

```

/*
   Prognose Turnier 1989
   Edilbert Kirk
*/

#include <string.h>
#include <stdio.h>

#define MAX(a,b) ((a)>(b)?(a):(b))
#define MIN(a,b) ((a)<(b)?(a):(b))

#define WEITE          78

#define Staedte        3
#define Teilnehmer     27
#define Parameter      9
#define Linefeed       10
#define Formfeed       12

short Kyocera = 1;
short LineCount;
short Prognosen_Zahl;

char Wochentag[2][40];
char Strich[80];
char Sterne[80];
char Kopf[80];
char SKopf[2][80];
char Tabellenstrich[80];
char Tabellenkopf[40];
char *Abstand      = "  " ;
char *LR           = " I " ;
char *RR           = "  I" ;
char *LS           = "*  " ;
char *RS           = "  *" ;

short Wettermatrix[10][10] =
/* 0  1  2  3  4  5  6  7  8  9 */
/*-----*/
{{ 0,99,99,99,18,20,20,20,14,14}, /*0 : keine      */
 {99,99,99,99,99,99,99,99,99,99}, /*1 : unbenutzt */
 {99,99,99,99,99,99,99,99,99,99}, /*2 : unbenutzt */
 {99,99,99,99,99,99,99,99,99,99}, /*3 : unbenutzt */
 {14,99,99,99, 0, 2, 8, 8,14,20}, /*4 : Nebel      */
 {14,99,99,99, 2, 0, 2, 8, 8,14}, /*5 : Niesel     */
 {14,99,99,99, 8, 6, 0, 6, 6,10}, /*6 : Regen      */
 {14,99,99,99, 8, 8, 2, 0, 8,20}, /*7 : Schnee     */
 {12,99,99,99,18,12, 8, 8, 0, 2}, /*8 : Schauer    */
 {12,99,99,99,20,14,10,20, 2, 0}}; /*9 : Gewitter   */

short Wertungen[Parameter][2] = {
    { 4,2}, /* Bedeckung      */
    { 9,1}, /* Windrichtung  */
    { 8,1}, /* Windstaerke   */
    {20,1}, /* Wetter        */
    {15,1}, /* Druck         */
    { 5,2}, /* Temperatur    */
    { 5,2}, /* Taupunkt      */
    { 5,2}, /* Minimum-Temperatur */
    {10,1}} ;/* Niederschlag  */

typedef struct TabStruct {
    short Platz      ;
    short Vorplatz   ;
    short Punkte     ;
    short Vorpunkte  ;
    char Name[15]    ;
} Tabelle;

short I,J,K      ;
short Von,Bis    ;
short Person = -1 ;

```

```

short   Zeit           ;
int     Nummer        ;
short   Beobachtung    [Parameter][Staedte ][2];
short   Vorhersage    [Parameter][Teilnehmer][2];
short   Differenz     [Parameter][Teilnehmer][2];
short   Punkte        [Parameter][Teilnehmer][2];
short   Wochenindex   [3][Teilnehmer];
short   Gesamtindex   [3][Teilnehmer];
Tabelle Wochenwertung[3][Teilnehmer];
Tabelle Gesamtwertung[3][Teilnehmer];
char    Stadt         [Staedte][2][15] ;
char    Name          [Teilnehmer][15] ;
short   Abgegeben    [Teilnehmer]      ;
FILE    *Ausgabe      ;
char    Ausgabe_Name[21] ;
char    Extension     [20] ;

int TNUM(Textzahl)
char *Textzahl;
{
    int L;
    sscanf(Textzahl, "%d", &L);
    return(L);
}

/* ===== */
void Initialisierung()
/* ===== */
{
    register short I,J;

    for ( J = 0 ; J < 3 ; J++)
    for ( I = 0 ; I < Teilnehmer ; I++)
    {
        strcpy(Wochenwertung[J][I].Name, &Name[I][0]);
        strcpy(Gesamtwertung[J][I].Name, &Name[I][0]);
    }
}

/* ===== */
void Synops_Lesen()
/* ===== */
{
    FILE *Eingabe;
    char  Eingabe_Name[20];
    char  Zeile[80];
