```
(*
Title   : PRInputP.Pas
Main    : PackReda.Pas
LastEdit: 1992-12-28 17.03 vers 1.00
Author  : Evalds Sport-Data HB, Bo Engborg, 1992-12
System  : Borland Pascal 7.0,
          Stony Brook Pascal+ 6.13i,
          Turbo Power Professional 5.20,
          B-Tree Filer 5.40
*)

unit prInputP;

{$F+,I-,O+}
{.. det g†r ungef„r dubbelt s† fort med $R-}

interface

uses
  TPcrt, filer, TPstring, TPedit, TPentry, TPpick,
  graph,

  Esdtypes, GloDefs, ESmall, ETider, EScreen,
  EError, EPrinter, EConsol, Efiler, Ebild, Emousex,

  prDefs, prMisc, prFilerI, prFilerY, prFilerP, prPick, prEntry;

procedure ErasePackets;
procedure PacketErase;
procedure PacketReg(add: boolean);

implementation

Const
  pcl= 1; pln= 13; pwd= 80; phg= 11; pcmdix= 13;
  ncl= 1; nln= 3; nwd= 80; nhg= 21;
  nnixdeln= 4; ncmdix= (nhg- 3)* nnixdeln;
  xw= 40; yl= 80; zh= 80;

Type
  scrPackArr= record
    rec: Array[1..nhg- 3] Of longint;
    nmn: Array[1..nhg- 3] Of Strng24;
    nn: Array[1..nhg- 3] Of integer;
  end;
  SytterArr= Array[1..mxYtterKartong] Of record
    rec: longint;
    vol: real;
  end;
  PinnerArr= ^innerArr;
  innerArr= Array[1..mxInnerkartong] Of record
    box: PaketRec;
    nn: integer;
  end;
  PspaceArr= ^spaceArr;
  spaceArr= Array[1..zh,1..yl,1..(xw Div 4)] Of byte;
  vviType= Array[1..6] Of integer;

Var
```

```
  tmpScrPack, scrPack: ^scrPackArr;
  lpspage, pspage: integer;
  tmpPacket: ^PacketRec;
  tmpPakets1, tmpPakets2: InPaketArr;
  adding: boolean;
  tmpPakets: InPaketArr;

(*-------------------------------------------------------------------------*)
procedure ErasePackets;
Var
  kRec, i: longint;
  ii: integer;
begin
       kRec:= EUsedRecs(packetF); If (kRec< 1) Then exit;
  MyMsg:= 'Alla pakets„ndningarna kommer att raderas Ok!';
  If ErrorMsg(101) Then begin
    MakeCountW(4,'');  WriteCountW(2,'Skapar nya pakets„ndningsfiler');
    ECloseFileBlock(packetF);
    EcreateFileBlock(Concat(Glo^.dataPathText,'\'+ paketname),SizeOf(PacketRec),0,dummyIID
    EOpenFileBlock(packetF,Concat(Glo^.dataPathText,'\'+ paketname));
    ZoPacket;
    RestoreCountW;
  end;
end;
(*-------------------------------------------------------------------------*)
procedure PacketErase;
Var
  krec: longint;
  nn: integer;
begin
  krec:= -1; escesc:= true;
  If PacketPick('',false,false,false,nn) Then
    If Not(escesc) Then krec:= GetPacketPickItem(1);
  PacketPickExit;
  If (krec> 0) Then
    DelRecSelectPacket(RcRtn,krec,true);
end;
(*-------------------------------------------------------------------------*)
procedure aLokalInfo(Var ESR: ESrecord);
begin
  ESRInfo(ESR);
end;
(*-------------------------------------------------------------------------*)
procedure aLokalInMsg(Var ESR: ESRecord);
Var
  ss: Strng42;
  pp: Strng80;
  ii, kk: integer;
begin
  ss:= ' <F10>= klart, <esc>= avbryt';
  kk:= ESR.CurrentID;
  Case kk Of
    0..ncmdix- 1:
    begin
      kk:= kk Mod nnixdeln;
      If (kk In [255]) Then ss:= ss+ ' <sp>= v„xlar';
      Case kk Of
         0: pp:= 'V„lj innerpaket f”r s„ndningen';
         1: pp:= 'Paketnamn';
```

```
        2: pp:= 'Antal av paketet';
        3: pp:= 'Ta bort paketet ur s„ndningen';
      end;
    end;
  else pp:= ESRInMsg(ESR);
  end;
  FastWrite('',nln+ nhg- 4,ncl+ 2,CtrlAttr);
  InMsg(pp+ ss);
end;
(*-------------------------------------------------------------------------*)
procedure aLokalUtMsg(Var ESR: ESRecord);
Var
  kk, ii: integer;
  rr: float;
  ch: char;
begin
  kk:= ESR.CurrentID;
  Case kk Of
    0..ncmdix- 1:
    begin
      ii:= Succ(kk Div nnixdeln);
      kk:= (kk Mod nnixdeln);
      Case kk Of
        0: ;
        1: ;
        2: ;
        3: ;
      end;
    end;
  else ESRUtMsg(ESR);
  end;
  FastWrite('',nln+ nhg- 4,ncl+ 2,CtrlAttr);
end;
(*-------------------------------------------------------------------------*)
function aValidateLokal(Var ESR: ESrecord; Var ff: word): boolean;
Var
  bb: boolean;
begin
  bb:= false;
  bb:= Not(bb);
  aValidateLokal:= bb;
end;
