



# AKADEMIA OBRONY NARODOWEJ

WYDZIAŁ WOJSK LOTNICZYCH I OBRONY POWIETRZNEJ  
KATEDRA TAKTYKI LOTNICTWA  
CENTRUM INFORMATYKI

~~Do użytku wewnętrznego~~

Egz. pojedynczy

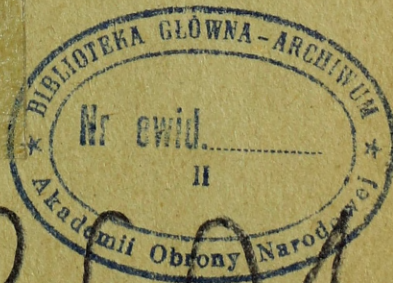
ZASTOSOWANIE KOMPUTEROW SERII IBM PC W PROCESIE  
WYPRACOWANIA DECYZJI O UŻYCIU LOTNICTWA  
MYŚLIWSKO - BOMBOWEGO

Z A Ł A C Z N I K  
P R O G R A M Y U Ż Y T K O W E

Biblioteka Główna  
Akademii Obrony Narodowej  
S/1638



05-001638-001-0



62601

WARSZAWA

1991



# AKADEMIA OBRONY NARODOWEJ

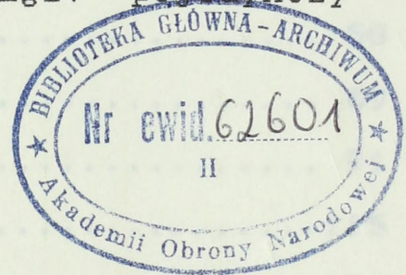
WYDZIAŁ WOJSK LOTNICZYCH I OBRONY POWIETRZNEJ

KATEDRA TAKTYKI LOTNICTWA

CENTRUM INFORMATYKI

- 1. Program STERLN B.....
- 2. Program FELB..... ~~Do użytku wewnętrznego~~
- 3. Program LOT\_BAZ.....
- 4. Program LSPV.....
- 5. Program QSD.....
- 6. Program TPRD.....
- 7. Program WAD.....
- 8. Program.....

Egz. pojedynczy



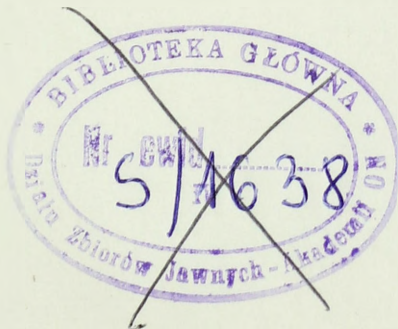
**ZASTOSOWANIE KOMPUTERÓW SERII IBM PC W PROCESIE**

**WYPRACOWANIA DECYZJI O UŻYCIU LOTNICTWA**

**MYSLIWSKO - BOMBOWEGO**

**Z A Ł A C Z N I K**

**P R O G R A M Y U Ż Y T K O W E**

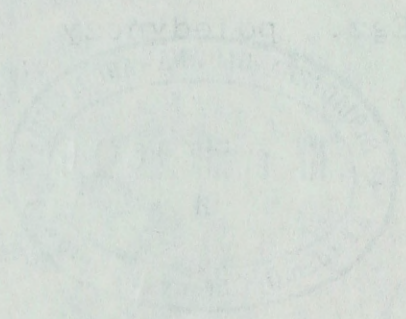
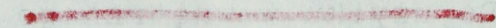


AKADEMIA OBRONY WARSZAWY

WYDZIAŁ WÓDKI I OBRONY POWIERSZNEJ

KATEDRA TAKTYKI I OBRONY

CENTRUM BADAŃ I OBRONY



ZASTOSONOWANIE KONTROLI SERII W W. POKOJIE

WYKONANIE PRACY O WYKONANIU

Opracował zespół:

<i>plk dr</i>	<i>Roman</i>	<i>SZYMANSKI</i>
<i>pplk dr inż.</i>	<i>Zbigniew</i>	<i>KLIMKIEWICZ</i>
<i>mjr mgr. inż.</i>	<i>Wiktor</i>	<i>OLSZOWY</i>