    char  TWert[ 8];
    register short I,J,K;

    strcpy(TWert, " ");
    strcpy(Eingabe_Name, "PROGNO.");
    strcat(Eingabe_Name, Extension);
    printf("\nEingabe von File %s\n", Eingabe_Name) ;
    Eingabe = fopen (Eingabe_Name, "r");
    if (Eingabe == NULL)
    {
        printf ("\nDatei <%s> nicht gefunden \n", Eingabe_Name);
        exit(1);
    }
    fgets (Zeile, 80, Eingabe) ;
    fgets (Zeile, 80, Eingabe) ;
    strncpy(&SKopf[0][2], &Zeile[1], 18);
    strncpy(&Wochentag[0][0], &Zeile[1], 18);
    fgets (Zeile, 80, Eingabe) ;
    strncpy(&SKopf[1][2], &Zeile[1], 18);
    strncpy(&Wochentag[1][0], &Zeile[1], 18);
    for ( I = 0 ; I < 3 ; I++) fgets (Zeile, 80, Eingabe) ;
    for ( J = 0 ; J < Staedte ; J++)
    for ( K = 0 ; K < 2 ; K++)
    {
        fgets (Zeile, 80, Eingabe) ;
        strncpy(&Stadt[J][K][0], &Zeile[1], 14) ;
    }
}

```

```

    for ( I = 0 ; I <Parameter ; I++)
    {
        strncpy(TWert,&Zeile[I*3+17],3);
        Beobachtung[I][J][K] = TNUM(TWert);
    }
}
for ( I = 0 ; I < 3 ; I++) fgets(Zeile,80,Eingabe) ;
for ( J = 0 ; J < Teilnehmer ; J++)
for ( K = 0 ; K < 2 ; K++)
{
    fgets(Zeile,80,Eingabe) ;
    if (K == 0) strncpy(&Name[J][0],&Zeile[1],14);
    for ( I = 0 ; I < Parameter ; I++)
    {
        strncpy(TWert,&Zeile[I*3+17],3);
        if (Vorhersage[I][J][K] = TNUM(TWert)) Abgegeben[J] = 1;
    }
}
fclose( Eingabe ) ;
}

int Tabelle_lesen()
{
    FILE *Eingabe;
    char Eingabe_Name[20];
    char Zeile[80];
    char Alte_Ext[32];
    char TWert[10];
    short J,K;
    int Wert;

    sscanf(Extension,"%d",&Wert);
    sprintf(Alte_Ext,"%03d",Wert-1);
    strcpy(Eingabe_Name,"Tabelle.");
    strcat(Eingabe_Name,Alte_Ext);
    printf("\nEingabe von File %s\n",Eingabe_Name) ;
    Eingabe = fopen (Eingabe_Name,"r");
    if (Eingabe == NULL)
    {
        printf ("\nDatei <%s> nicht gefunden \n",Eingabe_Name);
        if (Nummer > 1) exit(1);
        else return(1);
    }
    for ( K = 0 ; K < 3 ; K++)
    {
        for ( J = 0 ; J < Teilnehmer ; J++)
        {
            fgets (Zeile,80,Eingabe) ;
            stccpy(TWert,&Zeile[2],3);
            Wochenwertung[K][J].Vorplatz = TNUM(TWert) ;
            stccpy(TWert,&Zeile[42],3);
            Gesamtwertung[K][J].Vorplatz = TNUM(TWert) ;
            stccpy(TWert,&Zeile[26],4);
            Wochenwertung[K][J].Vorpunkte = TNUM(TWert) ;
            stccpy(TWert,&Zeile[65],5);
            Gesamtwertung[K][J].Vorpunkte = TNUM(TWert) ;
            Gesamtwertung[K][J].Punkte = Gesamtwertung[K][J].Vorpunkte ;
        }
        fgets (Zeile,80,Eingabe) ;
    }
    fclose ( Eingabe ) ;
    return(0);
}

/* ===== */
void Drucke_Block (Block)
/* ===== */
short Block[Parameter][Teilnehmer][2];
{
    register short I,J;
    double Box;

```

```

if (Kyocera)
{
Box = (Prognosen_Zahl + 1) * 0.425;
fputs("!\R! SCP; MRP 0, -0.1; BOX 16.5, 0.85, V;",Ausgabe);
fprintf(Ausgabe,"BOX 16.5, %5.2f; RPP; FONT 50; EXIT;\n",Box);
