
        cleardevice();
        for (i=0,x=0;i<nb;i++,x += dx)
                {
                y= (5.+dy*(max-a[i]));
                putpixel(x,y,63);
                }
        for (i=0,x=0;i<nb;i++,x += dx)
                {
                y= (maxy/3+dy*(max-b[i]));
                putpixel(x,y,63);
                }
        for (i=0,x=0;i<nb;i++,x += dx)
                {
                y= (maxy*2/3+dy*(maxc-c[i]));
                putpixel(x,y,63);
                }
        } /* end if */
else {
        max = 0.;
        sum = 0.;
        for(i=0;i<512;i++){
                sum = sum + p[i];
                max = ((p[i]>max) ? p[i] : max );
        }
        dy = (maxy-10)/255;
        average = sum/512;
        /* printf("average=%f ,dx=%d \n",average,dx);
        getch();
        */
```

```
        if (max < 255.0)
                max = 255.0;
        setgraphmode(cmode);
        cleardevice();
        for (i=0,x=0;i<512;i++,x += dx)
                {
                y= (30.+dy*(max-p[i]));
                putpixel(x,y,62);
                /* printf("x=%d ,y=%f,nb=%d \n",x,y,nb);
                getch();*/

                }
    } /* end else */
                setviewport(0,0,maxx,20,0);
                size = imagesize(0,0,maxx,20);
                buff1 = malloc(size);
                getimage(0,0,maxx,20,buff1);
                clearviewport();
                outtext("x-detail ,y-detail y, r-repeat, q-quit");
                disp = getch();
                check = disp;
                if (disp =='x') {
                  putimage(0,0,buff1,COPY_PUT);
                  setlinestyle(0,0,1);  /*solidln,0,norwidth */
                  setwritemode(1);      /* xorput */
                  xold = 0;
                  setviewport(0,0,maxx,maxy,0);

                  moveto(maxx*0.4,0);
                  outtext("r-right,l-left,q-quit, time( m-sec) = ");
                  x1 = getx();
                  y1 = gety();
/*                outtext(dis);*/
                  line(0,0,0,maxy);
```

```
                move = getch();
                do {
                  switch (move) {
                        case 'l' :xnew = xold-1;
                             plotline(xold,xnew,x1,y1,time);
                             xold = xnew;
                             break;
                         case 'r' :xnew = xold+1;
                             plotline(xold,xnew,x1,y1,time);
                             xold = xnew;
                             break;
                         case 'R'  :xnew = xold + 5;
                             plotline(xold,xnew,x1,y1,time);
                             xold = xnew;
                             break;
                         case 'L'  :xnew = xold - 5;
                             plotline(xold,xnew,x1,y1,time);
                             xold = xnew;
                             break;

                   }  /* switch */
                    move = getch();
                  } while (move != 'q');
               } /* if */

               if (disp =='y') {
                  putimage(0,0,buff1,COPY_PUT);
                  setlinestyle(0,0,1);  /*solidln,0,norwidth */
                  setwritemode(1);      /* xorput */
                  yold = (5.+dy*(max-average));
                  yold1 = yold;
/*                printf("yold1 = %d , average = %d \n",yold1,average);
                  getch();*/
                  setviewport(0,0,maxx,maxy,0);
```

```
                moveto(maxx*0.4,0);
                outtext("u-up ,d-down ,q-quit , y = ");
                x1 = getx();
                y1 = gety();
                line(0,yold1,getmaxx(),yold1);
                move = getch();
                do {
                  switch (move) {
                        case 'u' :ynew = yold-1;
                             ploty(yold,ynew,x1,y1,yold1);
                             yold = ynew;
                             break;
                         case 'd' :ynew = yold+1;
                             ploty(yold,ynew,x1,y1,yold1);
                             yold = ynew;
                             break;
                         case 'U'  :ynew = yold - 10;
                             ploty(yold,ynew,x1,y1,yold1);
                             yold = ynew;
                             break;
                         case 'D'  :ynew = yold + 10;
                             ploty(yold,ynew,x1,y1,yold1);
                             yold = ynew;
                             break;

                 }  /* switch */
                  move = getch();
                } while (move != 'q');
}   /*  end if */

free(buff1);
restorecrtmode();
if ( disp == 'r' )
{
```

```
                sport(port,0); /* signal 8031 for repeat */
                readdata(p);
            }

 }while (check == 'r') ;