# SPIS TRESCI

1. Program	STERLM_B.....	3
2. Program	PKLB.....	8
3. Program	LOT_BAZ.....	28
4. Program	LSRW.....	50
5. Program	OBUD.....	69
6. Program	TPRD.....	91
7. Program	WARU.....	115
8. Program	OWPOLS.....	138

```
PROGRAM STERLM_B;
```

```
($M 8192,0,0) ($$78);
```

```
uses crt,dos;
```

```
const  
  ilpr=7;                                     {ilosc programow obslugiwanych przez}  
                                              {PROCEDURE Słowniki}
```

```
type  
  nazpr=string[13];                          {integer}  
  tnpr=array[1..ilpr] of nazpr;             {typ tablicy z nazwami programow}
```

```
const  
  tnpr1:tnpr=('PKLB.EXE      ',  
             'LOT_BAZ.EXE   ',  
             'LSRW.EXE      ',  
             'OBUD.EXE      ',  
             'TPRD.EXE      ',  
             'WARU.EXE      ',  
             'OWPOL.S.EXE   ');
```

```
var  
  i,  
  kbl2,  
  pozk2      :integer;  
  zn4        :char;  
  wczytnum,  
  kpmod,  
  kpsys      :boolean;  
  bnum       :array[1..2] of char;  
  lzam2      :byte;
```

```
PROCEDURE BokPion;
```

```
var  
  i:integer;  
Begin  
  for i:=1 to 23 do  
    Begin  
      write(#186);  
      gotoxy(wherex-1,wherey+1);  
    End;  
End;
```

```
PROCEDURE BokPoz;
```

```
var  
  i:integer;  
Begin  
  for i:=1 to 77 do  
    write(#205);  
End;
```

```
PROCEDURE Ramka;
```

```
Begin  
  clrscr;  
  write(#201,#187:78);  
  gotoxy(2,1);
```

```
BokPoz;  
gotoxy(1,25);  
write(#200,#188:78);  
gotoxy(2,25);  
BokPoz;  
gotoxy(1,2);  
BokPion;  
gotoxy(79,2);  
BokPion;  
End;
```

```
PROCEDURE WczytNumer(dkl:integer);  
var  
  ww:byte;  
Begin  
  ww:=0;  
  wczytnum:=false;  
  pozk2:=1;  
  textbackground(white);  
  textcolor(black);  
  lowvideo;  
  FillChar(bnum,dkl,' ');  
  write(copy(bnum,1,dkl));  
  gotoxy(wherex-dkl,wherey);  
  while not wczytnum do  
    Begin  
      zn4:=readkey;  
      case zn4 of  
        #13 : wczytnum:=true;  
        #0  : zn4:=readkey;  
      else  
        Begin  
          if (zn4>='0') and (zn4<='9') then  
            Begin  
              write(zn4);  
              bnum[pozk2]:=zn4;  
              pozk2:=pozk2+1;  
            End;  
          End;  
        end;  
      if pozk2>dkl then  
        Begin  
          ww:=1;  
          gotoxy(wherex-1,wherey);  
          pozk2:=pozk2-1;  
        End;  
      End;  
    textbackground(black);  
    textcolor(white);  
    lowvideo;  
    if ww=0 then  
      pozk2:=pozk2-1;  
    End;  
End;
```

```
PROCEDURE Slowniki;  
Begin  
  kpmod:=false;
```

```
while not kpmođ do
  Begin
    Ramka;
    gotoxy(3,2);
    write('MODEL SYMULACYJNY DZIALAN BOJOWYCH');
    write(' LOTNICTWA MYSLIWSKO-BOMBOWEGO pk IKAR-1');
    gotoxy(10,4);
    write('SLOWNIKI :');
    gotoxy(12,6);
    write('Funkcje :');