fputs(Kopf ,Ausgabe);
fputs("!\R! FONT 8; EXIT;\n",Ausgabe);
}
else
{
fputs(Strich,Ausgabe);
fputs(Kopf ,Ausgabe);
fputs(Strich,Ausgabe);
}
for ( J = 0 ; J < Teilnehmer ; J++)
if (Abgegeben[J] )
{
fprintf( Ausgabe ,"%s%14s      ", LR , Name[J]) ;
for ( I = 0 ; I < Parameter ; I++)
fprintf( Ausgabe ,"%6d", Block[I][J][Zeit]) ;
fprintf (Ausgabe,"%s\n",RR) ;
}
if (Kyocera) fputc(Linefeed,Ausgabe);
else fputs(Strich ,Ausgabe);
}

/* ===== */
void Drucke_Eintrag (Element,P)
/* ===== */
Tabelle *Element;
short P;
{
if (Kyocera && (P == Person)) fputs("!\R! FONT 9; EXIT, E;\n",Ausgabe);
fprintf(Ausgabe,"%s%2d.(%2d) %14s%5d (%4d)%s",
LR,(Element->Platz)+1,Element->Vorplatz,Element->Name,
Element->Punkte,Element->Vorpunkte,RR);
if (Kyocera && (P == Person)) fputs("!\R! FONT 8; EXIT, E;\n",Ausgabe);
}

/* ===== */
void Drucke_Tabelle (K)
/* ===== */
short K;
{
register short I,J;
double Box;

if (Kyocera)
{
Box = (Teilnehmer + 1) * 0.425;
fputs("!\R! SCP; MRP 0, -0.1; BOX 8.1, 0.85, V;",Ausgabe);
fprintf(Ausgabe,"BOX 8.1, %5.2f;",Box);
fputs("MRP 8.4, -0.85; BOX 8.1, 0.85, V;",Ausgabe);
fprintf(Ausgabe,"BOX 8.1, %5.2f; RPP; EXIT;\n",Box);
fprintf(Ausgabe,"%s%s%s\n\n",Tabellenkopf,Abstand,Tabellenkopf) ;
}
else
{
fputs(Tabellenstrich,Ausgabe);
fprintf (Ausgabe,"%s%s%s\n",Tabellenkopf,Abstand,Tabellenkopf) ;
fputs(Tabellenstrich,Ausgabe);
}
for ( I = 0 ; I < Teilnehmer ; I++)
{
J = Wochenindex[K][I] ;
Drucke_Eintrag(&Wochenwertung[K][J],J) ;
fprintf(Ausgabe,"%s",Abstand) ;
J = Gesamtindex[K][I] ;
Drucke_Eintrag(&Gesamtwertung[K][J],J) ;
fputc(Linefeed,Ausgabe);
}
if (Kyocera) fputc(Linefeed,Ausgabe);
}

```

```

    else          fputs(Tabellenstrich,Ausgabe);
}

/* ===== */
void Drucke_Tab20()
/* ===== */
{
    register short I,J,Lim;
    double        Box;

    Lim = Teilnehmer ;
    if (Lim > 21 ) Lim = 20 ;
    if (Kyocera)
    {
        if (Lim < Teilnehmer) Box = (Lim + 2) * 0.425;
        else                 Box = (Lim + 1) * 0.425;
        fputs("!R! SCP; MRP 0, -0.1; BOX 8.1, 0.85, V;",Ausgabe);
        fprintf(Ausgabe,"BOX 8.1, %5.2f;",Box);
        fputs("MRP 8.4, -0.85; BOX 8.1, 0.85, V;",Ausgabe);
        fprintf(Ausgabe,"BOX 8.1, %5.2f; RPP; EXIT;\n",Box);
        fprintf(Ausgabe,"%s%s\n\n",Tabellenkopf,Abstand,Tabellenkopf) ;
    }
    else
    {
        fputs(Tabellenstrich,Ausgabe);
        fprintf (Ausgabe,"%s%s\n",Tabellenkopf,Abstand,Tabellenkopf) ;
        fputs(Tabellenstrich,Ausgabe);
    }
    for ( I = 0 ; I < Lim ; I++)
    {
        J = Wochenindex[2][I] ;
        Drucke_Eintrag(&Wochenwertung[2][J],J) ;
        fprintf(Ausgabe,"%s",Abstand) ;
        J = Gesamtindex[2][I] ;
        Drucke_Eintrag(&Gesamtwertung[2][J],J) ;
        fputc(Linefeed,Ausgabe);
    }
    if (Lim < Teilnehmer)
    {
        if (Wochenwertung[2][Person].Platz < 20) J = Wochenindex[2][Lim];
        else                                     J = Person ;