(*-------------------------------------------------------------------------*)
procedure PacketCount(sgn: integer);
Var
  ii, jj, nn: integer;
  li: longint;
  Exst, FileEnd: boolean;
  rr: real;
  tmpnn: longint;
  sorta: Array[1..mxInPaket] Of real;
begin
  MakeCountW(4,'');
  WriteCountW(2,'V„nta! Ordnar r„knare..');
  FillChar(sorta,SizeOf(sorta),#0);
  nn:= 0; rr:= 0.0; ii:= 0; jj:= 0;
  Repeat
    Inc(ii); li:= Packet.pakets[ii].rec;
    If (li> 0) Then begin
```

```
      jj:= ii;
      nn:= nn+ Packet.pakets[ii].nn;
      FixtSelectInner(RcRtn,li,Exst,FileEnd);
      WriteCountW(3,'Antal:'+ RightNumbStr(ii,4)+
                    '['+ RightNumbStr(mxInPaket,4)+ ']');
      If Exst Then begin
        rr:= rr+ Inner.box.volume* Packet.pakets[ii].nn;
        sorta[ii]:= Inner.Box.volume;
      end;
    end;
  Until (ii>= mxInPaket);
  Packet.npakets:= Packet.npakets+ sgn* nn;
  Packet.innervolume:= Packet.innervolume+ sgn* rr;
  nn:= jj;
  If (sgn= +1) And (nn> 0) Then begin
    WriteCountW(2,'V„nta! Sorterar i ordning!');
    For ii:= 1 To Pred(nn) Do begin
      WriteCountW(3,'Antal:'+ RightNumbStr(ii,4)+
                    ' ['+ RightNumbStr(nn,4)+ ']');
      For jj:= Succ(ii) To nn Do begin
        If (sorta[ii]<= sorta[jj]) Then begin
          rr:= sorta[ii]; sorta[ii]:= sorta[jj]; sorta[jj]:= rr;
          li:= Packet.pakets[ii].rec;
          Packet.pakets[ii].rec:= Packet.pakets[jj].rec;
          Packet.pakets[jj].rec:= li;
          tmpnn:= Packet.pakets[ii].nn;
          Packet.pakets[ii].nn:= Packet.pakets[jj].nn;
          Packet.pakets[jj].nn:= tmpnn;
        end;
      end;
    end;
    tmpPakets2:= Packet.pakets;
  end;
  RestoreCountW;
end;
(*-----------------------------------------------------------------------*)
procedure aReSetLokalPage;
Var
  changed: boolean;
  ct: CompareType;
  ii, kk: integer;
begin
  ct:= CompStruct(tmpScrPack,scrPack,SizeOf(scrPack));
  changed:= (ct <> equal);
  If changed Then begin
    ii:= (Pred(lpspage)* (nhg- 3)); kk:= 0;
    Repeat
      Inc(ii); Inc(kk);
      If (ii<= mxInPaket) Then begin
        Packet.pakets[ii].rec:= scrPack^.rec[kk];
        Packet.pakets[ii].nn:= scrPack^.nn[kk];
      end;
    Until (kk>= nhg- 3);
  end;
  lpspage:= pspage; tmpScrPack^:= scrPack^;
end;
(*-----------------------------------------------------------------------*)
procedure aSetLokalPage;
Var
```

```
  ii, kk: integer;
  is, fe: boolean;
  li: longint;
begin
  ii:= (Pred(pspage)* (nhg- 3)); kk:= 0;
  Repeat
    Inc(ii); Inc(kk);
    If (ii<= mxInPaket) Then
      scrPack^.rec[kk]:= Packet.pakets[ii].rec;
    is:= false;
    li:= scrPack^.rec[kk];
    If (li> 0) Then
      FixtSelectInner(RcRtn,li,is,fe)
    else is:= false;
    If Not(is) Then ZoInner;
    scrPack^.nmn[kk]:= Inner.Lnr+ ' '+ Inner.pnamn;
    scrPack^.rec[kk]:= innern[1];
    scrPack^.nn[kk]:= Packet.pakets[ii].nn;
  Until (kk>= nhg- 3);
  tmpScrPack^:= scrPack^; lpspage:= pspage;
end;
(*-------------------------------------------------------------------------*)
{$V-}
procedure aSetLokalESR(Var ESR: ESrecord);
Var
  kk, jj, ii, y, x: integer;
begin
  ii:= 0; y:= nln; jj:= 0;
  Repeat
    Inc(jj); Inc(y);
    AddNestedField(ESR,'',y,ncl+ 3,'X',y,ncl+ 3,1,ii);
    Inc(ii);
    SetProtection(On);
    AddStringField(ESR,'',y,ncl+ 7,CharStr('X',Pred(SizeOf(scrPack^.nmn[jj]))),
                   y,ncl+ 7,Pred(SizeOf(scrPack^.nmn[jj])),ii,nil,scrPack^.nmn[jj]);
    SetProtection(Off);
    Inc(ii);
    SetNumeric(On);
    AddIntField(ESR,'',y,ncl+ 33,CharStr('9',4),
                y,ncl+ 33,ii,0,9999,scrPack^.nn[jj]);
    SetNumeric(Off);
    Inc(ii);
    AddNestedField(ESR,'',y,ncl+ 40,'X',y,ncl+ 40,1,ii);
  Until (jj>= nhg- 3);
end;
{$V+}
(*-------------------------------------------------------------------------*)
procedure aLokalReg;
Var
  i, j, kk, ii, npages, antal: integer;
  ff: word;
  rr: float;
  krec, trec, ttrec: longint;
  ExitCmd: ESType;
  ESR: ESRecord;
  ESRi: ESRInitRec;
  alla, vv, Exst, FileEnd: boolean;