    gotoxy(12,8);
    write('1-Klasyfikacja lotnisk bazowania');
    gotoxy(12,9);
    write('2-Lotniska bazowania LMB');
    gotoxy(12,10);
    write('3-Lotnicze srodki walki');
    gotoxy(12,11);
    write('4-Obiekty uderzenia');
    gotoxy(12,12);
    write('5-Taktyczne promienie dzialania samolotow');
    gotoxy(12,13);
    write('6-Warianty uzbrojenia samolotow');
    gotoxy(12,14);
    write('7-Orientacyjna wartosc potrzebnej ogniowej');
    gotoxy(12,15);
    write(' (poligonowej) liczby samolotow M_B');
    gotoxy(12,16);
    write('8-Powrot');
    gotoxy(10,24);
    write('Podaj numer i naciśnij Enter : ');
    WczytNumer(2);
    val(copy(bnum,1,pozsk2),lzam2,kbl2);

    textbackground(black);
    textcolor(white);
    lowvideo;

    case lzam2 of
      1..ilpr:Begin
        Exec('C:\DZLOTM_B\'+tnpri[lzam2],');
        if DosError <>0 then
          Begin
            clrscr;
            writeln('Wystapil blad przy probie uruchomienia'+
              ' programu ',tnpri[lzam2]);
            writeln('Numer bledu = ',DosError,+
              ' Patrz bledy systemu DOS');
            write('Naciśnij klawisz Enter');
            repeat
              zn4:=readkey;
            until zn4=#13;
          End;
        End;
        ilpr+1:kpmođ:=true;
      End;
    End;
  End;
End;
```

```
PROCEDURE ModKal_Dec;
```

```
Begin End;  
End;
```

```
( P O C Z A T E K   P R O G R A M U )
```

```
BEGIN
```

```
  TextMode(2);  
  textbackground(black);  
  textcolor(white);  
  lowvideo;  
  Ramka;  
  gotoxy(13,2);  
  write('A K A D E M I A   O B R O N Y   N A R O D O W E J');  
  gotoxy(19,4);  
  write('Wydział Lotnictwa i Obrony Powietrznej');  
  gotoxy(28,5);  
  write('Centrum Informatyki');  
  gotoxy(20,10);  
  write('MODEL SYMULACYJNY DZIAŁAN BOJOWYCH');  
  gotoxy(22,11);  
  write('LOTNICTWA MYSLIWSKO-BOMBOWEGO');  
  gotoxy(30,12);  
  write('pk I K A R - 1');  
  gotoxy(22,20);  
  write('***ZAPRASZAM DO WSPOLPRACY***');  
  gotoxy(13,22);  
  write('WARSZAWA');  
  gotoxy(55,wherey);  
  write('1991 rok');  
  gotoxy(2,23);  
  for i:=1 to 77 do  
    write(#196);  
    gotoxy(25,24);  
    write('Nacisnij dowolny klawisz...');  
    zn4:=readkey;
```

```
kpsys:=false;  
while not kpsys do
```

```
  Begin
```

```
    Ramka;  
    gotoxy(3,2);  
    write('MODEL SYMULACYJNY DZIAŁAN BOJOWYCH');  
    write(' LOTNICTWA MYSLIWSKO-BOMBOWEGO pk IKAR-1');  
    gotoxy(10,4);  
    write('Funkcje systemowe');  
    gotoxy(10,6);  
    write('1-SLOWNIKI');  
    gotoxy(10,7);  
    write('2-MODUL KALKULACYJNO-DECYZYJNY');  
    gotoxy(10,8);  
    write('3-KONIEC PRACY SYSTEMU');gotoxy(10,24);  
    write('Podaj numer i naciśnij Enter : ');  
    WczytNumer(2);  
    val(copy(bnum,1,poz2),lzam2,kb12);
```

```
  case lzam2 of
```