}

 plotline(int old, int new, int xx1, int yy1, float time)
{
    char dis[25];
    int sig = 5;
    float num;
    if ( new < 0 || new > getmaxx())
            return(1);
    moveto(xx1,yy1);
    setcolor(0);
    num = old*time*1000;
    gcvt(num,sig,dis);
    outtext(dis);
    setcolor(63);
    line(old,0,old,getmaxy());
    line(new,0,new,getmaxy());
    moveto(xx1,yy1);
    num = new*time*1000;
    gcvt(num,sig,dis);
    outtext(dis);
}

 plotvector(int old, int new, int xx1, int yy1,int yo,int yn, float time)
 {
    char dis[25];
    int sig = 3;
    float num;
```

```
    if ( new < 0 || new > getmaxx())
            return(1);
    moveto(xx1,yy1);
    setcolor(0); /* color of number of frequency */
    num =(old/dx2)*(1/(time*1024));
    /* printf("old=%d    num=%f   1/time=%f time=%f dx2=%d\r",old,num,1/(time*1024),time,dx2);
    delay(10);*/
    /* num = old*time*1000;*/
    gcvt(num,sig,dis);
    outtext(dis);
    setcolor(63);

    /*  line(old,0,old,getmaxy());
    line(new,0,new,getmaxy());
    */
    line(old,yo,old,yo+20);
    line(old,yo+20,old-2,yo+15);
    line(old,yo+20,old+2,yo+15);/*draw arrow*/

    line(new,yn,new,yn+20);
    line(new,yn+20,new-2,yn+15);
    line(new,yn+20,new+2,yn+15);
    moveto(xx1,yy1);
    num = (new/dx2)*(1/(time*1024));
    gcvt(num,sig,dis);
    outtext(dis);
}

ploty(int old, int new, int xx1, int yy1, int old1)
{
    char dis[25];
    int sig = 5;
    float num;
    if ( new < 0 || new > getmaxx())
```

```
            return(1);
    moveto(xx1,yy1);
    setcolor(0);
    num = old1-old;
/*    printf(" num = %f \n",num);
    getch();*/
    gcvt(num,sig,dis);
    outtext(dis);
    setcolor(63);
    line(0,old,getmaxx(),old);
    line(0,new,getmaxx(),new);
    moveto(xx1,yy1);
    num = old1-new;
/*    printf(" num = %f \n",num);
    getch();*/
    gcvt(num,sig,dis);
    outtext(dis);
}

void port_init(int port,int code)
{
            union REGS r;

            r.x.dx=port;
            r.h.ah=0;
            r.h.al=code;
            int86(0x14,&r,&r);
}

void plotdata(int nb, float *p,int q[])

{
    int x,dx1,y,y1,nb1,i,ck,maxx,maxy;
    float max,min,dy,dy1,max1;
```

```
  int graphdriver=DETECT,graphmode;
  char s[40];

  detectgraph(&graphdriver,&graphmode);
  initgraph (&graphdriver,&graphmode,"");
  setgraphmode(cmode);
  printf("Put in zoom level (1 to 9) then Enter (2 recommended) \n");
  scanf("%d",&dx2);
  cleardevice();

  nb = nb/8;/* nb=nb/4 or nb=nb/2 or /10  not change for freq */
  maxx=getmaxx();
  maxy=getmaxy();
  yyold = maxy*1.9/3; /*diff from reso25.c, see also y at p[0] below*/
  max = 0.;
  max1 = 0.;
  min = 0.;
  for(i=0;i<512;i++)
     max1=((q[i]>max1) ? q[i]:max1);
     /* printf("max1=%f q=%d i=%d \n",max1,q[i],i);getch();*/
  for (i=0 ;i<nb ;i++) {
     max = ((p[i]>max) ?p[i]:max);  /* maximum & min y */
     min = ((p[i]<min) ?p[i]:min);
     /*printf("max=%f  min=%f p=%f i=%d \n",max,min,p[i],i);getch();*/
  }

/*  max=max/3.;  to reduce the first line,diff from reso25.c */
/*  dx = maxx/nb;*/
/*  dx = 2;  to reduce interval of x axis */
/*  dx1= maxx/nb1;*/
    dx1= maxx/512 ;
/*  printf("dx1 = %d, nb= %d ",dx1,nb1);*/
/*  getch();*/
  dy = ((maxy-10)/(max-min));
```