  nocmds: CommandSet;
  first: boolean;
```

```
begin
  If Not(HeapMemOk(SizeOf(scrPackArr)* 2)) Then begin
    MyMsg:= 'Minnet r„cker inte f”r variabler';
    If ErrorMsg(1) Then;
    exit;
  end;
  New(scrPack); New(tmpScrPack);
  first:= true; pspage:= 1; lpspage:= -1;
  npages:= (mxInPaket Div (nhg- 3));
  With ESRi Do begin
    cmix:= ncmdix;
    cl:= ncl;
    ln:= nln;
    wd:= nwd;
    hg:= nhg;
    editw:= false;
    Error:= nil;
    InMsg:= @aLokalInMsg;
    UtMsg:= @aLokalUtMsg;
    InfoMsg:= @aLokalInfo;
    SetESRuser:= aSetLokalESR;
  end;
  PacketCount(-1);
  nocmds:= [ESuser9];
  SetESR('innerpaket',ESR,ESRi,nocmds);
  tmpPakets2:= Packet.pakets; ff:= 0; aSetLokalPage;
  Repeat
    FastWrite(RightNumbStr(pspage,4)+
              '['+ RightNumbStr(npages,4)+ ']'
              ,nln+ nhg- 2,ncl+ nwd- 11,CtrlAttr);
    Repeat
      exitcmd:= EditScreen(ESR,ff,false);
      ff:= ESR.currentID;
      If (exitcmd = ESnested) Then begin
        Case ff Of
          ncmdix+ 0: exitcmd:= ESdone;
          ncmdix+ 1: exitcmd:= ESquit;
          ncmdix+ 2: exitcmd:= ESnextRec;
          ncmdix+ 3: exitcmd:= ESprevRec;
        end;
      end;
      If (exitcmd <> ESquit) Then begin
        aLokalUtMsg(ESR);
        Case exitcmd Of
          ESnested:
          begin
            kk:= (ff Mod nnixdeln);
            ii:= Succ(ff Div nnixdeln);
            Case kk Of
              0:
              begin
                aReSetLokalPage;
                aSetLokalPage;
                alla:= InnerPick('',true,false,false,antal);
                If Not(escesc) And (antal> 0) Then begin
                  j:= 0;
                  Repeat
                    Inc(j);
                    trec:= GetInnerPickItem(j);
```

```
                If (trec> 0) Then begin
                  Exst:= true; If Not(Exst) Then ZoInner;
                  If (ii> (nhg- 3)) Then begin
                    aReSetLokalPage;
                    If (pspage< npages) Then
                      Inc(pspage)
                    else pspage:= 1;
                    ii:= 1;
                    aSetLokalPage;
                    trec:= GetInnerPickItem(j);
                  end;
                  scrPack^.nmn[ii]:= Inner.Lnr+ ' '+ Inner.pnamn;
                  scrPack^.rec[ii]:= trec;
                  scrPack^.nn[ii]:= 1;
                  For i:= ff To ff+ Pred(nnixdeln) Do
                    DrawField(ESR,i);
                  ff:= Pred(ii)* nnixdeln; Inc(ii);
                end;
              Until (trec< 0);
            end;
            InnerPickExit;
            aReSetLokalPage;
            aSetLokalPage;
            escesc:= false;
          end;
          3:
          begin
            scrPack^.nmn[ii]:= '';
            scrPack^.rec[ii]:= 0;
            scrPack^.nn[ii]:= 0;
            For i:= ff To Pred(nnixdeln) Do DrawField(ESR,i);
            escesc:= false;
          end;
        end;
      end;
      EShelp: ;
      ESuser0, ESdone: ;
      ESnextRec:
      If (pspage< npages) Then
        Inc(pspage)
      else pspage:= 1;
      ESprevRec:
      If (pspage> 1) Then
        Dec(pspage)
      else pspage:= npages;
    end;
  end;
  If (exitcmd In [ESdone,ESuser0,ESnextRec,ESprevRec]) Then vv:= aValidateLokal(ESR,ff
Until ((exitcmd In [ESdone,ESuser0,ESnextRec,ESprevRec]) And vv) Or (exitcmd In [ESqui
If (exitcmd In [ESquit]) Then begin
  Packet.pakets:= tmpPakets2;
end else begin
  aReSetLokalPage;
  aSetLokalPage;
end;
Until (exitcmd In [ESquit,ESdone,ESuser0]) Or escesc;
ESRExit(ESR);
Dispose(tmpscrPack); Dispose(scrPack);
PacketCount(+1);
```

```
end;
(*-------------------------------------------------------------------------*)
procedure LokalInfo(Var ESR: ESrecord);
Var
  wa: byte;
begin
  wa:= WriteAttr;
  FastWrite('Innerpaket:',pln+ 1,pcl+ 22,wa);
  FastWrite('antal:',pln+ 2,pcl+ 22,wa);
  FastWrite('Volym (L):',pln+ 3,pcl+ 22,wa);
  FastWrite('Ytterpaket:',pln+ 5,pcl+ 2,wa);
  FastWrite('Volym (L):',pln+ 6,pcl+ 55,wa);
  FastWrite('Luft (L):',pln+ 7,pcl+ 55,wa);