PROGRAM PKLB;

- 8 -

uses crt,dos,printer;

```
const
  ilpol=7;           {ilosc pol w rekordzie}
  dlnp=31;          {dlugosc nazwy pola}
  mdp=5;            {max dlugosc pola w znakach}
  knp=10;           {nr kolumny-początek nazwy pola}
  kwp=44;           {nr kolumny-początek wartosci pola}
  pwp=4;            {nr wiersza ekranu od ktorego
                    rozpoczyna sie wyswietlanie pol}
  owp=19;           {nr wiersza ekranu na ktorym
                    konczy sie wyswietlanie pol}
  ilwe=owp-pwp+1;   {ilosc wierszy ekranu w ktorych
                    wyswietlane sa pola}
  dlh=4;            {dlugosc hasla}
```

```
type
  nazp=string[dlnp];           {nazwa pola}
  tnazp=array[1..ilpol] of nazp; {typ tablicy z nazwami pol}
  trklb=record                 {typ rekordu z pliku KLLB.LOT}
    kkl :array[1..3] of char;
    dldst:word;
    szdst:byte;
    srddk:word;
    lstr :byte;
    ls_h :byte;
    lobw :byte;
  end;
  tpklb=file of trklb;         {typ pliku}
  t1=string[ilpol];
  t2=array[1..ilpol] of byte;
  t3=array[1..4] of string[22];
  t4=array[1..mdp] of char;
  t5=string[49];
  t6=string[66];
```

```
const
  tnazp1:tnazp=('Kod klasyfikacyjny lotniska',
               'Dlugosc drogi startowej',
               'Szerokosc drogi startowej',
               'Slrednia dlugosc drogi kolowania',
               'Liczba stref rozsrodkowania',
               'Liczba schrono-hangarow',
               'Liczba obwalowan');

  nagl1:t5=' Zestawienie klas lotnisk bazowania';
  nagl2:t6='|KOD |DLUGOSC |SZEROKOSC|SREDNIA DLU-|'+
           '|LICZBA|LICZBA |LICZBA|';
  nagl3:t6='|KLASY |DROGI |DROGI |GOSC DROGI |'+
           '|STREF |SCHRONO-|OBWA- |';
  nagl4:t6='|LOTNISKA|STARTOWEJ|STARTOWEJ|KOLDWANIA |'+
           '|ROZSR.|HANGAROW|LOWAN |';
  nagl5:t6='| 1 | 2 | 3 |'+
           '| 4 | 5 | 6 | 7 |';
  nagl6:t6='|-----|-----|-----|-----|'+
           '|-----|';

  ttp : t1='CWBWBBB'; {tablica typow pol}
```

```

tdp : t2=(3,5,3,5,2,2,2);          - 9 -   {tablica dlugosci pol}
tk  : t3=(' Oczekiwana cyfra      ',      {tablica komunikatow}
        ' Bledny znak            ',
        ' Za duza liczba         ',
        ' Bledna postac liczby   ');

```

```

var
ppole : integer;      {nr pierwszego pola na ekranie}
ostp  : integer;      {nr ostatniego pola na ekranie}
wekrk : integer;      {nr wiersza w oknie wyswietlania pol
                        rekordu ,w ktorym znajduje sie kursor}
pekrk : integer;      {nr pola wyswietlonego na ekranie ,
                        w ktorym znajduje sie kursor}

rob    :integer;
lrek   :integer;      {licznik rekordow}
zrklb1,
zrklb :trklb;        {zmienna rekordowa}
zpklb1,
zpklb :tpklb;        {zmienna plikowa}
whas1,
whas2,
bl,
kpsz,
kprek,
usun,
jestzb,
jestakt,
konwd,
konpr :boolean;
zn1,
zn2,
zn     :char;
wyb    :integer;
milrek :integer;
xk,yk  :integer;
bhas  :array[1..dlh] of char; {bufor na haslo}
buf    : t4;           {bufor dla znakow z klawiatury}
lpol   : integer;     {licznik pol}
pozsk  : integer;     {wskaznik pozycji kursora w "buf"}
wczytzn,
wczytpol : boolean;
nrk    : integer;     {numer komunikatu}
lzam   : longint;    {dla procedury VAL}
kbl    : integer;
lcyfr  : integer;     {licznik cyfr w "buf"}
pcb    : integer;     {pozycja pierwszej cyfry w "buf"}

```

```
PROCEDURE CzytZn;
```

```

Begin
zn:=readkey;
if zn=#0 then
zn:=readkey;
if zn<>#13 then
Begin
write(zn);
gotoxy(wherex-1,wherey);
End;
End;