```
dy1 =((maxy-10)/max1)/2;  /* see display1 */

/* data plot starts here */
for(i=0,x=0;i<512;i++,x+=dx1) {
  y1 = (20.+dy1*(max1-q[i]));
  putpixel(x,y1,63);
}

/* here is fft plot */
for (i=0,x=0;i<512 ;i++, x += dx2)
{
  y = (5.+dy*(max-p[i]));
       if (i==0)
         y = 473;
  /* y = maxy*2/3;*/  /*  long vertical line at f=0 Hz */
  line(x,maxy,x,y);
  /* printf("x=%d dx=%d y=%d maxy=%d  i=%d\r ",x,dx2,y,maxy,i);delay(10);*/
}
                    setviewport(0,0,maxx,20,0);
                    size = imagesize(0,0,maxx,20);
                    /*printf("size=%u ",size);getch();*/
                    buff1 = malloc(size);
                    getimage(0,0,maxx,20,buff1);
                    clearviewport();
                      putimage(0,0,buff1,COPY_PUT);
                      setlinestyle(0,0,1);  /*solidln,0,norwidth */
                      setwritemode(1);      /* xorput */
                      setviewport(0,0,maxx,maxy,0);

                     /* moveto(maxx*0.4,maxy*2/3); /* ? */
                      moveto(100,240);
                      outtext("Press r-right,l-left,q-quit, freq (Hz) = ");
                      x1 = getx();
                      y1 = gety();
```

```
/* printf("x1=%d  y1=%d  ",x1,y1);getch();*/
/* outtext(dis);*/
/* line(0,0,0,maxy); */

/*  xold = dx2;*/
 xold = 0; /* not double arrow some time */

 /* xnew = xold+dx;
 yynew = yyold;
 plotvector(xold,xnew,x1,y1,yyold,yynew,time);
 */
 move = getch();
 do {
   switch (move) {
          case 'l' :xnew = xold-dx2;
               yynew = yyold;
               plotvector(xold,xnew,x1,y1,yyold,yynew,time);
               xold = xnew;
               break;
           case 'r' :xnew = xold+dx2;
               yynew = yyold;
               plotvector(xold,xnew,x1,y1,yyold,yynew,time);
               xold = xnew;
               break;
           case 'u'  :yynew = yyold-10;
               plotvector(xold,xold,x1,y1,yyold,yynew,time);
               yyold = yynew;
               break;
           case 'd'  :yynew = yyold+10;
               plotvector(xold,xold,x1,y1,yyold,yynew,time);
               yyold = yynew;
               break;
```

```
                        }  /* switch */
                         move = getch();
                        } while (move != 'q');
    free(buff1);
    restorecrtmode();
}


void filehandle(int nb,char d)
{
    int i,j;
    char name[10];
    FILE *outfile,*infile;
    if (d== 'w'){
            printf("\n put in your filename to store  ");
            scanf("%s",&name);
            while((outfile = fopen(name,"w"))==0) {
                printf("  give  other filename \n");
                scanf("%s",&name);
            }
            fprintf(outfile,"%e\n",time);
            for (i=0;i<nb;i++)
              fprintf(outfile,"%d %d %d\n",a[i],b[i],c[i]);
            fclose(outfile);
            return;
    }

/* read file */

    printf("\n put in filename to read from   ");
    scanf("%s",&name);
    while((infile=fopen(name,"r"))==0) {
            printf(" wrong name ! put in new name  ");
            scanf("%s",&name);
```

```
    }
    fscanf(infile,"%e",&time);
    for (i=0;i<nb;i++)
            fscanf(infile,"%d %d %d",&a[i],&b[i],&c[i]);
    fclose(infile);
}
void sport(int port,int c)
{
            union REGS r;
            r.x.dx=port;
            r.h.al=c;
            r.h.ah=1;
            int86(0x14,&r,&r);
            if (r.h.ah & 128)
            {
                        printf("\nSend error detected in serial port.\n");
                        exit(1);
            }
}
char rport(int port)
{
            union REGS r;
            while (!(check_stat(port)&256))
                        if (kbhit())
       {
          getch();
          exit(1);
       }
            r.x.dx=port;
            r.h.ah=2;
            int86(0x14,&r,&r);
            if (r.h.ah & 128)
       printf("Read error detected in serial port");
            return r.h.al;
```

```
}
int check_stat(int port)
{
        union REGS r;
        r.x.dx=port;
        r.h.ah=3;
        int86(0x14,&r,&r);
        return r.x.ax;
}
```