  ESRInfo(ESR);
end;
(*-------------------------------------------------------------------------*)
procedure LokalInMsg(Var ESR: ESRecord);
Var
  ss: Strng42;
  pp: Strng80;
  ii, kk: integer;
begin
  ss:= ' <F10>= klart, <esc>= avbryt';
  If (ESR.CurrentID In [255]) Then ss:= ss+ ' <sp>= v„xlar';
  Case ESR.CurrentID Of
     0: pp:= 'L”pande nummer.';
     1: pp:= 'Packningsdatum (††††-mm-dd)';
     2:
     begin
       tmpPakets:= Packet.pakets;
       pp:= 'V„lj ingŧende paketstorlekar';
     end;
     3: pp:= 'Mottagare, namn';
     4: pp:= 'Mottagare, adress';
     5: pp:= 'Mottagare, adress';
     6: pp:= 'Mottagare, adressort';
     7: pp:= 'Totalt antal paket';
     8: pp:= 'Total volym av innerpaketen (L)';
     9: pp:= 'L”pande nummer';
    10: pp:= 'Ytterpaketets namn';
    11: pp:= 'Ytterpaketets volym (L)';
    12: pp:= 'Luftvolymen i ytterpaketet (L)';
  else pp:= ESRInMsg(ESR);
  end;
  FastWrite('',pln+ phg- 4,pcl+ 2,CtrlAttr);
  InMsg(pp+ ss);
end;
(*-------------------------------------------------------------------------*)
procedure LokalUtMsg(Var ESR: ESRecord);
Var
  pp: IsamKeyStr;
  rr: real;
  k, f: integer;
begin
  k:= ESR.CurrentId;
  Case k Of
     0: ;
     2:
     begin
```

```
      If (CompStruct(tmpPakets,Packet.pakets,SizeOf(InpaketArr)) <> equal) Then begin
        ZoYtter;
        Packet.ytterrec:= 0;
        Packet.airvolume:= 0;
        DrawField(ESR,7); DrawField(ESR,8);
        For f:= 10 To 13 Do DrawField(ESR,f);
      end;
    end;
  else ESRUtMsg(ESR);
  end;
  FastWrite('',pln+ phg- 4,pcl+ 2,CtrlAttr);
end;
(*-------------------------------------------------------------------------*)
function ValidateLokal(Var ESR: ESrecord; Var ff: word): boolean;
Var
  bb: boolean;
begin
  bb:= false;
  bb:= Not(bb);
  ValidateLokal:= bb;
end;
(*-------------------------------------------------------------------------*)
function tryit(nn, vvx: integer; vvi: vviType; inn: PinnerArr): boolean;
Var
  doneok, done, found, break: boolean;
  i, j, mm, xbyte, xby1, xb1, xb2, xbit1, xbit2: integer;
  z, y, x, xx, yy, zz, zzi: integer;
  ll, ww, hh: real;
  li0, wi0, hi0: integer;
  vv, uu, li, wi, hi: integer;
  space: PspaceArr;
  bb: byte;
  ch: char;
begin
  doneok:= true; break:= false;
  If HeapMemOk(SizeOf(spaceArr)) Then begin
    New(space);
    FillChar(space^,SizeOf(space^),#0);
    doneok:= false;
    ll:= Ytter.box.len; ll:= ll / yl;
    ww:= Ytter.box.width; ww:= ww / xw;
    hh:= Ytter.box.height; hh:= hh / zh;
(*prov    MakeCountW(9); *)
    WriteCountW(9,'<space> stegning, <cr> k”r, <esc> avbryt');
    WriteCountW(3,'Skala'+
                  ' h”jd='+ RightRealStr(hh,4)+
                  ' l„ngd='+ RightRealStr(ll,4)+
                  ' bredd='+ RightRealStr(ww,4));
    i:= 0; done:= true; ch:= 'p'; uu:= 0;
    Repeat
      Inc(i); mm:= inn^[i].nn; j:= 0; zzi:= 0;
      Repeat
        Inc(j); Inc(uu);
        vv:= 0; found:= false;
        Repeat
          Inc(vv);
          WriteCountW(4,'GPass '+ RightNumbStr(vvx,1)+ ' [6]'+
                        ' LPass '+ RightNumbStr(vv,1)+ ' [6]'+
                        ' Packar kartong '+ RightNumbStr(i,3)+
```

```
                ' '+ RightNumbStr(j,3)+
                ' ['+ RightNumbStr(mm,3)+ ']');
        Case vvi[vv] Of
          1:
          begin
            li0:= Trunc(inn^[i].box.len/ ll);
            wi0:= Trunc(inn^[i].box.width/ ww);
            hi0:= Trunc(inn^[i].box.height/ hh);
          end;
          2:
          begin
            li0:= Trunc(inn^[i].box.len/ ll);
            hi0:= Trunc(inn^[i].box.width/ hh);
            wi0:= Trunc(inn^[i].box.height/ ww);
          end;
          3:
          begin
            wi0:= Trunc(inn^[i].box.len/ ww);
            hi0:= Trunc(inn^[i].box.width/ hh);
            li0:= Trunc(inn^[i].box.height/ ll);
          end;
          4:
          begin
            wi0:= Trunc(inn^[i].box.len/ ww);
            li0:= Trunc(inn^[i].box.width/ ll);