        Drucke_Eintrag(&Wochenwertung[2][J],J) ;
        fprintf (Ausgabe,"%s",Abstand) ;
        if (Gesamtwertung[2][Person].Platz < 20) J = Gesamtindex[2][Lim];
        else                                     J = Person ;
        Drucke_Eintrag(&Gesamtwertung[2][J],J) ;
        fputc(Linefeed,Ausgabe);
    }
    if (Kyocera) fputc(Linefeed,Ausgabe);
    else        fputs(Tabellenstrich,Ausgabe);
}

/* ===== */
void Intervall()
/* ===== */
{
    short I,J,Wert;
    short Maximum ;
    short Minimum ;
    short Delta ;

    for ( I = 0 ; I < Parameter ; I++)
    {
        Minimum = Beobachtung [I][0][Zeit] ;
        Maximum = Beobachtung [I][0][Zeit] ;
        for ( J = 1 ; J < Staedte ; J++)
        {
            Wert = Beobachtung [I][J][Zeit] ;
            Minimum = MIN (Minimum,Wert) ;
            Maximum = MAX (Maximum,Wert) ;
        }
        for ( J = 0 ; J < Teilnehmer ; J++)

```

```

    {
        Wert = Vorhersage [I][J][Zeit] ;
        Delta = 0 ;
        if (Wert < Minimum ) Delta = Minimum - Wert ;
        if (Wert > Maximum ) Delta = Wert - Maximum ;
        Differenz [I][J][Zeit] = Delta ;
    }
}

void Windrichtung()
{
    short J,Wert      ;
    short Maximum    ;
    short Minimum    ;
    short Diff1      ;
    short Diff2      ;
    short Delta      ;
    short Umlaufend  ;
    short Windstill  ;
    short Vorhanden  ;
    short Nordsektor;

    Minimum = 37 ;
    Maximum = -1 ;
    Umlaufend = 0 ;
    Windstill = 0 ;
    for ( J = 0 ; J < Staedte ; J++)
    {
        Wert = Beobachtung [1][J][Zeit] ;
        if (Wert == 0 ) Windstill = 1 ;
        if (Wert == 99 ) Umlaufend = 1 ;
        if ((Wert > 0) && (Wert <= 36) )
        {
            Minimum = MIN (Wert,Minimum) ;
            Maximum = MAX (Wert,Maximum) ;
        }
    }
    if ((Minimum == 37) && (Maximum > 0) ) Minimum = Maximum ;
    if ((Maximum == -1) && (Minimum < 37) ) Maximum = Minimum ;
    Vorhanden = (Maximum >= Minimum) && ((Maximum - Minimum) != 18) ;
    Nordsektor = (Maximum - Minimum) > 18 ;
    for ( J = 0 ; J < Teilnehmer ; J++)
    {
        Wert = Vorhersage [1][J][Zeit] ;
        Delta = 7 ;
        if (Vorhanden)
        {
            Delta = 0 ;
            if (Nordsektor)
            {
                if ((Wert > Minimum) && (Wert < Maximum) )
                {
                    Diff1 = abs(Wert-Minimum) ;
                    if (Diff1 > 18 ) Diff1 = 36 - Diff1 ;
                    Diff2 = abs(Wert-Maximum) ;
                    if (Diff2 > 18 ) Diff2 = 36 - Diff2 ;
                    Delta = MIN (Diff1,Diff2) ;
                }
            }
            else
            {
                if ((Wert < Minimum) || (Wert > Maximum) )
                {
                    Diff1 = abs (Wert - Minimum) ;
                    Diff2 = abs (Wert - Maximum) ;
                    if (Diff1 > 18 ) Diff1 = 36 - Diff1 ;
                    if (Diff2 > 18 ) Diff2 = 36 - Diff2 ;