```

```
PROCEDURE ZnPoziom(var nu:text; m:integer);
```

```
var
  j:integer;

Begin
  for j:=1 to m do
    write(nu, '-');
  End;

PROCEDURE ZnPoz;

var
  j:integer;

Begin
  for j:=1 to 77 do
    write(#196);
  End;

PROCEDURE SzabEkr;
( Rysuje obrys okien )

var
  k,i:integer;
const
  tabl : array[1..5] of integer=(1,3,20,22,25);

PROCEDURE Pion;

var
  i:integer;
Begin
  for i:=1 to 23 do
    Begin
      write(#179);
      gotoxy(wherex-1,wherey+1);
    End;
  End;

Begin
  clrscr;
  write(#218,#191:78);
  gotoxy(1,25);
  write(#192,#217:78);
  gotoxy(1,2);
  Pion;
  gotoxy(79,2);
  Pion;

  for k:=2 to 4 do
    Begin
      gotoxy(1,tabl[k]);
      write(#195,#180:78);
    End;

  for k:=1 to 5 do
    Begin
      gotoxy(2,tabl[k]);
      ZnPoz;
```

```
End;
End;
PROCEDURE PiszNazPol;
( Pisz nazwy pol )
Begin
  case lpol of
    1..ilpol : write(tnazp[lpol]);
  end;
End;
PROCEDURE WczytWartPol;
( Wczytuje wartosci pol )
Begin
  with zrk1b do
    case lpol of
      1:move(buf,kk1,tdp[lpol]);
      2:dldst:=lzam;
      3:szdst:=lzam;
      4:erddk:=lzam;
      5:lstr:=lzam;
      6:ls_h:=lzam;
      7:lobw:=lzam;
    end;
  end;
End;
PROCEDURE StrzLewo;
Begin
  if pozk=1 then
    write(chr(7))
  else
    Begin
      pozk:=pozk-1;
      gotoxy((wherex-1),wherey);
    End;
  End;
PROCEDURE StrzPrawo;
Begin
  if pozk=tdp[lpol] then
    write(chr(7))
  else
    Begin
      pozk:=pozk+1;
      gotoxy((wherex+1),wherey);
    End;
  End;
PROCEDURE KasZn;
( Kasuje znak na ktorym stoi kursor )
Begin
  if (tdp[lpol]-pozk)>0 then
    move(buf[pozk+1],buf[pozk],(tdp[lpol]-pozk));
```

```
buf[tdp[lpol]]:= ' ';  
write(copy(buf,poz, (tdp[lpol]-poz+1)));  
gotoxy((wherex-(tdp[lpol]-poz+1)),wherey);  
End;
```

```
PROCEDURE WypKom(a:integer);  
{ Wypisuje w wierszu 21 komunikat nr 'a',  
wysyla sygnal dzwiekowy,ustawia kursor  
tam gdzie byl przed wywołaniem procedury }
```

```
var  
x,y:integer;  
Begin  
x:=wherex;  
y:=wherey;  
window(2,21,78,21);  
write(tk[a]);  
gotoxy(wherex-1,wherey);  
write(chr(7));  
delay(1000);  
  
textbackground(black);  
textcolor(white);  
lowvideo;  
  
gotoxy(1,1);  
clreol;  
window(2,pwp,78,owp);  
gotoxy(x,y);  
  
textbackground(white);  
textcolor(black);  
lowvideo;  
End;
```

```
PROCEDURE SprPol;  
{ Sprawdza czy wczytany znak jest odpowiedni dla danego typu pola }
```

```
Begin  
bl:=false;  
case ttp[lpol] of  
'C',  
'W',  
'B': if (not(zn in ['0'..'9'])) then  
Begin  
bl:=true;  
nrk:=1;  
End;  
'A': if not((zn>=' ') and (zn<='^')) then  
Begin  
bl:=true;  
nrk:=2;  
End;  
end; {case}  
  
if bl then  
WypKom(nrk);  
End;
```

```
PROCEDURE WypKom1;
```

```
Begin
  if ttp[lpoll] in ['W','B','C'] then
    nrk:=1
  else
    nrk:=2;
  WypKom(nrk);
End;
```

```
PROCEDURE AktRez;
```

```
Begin
  if wyb=3 then
    Begin
      jestakt:=false;
      wczytzn:=true;
    End
  else
    WypKom1;
  End;
```

```
PROCEDURE WczytZnPol;
( Wczytuje znaki do pol )
```

```
Begin
  pozk:=1;
  wczytzn:=false;
  while not wczytzn do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytzn:=true;
        #0 : Begin
              zn:=readkey;
              case zn of
                #19 : AktRez;           ( Alt-R )
                #75 : StrzLewo;
                #77 : StrzPrawo;
                #83 : KasZn;           ( Del )
              end;
            end;
        else
          WypKom1;
        end; {case zn of przy 0}
      End {#0}
    else {case}
      Begin
        SprPol;
        if not bl then
          Begin
            write(zn);
            buf[pozk]:=zn;
            pozk:=pozk+1;
          End;
        if (zn=#0) and (ttp[lpoll]='B') then

```

```
        if pozk>tdp[lipoll] then
            Begin
                gotoxy((wherex-1),wherey);
                pozk:=poz-1;
            End;
        End; {else od case}
    end; {case}
End; {while}
End;
```