            hi0:= Trunc(inn^[i].box.height/ hh);
          end;
          5:
          begin
            hi0:= Trunc(inn^[i].box.len/ hh);
            li0:= Trunc(inn^[i].box.width/ ll);
            wi0:= Trunc(inn^[i].box.height/ ww);
          end;
          6:
          begin
            hi0:= Trunc(inn^[i].box.len/ hh);
            wi0:= Trunc(inn^[i].box.width/ ww);
            li0:= Trunc(inn^[i].box.height/ ll);
          end;
        end;
        WriteCountW(5,'Skala'+
                ' h”jd='+ RightNumbStr(hi0,4)+
                ' l„ngd='+ RightNumbStr(li0,4)+
                ' bredd='+ RightNumbStr(wi0,4));
(* finn ledig volym i spacearray *)
        z:= zzi;
        Repeat
          Inc(z); y:= 0;
          Repeat
            Inc(y); x:= 0;
            Repeat
              Inc(x);
              xbyte:= Succ(Pred(x) Div 4);
              xbit1:= 1 shl ((Pred(x) Mod 4)* 2);
              xbit1:= xbit1+ xbit1 shl 1;
              If ((space^[z,y,xbyte] And xbit1)= 0) Then begin
                WriteCountW(6,'utg†ngspunkt'+
                        ' '+ RightNumbStr(z,2)+
                        ' ['+ RightNumbStr(zh,2)+
```

```
                ] '+ RightNumbStr(y,2)+
                ' ['+ RightNumbStr(yl,2)+
                '] '+ RightNumbStr(x,2)+
                ' ['+ RightNumbStr(xw,2)+ ']');
hi:= Pred(z)+ hi0; If (hi> zh) Then z:= zh;
li:= Pred(y)+ li0; If (li> yl) Then y:= yl;
wi:= Pred(x)+ wi0; If (wi> xw) Then x:= xw;
If (hi<= zh) And (li<= yl) And (wi<= xw) Then begin
  WriteCountW(7,'upptaget?   '+
                ' '+ RightNumbStr(z,2)+
                '->'+ RightNumbStr(hi,2)+
                ' '+ RightNumbStr(y,2)+
                '->'+ RightNumbStr(li,2)+
                ' '+ RightNumbStr(x,2)+
                '->'+ RightNumbStr(wi,2));
  zz:= Pred(z); found:= true;
  Repeat
    Inc(zz); yy:= Pred(y);
    Repeat
      Inc(yy); xx:= Pred(x);
      Repeat
        Inc(xx);
        xbyte:= Succ(Pred(xx) Div 4);
        xbit1:= 1 shl ((Pred(xx) Mod 4)* 2);
        xbit1:= xbit1+ xbit1 shl 1;
        found:= found And
          ((space^[zz,yy,xbyte] And xbit1)= 0);
      Until (xx>= wi) Or Not(found);
    Until (yy>= li) Or Not(found);
  Until (zz>= hi) Or Not(found);
  If found Then begin
    y:= Succ(li); li:= y- li0;
    x:= Succ(wi); wi:= x- wi0;
    z:= Succ(hi); hi:= z- hi0;
    WriteCountW(8,'upptaget!   '+
                  ' '+ RightNumbStr(Pred(z),2)+
                  '<-'+ RightNumbStr(hi,2)+
                  ' '+ RightNumbStr(Pred(y),2)+
                  '<-'+ RightNumbStr(li,2)+
                  ' '+ RightNumbStr(Pred(x),2)+
                  '<-'+ RightNumbStr(wi,2));
    zz:= z;
    Repeat
      Dec(zz); yy:= y;
      Repeat
        Dec(yy); xx:= x;
        Repeat
          Dec(xx);
          xbyte:= Succ(Pred(xx) Div 4);
          xbit1:= 1 shl ((Pred(xx) Mod 4)* 2);
          xbit2:= xbit1 shl 1;
          Case (uu Mod 3) Of
            0: xbit1:= xbit1+ xbit2;
            1: xbit1:= xbit1;
            2: xbit1:= xbit2;
          end;
          space^[zz,yy,xbyte]:=
            space^[zz,yy,xbyte] Or xbit1;
        Until (xx<= wi);
```

```
                    Until (yy<= li);
                  Until (zz<= hi);
                  zzi:= Pred(hi);
                end;
              end;
            end;
          Until (x>= xw) Or found;
        Until (y>= yl) Or found;
      Until (z>= zh) Or found;
      If Not(found) Then zzi:= 0;
    Until (vv>= 6) Or found;
    done:= done And found;
    If KeyPressed Or (ch= ' ') Then begin
      ch:= ReadKey;
      break:= (ch= #27);
    end;
  Until (j>= mm) Or Not(done) Or break;
Until (i>= nn) Or Not(done) Or break;
done:= done And Not(break);
ReadClockDate; TimerStart(1);

If done Then begin
(* grafikprov *)
  pushScreen;
  z:= detect; x:= detect;
  Opengraf(z,x);
  z:= 1;
  hh:= (GetMaxY / 160);
  ww:= (GetMaxX / 120);
  While (z<= zh) Do begin
    For y:= 1 To yl Do begin
      For x:= 1 To xw Do begin
        xbyte:= Succ(Pred(x) Div 4);
        xbit1:= 1 shl ((Pred(x) Mod 4)* 2);