                    Delta = MIN (Diff1,Diff2) ;
                }
            }
        }
    }
}

```

```

        if ((Wert = 0) && Windstill ) Delta = 0 ;
        if ((Wert = 99) && Umlaufend ) Delta = 0 ;
        if ((Delta > 5) && (Umlaufend || Windstill) ) Delta = 5 ;
        if ((Delta < 0) ) Delta = 7 ;
        Differenz [1][J][Zeit] = Delta ;
    }
}

void Bewoelkung()
{
    short J,Wert      ;
    short Maximum    ;
    short Minimum    ;
    short Diff1      ;
    short Diff2      ;
    short Delta      ;
    short Nebel      ;
    short Vorhanden  ;

    Minimum = 10 ;
    Maximum = -1 ;
    Nebel = 0 ;
    for ( J = 0 ; J < Staedte ; J++)
    {
        Wert = Beobachtung [0][J][Zeit] ;
        if (Wert == 9 ) Nebel = 1 ;
        if ((Wert != 9) )
        {
            Minimum = MIN (Wert,Minimum) ;
            Maximum = MAX (Wert,Maximum) ;
        }
    }
    if ((Minimum == 10) && (Maximum > -1) ) Minimum = Maximum ;
    if ((Maximum == -1) && (Minimum < 10) ) Maximum = Minimum ;
    Vorhanden = (Maximum >= Minimum) ;
    for ( J = 0 ; J < Teilnehmer ; J++)
    {
        Wert = Vorhersage [0][J][Zeit] ;
        Delta = 3 ;
        if (Vorhanden)
        {
            Delta = 0 ;
            if ((Wert < Minimum) || (Wert > Maximum) )
            {
                Diff1 = abs (Wert - Minimum) ;
                Diff2 = abs (Wert - Maximum) ;
                Delta = MIN (Diff1,Diff2) ;
            }
        }
        if (Wert == 9)
        {
            if (Nebel) Delta = 0 ;
            else      Delta = 3 ;
        }
        if ((Delta > 3) && Nebel ) Delta = 3 ;
        Differenz [0][J][Zeit] = Delta ;
    }
}

/* ===== */
void Wetter()
/* ===== */
{
    short I,J      ;
    short Wert     ;
    short Beob     ;
    short Delta    ;
    short W        ;

    for ( J = 0 ; J < Teilnehmer ; J++)
    {
        Wert = Vorhersage [3][J][Zeit] ;

```

```

Delta = 99 ;
for ( I = 0 ; I < Staedte ; I++)
{
    Beob = Beobachtung[3][I][Zeit] ;
    W     = Beob / 100 ;                               /* ww */
    if (W>0 ) Delta = MIN (Wettermatrix[Wert][W],Delta);
    W     = (Beob - 100 * W) / 10 ;                     /* W1 */
    Delta = MIN (Wettermatrix[Wert][W],Delta);
    W     = Beob - 10 * (Beob / 10) ;                   /* W2 */
    if (W>0 ) Delta = MIN (Wettermatrix[Wert][W],Delta);
}
Differenz[3][J][Zeit] = Delta ;
}
}

/* ===== */
void Punktwertung()
/* ===== */
{
    short I,J ;
    short Wert ;
    short Summe ;

    for ( J = 0 ; J < Teilnehmer ; J++)
    if (Abgegeben[J])
    {
        Summe = 0 ;
        for ( I = 0 ; I < Parameter ; I++)
        {
            Wert = Wertungen[I][0] - Differenz[I][J][Zeit] ;
            if (Wert < 0 ) Wert = 0 ;
            Wert = Wert * Wertungen[I][1] ;
            Punkte[I][J][Zeit] = Wert ;
            Summe += Wert ;
        }
        Wochenwertung[Zeit][J].Punkte = Summe ;
        Gesamtwertung[Zeit][J].Punkte += Summe ;