        xbit2:= xbit1 shl 1;
        xx:= x* Trunc(ww)+ y* Trunc(ww);
        yy:= z* Trunc(hh)+ y* Trunc(hh);
        yy:= GetMaxY- yy;
        zz:= 0;
        If (y= yl) Or (y= 1) Then zz:= 1;
        If (x= xw) Or (x= 1) Then zz:= 2;
        If (z= zh) Or (z= 1) Then zz:= 3;
        If (zz> 0) Then PutPixel(xx+ zz,yy+ zz,zz);
        zz:= 0;
        If (z< zh) Or (y< yl) Then begin
          i:= xbit1+ xbit2;
          j:= (space^[z,y,xbyte] And i);
          If (z< zh) Then begin
            zzi:= (space^[Succ(z),y,xbyte] And i);
            If (j <> zzi) Then zz:= 4;
          end;
          If (y< yl) Then begin
            zzi:= (space^[z,Succ(y),xbyte] And i);
            If (j <> zzi) Then zz:= 5;
          end;
        end;
        If (zz> 0) Then PutPixel(xx+ zz,yy+ zz,zz);
        zz:= 0;
        If ((space^[z,y,xbyte] And xbit1)= xbit1) And
```

```
          ((space^[z,y,xbyte] And xbit2)= xbit2) Then
          zz:= 7 else
        If ((space^[z,y,xbyte] And xbit1)= xbit1) Then
          zz:= 13 else
        If ((space^[z,y,xbyte] And xbit2)= xbit2) Then
          zz:= 11;
        If (zz> 0) Then PutPixel(xx,yy,zz);
      end;
    end;
    Inc(z);
  end;
  Gmenu('','');
  Closegraf;
  popScreen;

(*
  z:= 1;
  While (z<= zh) Do begin
    Writeln(lst,'H”jdTv„rsnitt=',z:3,' av 80');
    For y:= 1 To yl Do begin
      For x:= 1 To xw Do begin
        xbyte:= Succ(Pred(x) Div 4);
        xbit1:= 1 shl ((Pred(x) Mod 4)* 2);
        xbit2:= xbit1 shl 1;
        If ((space^[z,y,xbyte] And xbit1)= xbit1) And
          ((space^[z,y,xbyte] And xbit2)= xbit2) Then
          Write(lst,'3') else
        If ((space^[z,y,xbyte] And xbit1)= xbit1) Then
          Write(lst,'1') else
        If ((space^[z,y,xbyte] And xbit2)= xbit2) Then
          Write(lst,'2') else Write(lst,'0');
      end;
      Writeln(lst);
    end;
    Writeln(lst); Writeln(lst);
    Inc(z,10);
  end;
*)
end;
(* endprov *)

    doneok:= done;
    Dispose(space);
  end else begin
    MyMsg:= 'Minnet slut vid tilldelning av variabler! Tillr„cklig koll ej utf”rd!';
    If ErrorMsg(1) Then;
  end;
  tryit:= doneok;
end;
(*-------------------------------------------------------------------------*)
procedure FindBestMatch;
Var
  yrec, li: longint;
  vv, tmpnn, kk, x, y, mm, ii, nn, jj: integer;
  rra, rri, rr: real;
  Exst, done, FileEnd: boolean;
  inn: PInnerArr;
  ytt: SYtterArr;
  vvi: vviType;
```

```
begin
  If Not(HeapMemOk(SizeOf(InnerArr))) Then begin
    MyMsg:= 'Minnesbrist vid variabeltilldelning!';
    If ErrorMsg(1) Then;
    exit;
  end;
  New(inn);
  FillChar(inn^,SizeOf(inn^),#0);
  FillChar(ytt,SizeOf(ytt),#0);
  ReadClockDate; TimerStart(1);
  MakeCountW(10,'');
  WriteCountW(2,'V„nta! Ber„knar innerkartonger..');
  nn:= 0; rr:= 0.0; ii:= 0; jj:= 0; kk:= 0; yrec:= 0;
  Repeat
    Inc(ii); li:= Packet.pakets[ii].rec;
    If (li> 0) Then begin
      jj:= ii;
      nn:= nn+ Packet.pakets[ii].nn;
      FixtSelectInner(RcRtn,li,Exst,FileEnd);
      WriteCountW(3,'Antal:'+ RightNumbStr(ii,4)+
                    '['+ RightNumbStr(mxInPaket,4)+ ']');
      If Exst Then begin
        rr:= rr+ Inner.box.volume* Packet.pakets[ii].nn;
        Inc(kk);
        inn^[kk].box:= Paket.Box;
        inn^[kk].box.positionn:= integer(li);
        inn^[kk].nn:= Packet.pakets[ii].nn;
      end;
    end;
  Until (ii>= mxInPaket);
  Packet.npakets:= nn;
  Packet.innervolume:= rr; rri:= rr;
  rr:= (rr / nn);
  nn:= jj; rra:= 999999999.0; yrec:= 0; mm:= 0;
  If (nn> 0) Then begin
    WriteCountW(2,'V„nta! S”ker ytterkartong..');
    ii:= 0; nn:= Pred(EFileLen(ytterF));
    Repeat
      Inc(ii); li:= ii;
      If (li> 0) Then begin
        FixtSelectYtter(RcRtn,li,Exst,FileEnd);
        If Exst Then begin
          WriteCountW(3,'Antal:'+ RightNumbStr(ii,4)+
                        '['+ RightNumbStr(nn,4)+ ']');
          If (rri<= Ytter.box.volume) Then begin
            Inc(mm);
            ytt[mm].rec:= li;
            ytt[mm].vol:= Ytter.box.volume;