    }
}

/* ===== */
void Sleepy()
/* ===== */
{
    short J ;
    short Wert ;

    Wert = 0 ;
    Prognosen_Zahl = 0 ;
    for ( J = 0 ; J < Teilnehmer ; J++)
    if (Abgegeben[J])
    {
        Wert += Wochenwertung[Zeit][J].Punkte ;
        Prognosen_Zahl++;
    }
    Wert = Wert / Prognosen_Zahl ;
    for ( J = 0 ; J < Teilnehmer ; J++)
    if (!Abgegeben[J])
    {
        Wochenwertung[Zeit][J].Punkte = Wert ;
        Gesamtwertung[Zeit][J].Punkte += Wert ;
    }
}

/* ===== */
void Plazierung (Wertung,Index)
/* ===== */
Tabelle *Wertung;
short *Index;
{
    short I,J ;
    short Pos ;
}

```



```

for ( J = 0 ; J < Teilnehmer ; J++)
{
    Pos = 0 ;
    for ( I = 0 ; I < Teilnehmer ; I++)
        if ((Wertung+I)->Punkte > (Wertung+J)->Punkte ) Pos++;
        (Wertung+J)->Platz = Pos ;
}
Pos = 0;
for ( J = 0 ; J < Teilnehmer ; J++)
for ( I = 0 ; I < Teilnehmer ; I++)
if ((Wertung+I)->Platz == J ) *(Index + Pos++) = I ;
}

/* ===== */
void Drucke_Synops (SynKopf)
/* ===== */
char *SynKopf;
{
    short I,J;

    if (Kyocera)
    {
        fputs("!\R! SCP; MRP 0, -0.1; BOX 16.5, 0.85, V;",Ausgabe);
        fputs("BOX 16.5, 1.7; RPP; FONT 50; EXIT;\n",Ausgabe);
        fputs(SynKopf ,Ausgabe);
        fputs("!\R! FONT 8; EXIT;\n",Ausgabe);
    }
    else
    {
        fputs(Strich ,Ausgabe);
        fputs(SynKopf,Ausgabe);
        fputs(Strich ,Ausgabe);
    }
    for ( J = 0 ; J < Staedte ; J++)
    {
        fprintf(Ausgabe,"%s%s      ",LR,&Stadt[J][Zeit][0]) ;
        for ( I = 0 ; I < Parameter ; I++)
            fprintf (Ausgabe,"%6d",Beobachtung[I][J][Zeit]) ;
        fprintf (Ausgabe,"%s\n",RR) ;
    }
    if (Kyocera) fputc(Linefeed,Ausgabe);
    else        fputs(Strich,Ausgabe);
}

void Persoenlicher_Ausdruck()
{
    short I;

    if (Kyocera)
    {
        fprintf(Ausgabe,"!\R! FONT 10; MAP 8.25, 1; CTXT'%s', L; EXIT;\n",
                Name[Person]);
    }
    else
    {
        fputs(Sterne,Ausgabe);
        fprintf(Ausgabe,"%s",LS);
        for (I = 0 ; I < 5 ; I++) fprintf (Ausgabe,"%14s",Name[Person]) ;
        fprintf(Ausgabe,"%s\n",RS) ;
        fputs(Sterne,Ausgabe);
    }
    for (Zeit = 0; Zeit < 2 ; Zeit++)
    {
        fputc(Linefeed,Ausgabe);
        Drucke_Synops (&SKopf[Zeit][0]);
        if (Kyocera)
            fputs("!\R! SCP; MRP 0, -0.525; BOX 16.5, 1.7, V;",Ausgabe);
            fputs("BOX 16.5, 0.85; RPP; EXIT, E;\n",Ausgabe);
            fprintf(Ausgabe,"%s%s",LR,"Vorhersage      ") ;
            for (I = 0 ; I < Parameter ; I++)
                fprintf(Ausgabe,"%6d",Vorhersage[I][Person][Zeit]) ;
    }
}

```

```

    fprintf(Ausgabe, "%s\n", RR) ;
    fprintf(Ausgabe, "%s%s", LR, "Abweichungen" );
    for (I = 0 ; I < Parameter ; I++)
    fprintf(Ausgabe, "%6d", Differenz[I][Person][Zeit]) ;
    fprintf(Ausgabe, "%s\n", RR) ;
    fprintf(Ausgabe, "%s%s", LR, "Punkte" );
    for (I = 0 ; I < Parameter ; I++)
    fprintf(Ausgabe, "%6d", Punkte[I][Person][Zeit]) ;
    fprintf(Ausgabe, "%s\n", RR) ;
    if (Kyocera) fputc(Linefeed, Ausgabe);
    else fputs(Strich ,Ausgabe);
    Drucke_Eintrag(&Wochenwertung[Zeit][Person], Person) ;
    fprintf(Ausgabe, "%s", Abstand) ;
    Drucke_Eintrag(&Gesamtwertung[Zeit][Person], Person) ;
    fputc(Linefeed, Ausgabe);
    if (Kyocera) fputc(Linefeed, Ausgabe);
    else fputs(Strich ,Ausgabe);
}
fputc(Linefeed, Ausgabe);
if (Kyocera)
{
    fputs(" !R! FONT 8; SCP; MRP 0, -0.1; BOX 8.1, 0.85;" ,Ausgabe);
    fputs("MRP 8.4, 0; BOX 8.1, 0.85; RPP; EXIT;\n" ,Ausgabe);
}
else fputs(Tabellenstrich, Ausgabe);
fprintf(Ausgabe, "%sWochenwertung" %2d. Runde%s", LR, Nummer, RR);
fprintf(Ausgabe, "%s", Abstand);
fprintf(Ausgabe, "%sGesamtwertung" %2d. Runde%s\n", LR, Nummer, RR);
Drucke_Tab20() ;
if (!Kyocera)
{
    fputs(Sterne, Ausgabe);
    fprintf(Ausgabe, "%s", LS) ;
    for ( I = 0 ; I < 5 ; I++) fprintf (Ausgabe, "%14s", Name[Person]) ;
    fprintf(Ausgabe, "%s\n", RS ) ;
    fputs(Sterne, Ausgabe);
}
fputc(Formfeed, Ausgabe);
}

/* ===== */
void Tabellen_Speicherung()
/* ===== */
{
    register short I, J;
    for (I = 0; I < 3 ; I++)
    {
        for ( J = 0 ; J < Teilnehmer ; J++)
        {
            Drucke_Eintrag(&Wochenwertung[I][J], J) ;
            fputs(Abstand, Ausgabe);
            Drucke_Eintrag(&Gesamtwertung[I][J], J) ;
            fputc(Linefeed, Ausgabe) ;
        }
        fputc(Linefeed, Ausgabe) ;
    }
}

void main()
{
    short J;

    memset(Strich, '-', WEITE);
    Strich[WEITE] = Linefeed;
    memset(Sterne, '*', WEITE);
    Sterne[WEITE] = Linefeed;
    strcpy(Tabellenstrich, Strich);
    Tabellenstrich[38] = ' ';
    Tabellenstrich[39] = ' ';
    strcpy(Kopf, "I Name N dd f");
    strcat(Kopf, "f wWW pp TT Td Tn RR I\n");
    strcpy(Tabellenkopf, "I Nr. (LW) Name Punkte ( LW ) I");
}

```

```

if (Kyocera)
{
    *(LR    )      = ' ';
    *(RR+1 )      = ' ';
    Kopf[ 0]      = ' ';
    Kopf[77]      = ' ';
    Tabellenkopf[ 0] = ' ';
    Tabellenkopf[37] = ' ';
}
strcpy(&SKopf[0][0],Kopf);
strcpy(&SKopf[1][0],Kopf);
printf("Prognose Nr.: ");
scanf("%d",&Nummer) ;
sprintf(Extension,"9%02d",Nummer) ;
printf("Extension = %s\n",Extension);
/*-----*/
Synops_Lesen();
Initialisierung() ;
if (Nummer != 1) Tabelle_lesen();
strcpy(Ausgabe_Name,"LIST.");
strcat(Ausgabe_Name,Extension);
printf("Ausgabe auf File %s\n",Ausgabe_Name) ;
Ausgabe = fopen(Ausgabe_Name,"w");
if (Kyocera)
{
    fputs("!! UNIT C; SLM 2; FTMD 15; FONT 8; EXIT, E;\n",Ausgabe);
}
/*-----*/
for ( Zeit = 0 ; Zeit < 2 ; Zeit++)
{
    if (Kyocera) fputs("!! FONT 52; EXIT, E;\n",Ausgabe);
    fprintf (Ausgabe,"Beobachtungen am %s\n",Wochentag[Zeit]) ;
    Drucke_Synops (&SKopf[Zeit][0]) ;
    Intervall() ;
    Bewoelkung() ;
    Windrichtung() ;
    Wetter() ;
    Punktwertung() ;
    Sleepy() ;
    Plazierung (&Wochenwertung[Zeit],&Wochenindex[Zeit] ) ;