            rr:= Ytter.box.volume- rri;
            WriteCountW(4,'Funnen '+ Ytter.Lnr+ ' "'+ Ytter.pnamn+ '"'+
                          ' luft '+ RightRealStr(rr,4));
          end;
        end;
      end;
    Until FileEnd;
    If (mm> 0) Then begin
      For x:= 1 To Pred(mm) Do begin
        For y:= Succ(x) To mm Do begin
          If (ytt[x].vol>= ytt[y].vol) Then begin
```

```
          li:= ytt[x].rec; ytt[x].rec:= ytt[y].rec; ytt[y].rec:= li;
          rr:= ytt[x].vol; ytt[x].vol:= ytt[y].vol; ytt[y].vol:= rr;
        end;
      end;
    end;
  end;
end;
If (mm> 0) Then begin
  ii:= 0;
  Repeat
    Inc(ii); yrec:= ytt[ii].rec;
    ZoYtter; done:= false;
    If (yrec > 0) Then begin
      FixtSelectYtter(RcRtn,yrec,Exst,FileEnd);
      If Exst Then begin
        rra:= Ytter.box.volume- rri;
        Packet.ytterrec:= yrec;
        Packet.airvolume:= rra;
        WriteCountW(2,'St”rre ytter '+ RightNumbStr(ii,3)+
                      ' ['+ RightNumbStr(mm,3)+ ']'+
                      ' PackProv "'+ Ytter.pnamn+ '"'+
                      ' luft '+ RightRealStr(rra,4));
        vv:= 0;
        Repeat
          Inc(vv);
          Case vv Of
            1:
            begin
              vvi[1]:= 1; vvi[2]:= 2; vvi[3]:= 3;
              vvi[4]:= 4; vvi[5]:= 5; vvi[6]:= 6;
            end;
            2:
            begin
              vvi[1]:= 2; vvi[2]:= 1; vvi[3]:= 3;
              vvi[4]:= 5; vvi[5]:= 4; vvi[6]:= 6;
            end;
            3:
            begin
              vvi[1]:= 3; vvi[2]:= 1; vvi[3]:= 2;
              vvi[4]:= 6; vvi[5]:= 4; vvi[6]:= 5;
            end;
            4:
            begin
              vvi[1]:= 3; vvi[2]:= 2; vvi[3]:= 1;
              vvi[4]:= 6; vvi[5]:= 5; vvi[6]:= 4;
            end;
            5:
            begin
              vvi[1]:= 1; vvi[2]:= 3; vvi[3]:= 2;
              vvi[4]:= 4; vvi[5]:= 6; vvi[6]:= 5;
            end;
            6:
            begin
              vvi[1]:= 2; vvi[2]:= 3; vvi[3]:= 1;
              vvi[4]:= 5; vvi[5]:= 6; vvi[6]:= 4;
            end;
          end;
          done:= tryit(kk,vv,vvi,inn);
        Until done Or (vv>= 6);
```

```
      end;
    end;
    If Not(done) Then yrec:= 0;
  Until (ii>= mm) Or done;
 end;
 If (yrec< 1) Then begin
  MyMsg:= 'Kan ej finna n†gon ytterkartong stor nog i registret!';
  If ErrorMsg(1) Then;
  ZoYtter;
  Packet.ytterrec:= 0;
  Packet.airvolume:= 0;
 end else begin
  MyMsg:= TimerLapS(mmmss,1);
  MyMsg:= 'Packtiden= '+ MyMsg;
  If ErrorMsg(1) Then;
 end;
 RestoreCountW;
 Dispose(inn);
end;
(*-------------------------------------------------------------------------*)
{$V-}
procedure SetLokalESR(Var ESR: ESrecord);
Var
 kk, ii, y, x: integer;
begin
 ii:= 0; y:= pln+ 1;
 SetProtection(On);
 AddStringField(ESR,'',y,pcl+ 2,CharStr('9',Pred(SizeOf(Packet.Lnr))),
                y,pcl+ 2,Pred(SizeOf(Packet.Lnr)),ii,nil,Packet.Lnr);
 SetProtection(Off);
 Inc(ii);
 AddStringField(ESR,'',y,pcl+ 10,CharStr('X',Pred(SizeOf(Packet.Date))),
                y,pcl+ 10,Pred(SizeOf(Packet.Date)),ii,nil,Packet.Date);
 Inc(ii); y:= y+ 2;
 AddNestedField(ESR,'Paket',y,pcl+ 3,'XXXXX',y,pcl+ 3,5,ii);
 y:= pln;
 For kk:= 1 To 4 Do begin
  Inc(ii); Inc(y);
  AddStringField(ESR,'',y,pcl+ 47,CharStr('X',Pred(SizeOf(Packet.adr[kk]))),
                 y,pcl+ 47,Pred(SizeOf(Packet.adr[kk])),ii,nil,Packet.adr[kk]);
 end;
 Inc(ii); y:= pln+ 2;
 SetProtection(On);
 SetNumeric(On);
 AddIntField(ESR,'',y,pcl+ 39,CharStr('9',4),
             y,pcl+ 39,ii,0,9999,Packet.npakets);
 Inc(ii); Inc(y);
 AddRealField(ESR,'',y,pcl+ 34,'#########',
              y,pcl+ 34,ii,1,1,0,Packet.innervolume);
 SetNumeric(Off);
 Inc(ii); y:= pln+ 6;
 AddStringField(ESR,'',y,pcl+ 13,CharStr('X',Pred(SizeOf(Ytter.Lnr))),
                y,pcl+ 3,Pred(SizeOf(Ytter.Lnr)),ii,nil,Ytter.Lnr);
 Inc(ii);
 AddStringField(ESR,'',y,pcl+ 20,CharStr('X',Pred(SizeOf(Ytter.pnamn))),
                y,pcl+ 20,Pred(SizeOf(Ytter.pnamn)),ii,nil,Ytter.pnamn);