    Plazierung (&Gesamtwertung[Zeit],&Gesamtindex[Zeit] ) ;
    if (Kyocera) fputs("!! FONT 52; EXIT, E;\n",Ausgabe);
    fprintf(Ausgabe,"\n%2d-st\xFCndige Vorhersagen:\n",24*Zeit+24) ;
    Drucke_Block (Vorhersage) ;
    if (Kyocera) fputs("!! FONT 52; EXIT, E;\n",Ausgabe);
    LineCount = 21 + 2 * Prognosen_Zahl;
    if (LineCount > 64)
    {
        fputc(Formfeed,Ausgabe);
        LineCount = 6 + Prognosen_Zahl;
    }
    fprintf(Ausgabe,"\nAbweichungen:\n") ;
    if (Kyocera) fputs("!! FONT 8; EXIT, E;\n",Ausgabe);
    Drucke_Block (Differenz ) ;
    LineCount += 7 + Prognosen_Zahl;
    if (LineCount > 64)
    {
        fputc(Formfeed,Ausgabe);
        LineCount = 7 + Prognosen_Zahl;
    }
    if (Kyocera) fputs("!! FONT 52; EXIT, E;\n",Ausgabe);
    fprintf(Ausgabe,"\nPunkte:\n") ;
    if (Kyocera) fputs("!! FONT 8; EXIT, E;\n",Ausgabe);
    Drucke_Block (Punkte) ;
    fputc(Linefeed,Ausgabe);
    LineCount += 7 + Teilnehmer;
    if (LineCount > 64)
    {
        fputc(Formfeed,Ausgabe);
        LineCount = 7 + Teilnehmer;
    }
    if (Kyocera) fputs("!! FONT 52; EXIT, E;\n",Ausgabe);
}

```

```

fprintf(Ausgabe, "\nTabelle\n" ) ;
if (Kyocera)
{
    fputs("!\R! FONT 8; SCP; MRP 0, -0.1; BOX 8.1, 0.85;", Ausgabe);
    fputs("MRP 8.4, 0; BOX 8.1, 0.85; RPP; EXIT;\n", Ausgabe);
}
else fputs(Tabellenstrich, Ausgabe);

fprintf(Ausgabe, "%s%2d Stunden           Einzelwertung%s", LR, Zeit*24+24, RR);
fprintf(Ausgabe, "%s", Abstand);
fprintf(Ausgabe, "%s%2d Stunden           Gesamtwertung%s\n", LR, Zeit*24+24, RR);
    Drucke_Tabelle (Zeit) ;
    fputs(Formfeed, Ausgabe);
}
for ( J = 0 ; J < Teilnehmer ; J++)
{
    Wochenwertung[2][J].Punkte =
    Wochenwertung[1][J].Punkte + Wochenwertung[0][J].Punkte ;
    Gesamtwertung[2][J].Punkte =
    Gesamtwertung[1][J].Punkte + Gesamtwertung[0][J].Punkte ;
}
Plazierung (&Wochenwertung[2], &Wochenindex[2] ) ;
Plazierung (&Gesamtwertung[2], &Gesamtindex[2] ) ;
if (Kyocera) fputs("!\R! FONT 52; EXIT, E;\n", Ausgabe);
fprintf(Ausgabe, "Tabellen der %2d. Runde\n", Nummer);
if (Kyocera)
{
    fputs("!\R! FONT 8; SCP; MRP 0, -0.1; BOX 8.1, 0.85;", Ausgabe);
    fputs("MRP 8.4, 0; BOX 8.1, 0.85; RPP; EXIT;\n", Ausgabe);
}
else fputs(Tabellenstrich, Ausgabe);
fprintf(Ausgabe, "%sWochenwertung  %2d. Runde           %s", LR, Nummer, RR);
fprintf(Ausgabe, "%s", Abstand);
fprintf(Ausgabe, "%sGesamtwertung  %2d. Runde           %s\n", LR, Nummer, RR);
Drucke_Tabelle (2) ;
fclose(Ausgabe);
strcpy(Ausgabe_Name, "TABELLE.");
strcat(Ausgabe_Name, Extension);
printf("Ausgabe auf File %s\n", Ausgabe_Name);
Ausgabe = fopen(Ausgabe_Name, "w");
Tabellen_Speicherung();
fclose(Ausgabe) ;
strcpy(Ausgabe_Name, "PERS.");
strcat(Ausgabe_Name, Extension);
printf("Ausgabe auf File %s\n", Ausgabe_Name);
Ausgabe = fopen(Ausgabe_Name, "w");
for ( Person = 0 ; Person < Teilnehmer ; Person++)
if (Abgegeben[Person]) Persoenlicher_Ausdruck();
fclose(Ausgabe);
}

```