 Inc(ii);
 AddRealField(ESR,'',y,pcl+ 65,'#########',
              y,pcl+ 65,ii,1,1,0,Ytter.box.volume);
```

```
  Inc(ii); Inc(y);
  AddRealField(ESR,'',y,pcl+ 65,'#########',
               y,pcl+ 65,ii,1,1,0,Packet.airvolume);
  SetProtection(Off);
end;
{$V+}
(*---------------------------------------------------------------------------*)
procedure PacketReg(add: boolean);
Var
  i, j, k, antal, f: integer;
  ff: word;
  krec, trec, ttrec: longint;
  ExitCmd: ESType;
  ESR: ESRecord;
  ESRi: ESRInitRec;
  vv, Exst, FileEnd, nextprev: boolean;
  nocmds: CommandSet;
  first: boolean;
  tmpPacket: ^PacketRec;
begin
  If Not(HeapMemOk(SizeOf(PacketRec))) Then begin
    MyMsg:= 'Minnet r„cker inte f”r variabler';
    If ErrorMsg(1) Then;
    exit;
  end;
  New(tmpPacket);
  adding:= add; first:= true; nextprev:= false;
  If Not(add) And (EUsedRecs(packetF)= 0) Then begin
    MyMsg:= 'Inga s„ndningar i registret f”r modifiering!';
    If ErrorMsg(1) Then;
    exit;
  end;
  ZoPacket; ZoYtter; ZoInner;
  With ESRi Do begin
    cmix:= pcmdix;
    cl:= pcl;
    ln:= pln;
    wd:= pwd;
    hg:= phg;
    editw:= false;
    Error:= nil;
    InMsg:= @LokalInMsg;
    UtMsg:= @LokalUtMsg;
    InfoMsg:= @LokalInfo;
    SetESRuser:= SetLokalESR;
  end;
  nocmds:= [];
  If add Then nocmds:= nocmds+ [ESnextRec,ESprevRec];
  SetESR('s„ndningsregister',ESR,ESRi,nocmds);
  Repeat
    If add Then begin
      If (EUsedRecs(packetF)>= mxPackets- 1) Then begin
        MyMsg:= 'Maxantalet uppn†tt kan ej addera s„ndning!';
        If ErrorMsg(1) Then ;
        escesc:= true;
      end else begin
        ZoPacket;
        Packet.Lnr:= NumbStr(EFileLen(packetF),Pred(SizeOf(Packet.Lnr)));
        krec:= 0; escesc:= false;
```

```
  end;
end else begin
  If nextprev Then begin
    krec:= GetNextPrev(packetF,nextprev,krec,exitcmd,Packet);
  end else begin
    krec:= -1; escesc:= true;
    If PacketPick('',false,false,false,antal) Then
      If Not(escesc) Then krec:= GetPacketPickItem(1);
    PacketPickExit;
  end;
  If (krec> 0) Then
    DelRecSelectPacket(RcRtn,krec,false);
  ZoYtter;
  If (Packet.ytterrec> 0) Then begin
    FixtSelectYtter(RcRtn,Packet.ytterrec,Exst,FileEnd);
    If Not(Exst) Then ZoYtter;
  end;
end;
If Not(escesc) And (krec>= 0) Then begin
  tmpPacket^:= Packet; tmpPakets1:= Packet.pakets;
  ff:= 1;
  FastWrite(RightNumbStr(kRec,4)+
            '['+ RightNumbStr(EUsedRecs(packetF),4)+ ']'
            ,pln+ phg- 2,pcl+ pwd- 11,CtrlAttr);
  Repeat
    exitcmd:= EditScreen(ESR,ff,false);
    ff:= ESR.currentID;
    If (exitcmd = ESnested) Then begin
      Case ff Of
        pcmdix+ 0: exitcmd:= ESdone;
        pcmdix+ 1: exitcmd:= ESquit;
        pcmdix+ 2: exitcmd:= ESnextRec;
        pcmdix+ 3: exitcmd:= ESprevRec;
        pcmdix+ 4: exitcmd:= ESuser9;
      end;
    end;
    If (exitcmd <> ESquit) Then begin
      Case exitcmd Of
        ESnested:
          Case ff Of
            2:
            begin
              aLokalReg;
              LokalUtMsg(ESR);
            end;
          end;
        ESuser9:
        begin
          FindBestMatch;
          DrawField(ESR,7); DrawField(ESR,8);
          For f:= 10 To 13 Do DrawField(ESR,f);
        end;
        EShelp: ;
        ESuser0, ESdone: ;
      end;
    end;
    nextprev:= (exitcmd In [ESprevRec,ESnextRec]) And Not(adding);
    If (exitcmd In [ESdone,ESuser0]) Or nextprev Then vv:= ValidateLokal(ESR,ff);
  Until (((exitcmd In [ESdone,ESuser0]) Or nextprev) And vv) Or (exitcmd In [ESquit]);
```

```
      If (exitcmd In [ESquit]) Then begin
        Packet:= tmpPacket^;
        Packet.pakets:= tmpPakets1;
      end;
      If (add And Not(exitcmd In [ESquit])) Or Not(add) Then begin
        AddRecSelectPacket(add,krec);
      end;
    end;
  Until (exitcmd In [ESquit]) Or escesc;
  ESRExit(ESR);
  packen[1]:= 0;
  Dispose(tmpPacket);
end;
(*=========================================================================*)
 {$ifdef debug}
begin
  Write('PRINPUTP  ');
 {$endif}
end.
```