PROCEDURE SprWartPol;  
{ Sprawdza czy wczytana wartosc jest odpowiednia dla danego typu pola }

```
var
    i:integer;
    wcyfr:boolean;
Begin
    lcyfr:=0;
    pcb:=1;
    wcyfr:=false;
    case ttp[lipoll] of
        'W',
        'B':Begin
            for i:=1 to tdp[lipoll] do
                Begin
                    if buf[i] in ['0'..'9'] then
                        Begin
                            lcyfr:=lcyfr+1;
                            wcyfr:=true;
                        End;
                    else
                        if not wcyfr then
                            pcb:=pcb+1;
                        End;
                End;
            if lcyfr=0 then
                Begin
                    kbl:=0;
                    lzam:=0;
                End
            else
                val(copy(buf,pcb,lcyfr),lzam,kbl);
            if (kbl=0) then
                Begin
                    if (lzam>65535) and (ttp[lipoll]='W') then
                        Begin
                            nrk:=3;
                            WypKom(nrk);
                            wczytpol:=false;
                        End
                    else
                        wczytpol:=true;
                    if (lzam>255) and (ttp[lipoll]='B') then
```

```
Begin : Klawisz Enter );  
  nrk:=3;  
  WypKom(nrk);  
  zaczytpol:=false;  
End  
else  
  if ttp[lpol]='B' then  
    zaczytpol:=true;  
  End  
else  
  ( Zbiór KLLB.LOT jest pusty );  
  delay(1000);  
  Begin  
    nrk:=4;  
    WypKom(nrk);  
  End;  
End;  
'C',  
'A' : zaczytpol:=true;  
End; { case }  
End;
```

PROCEDURE WczytajPole;

```
Begin  
  zaczytpol:=false;  
  repeat  
    FillChar(buf,tdp[lpol],' ');  
    gotoxy (kwp,wherey);  
    textbackground(white);  
    textcolor(black);  
    write(copy(buf,1,tdp[lpol]));  
    gotoxy(kwp,wherey);  
    WczytZnPol;  
    if (wyb=3) and (jestakt=false) then  
      zaczytpol:=true  
    else  
      SprWartPol;  
  until zaczytpol;  
  
  textbackground(black);  
  textcolor(white);  
  lowvideo;  
End;
```

PROCEDURE BadDosZb;  
( Bada dostep do zbioru KLLB.LOT )

```
Begin  
  window(1,1,80,25);  
  clrscr;  
  assign(zpk1b,'KLLB.LOT');  
  {$I-}  
  reset(zpk1b);  
  {$I+}  
  if IOResult<>0 then  
    Begin  
      gotoxy(20,5);  
      write('Zbiór KLLB.LOT niedostepny');  
      gotoxy(20,wherey+2);  
    End  
  End
```

```
write('Nacisnij klawisz Enter');
readln;
jestzb:=false
end
else
Begin
  if (FileSize(zpk1b)=0) and (wyb<>1) then
    Begin
      gotoxy(10,1);
      write('Zbior KLLB.LOT jest pusty');
      delay(1500);
      jestzb:=false;
      close(zpk1b);
    End
  else
    jestzb:=true;
  End;
End;

PROCEDURE WproRekDanych;
{ Sluzy do wprowadzania danych }

Begin
  BadDosZb;
  if not jestzb then
    Begin
      gotoxy(20,wherey+1);
      write('Czy utworzyc nowy zbior ? : [T/N] ');
      repeat
        CzytZn;
      until upcase(zn) in ['N','T'];

      if upcase(zn)='T' then
        rewrite(zpk1b)
      else
        konpr:=true;
    End;

    if not konpr then
      Begin

        milrek:=filesize(zpk1b);
        seek(zpk1b,milrek);

        konwd:=false;
        SzabEkr;
        gotoxy(2,2);
        write('Wprowadzanie danych do zbioru  KLASA LOTNISK BAZOWANIA');
        gotoxy(2,23);
        write('Enter : Koniec  Del : Kasowanie');
        write('  #27 : Lewo  #26 : Prawo');
        window(2,pwp,78,owp);
        repeat
          textbackground(black);
          textcolor(white);
          lowvideo;
        until true;
      End;
      clrscr;

      for lpol:=1 to ilpol do
```

```
Begin
  gotoxy(knp,wherey);
  PiszNazPol;
  WczytajPole;
  WczytWartPol;
  writeln;
End;
write(zpk1b,zrk1b);

write ('Czy kontynuujesz wprowadzanie ? : [T/N] ');

repeat
  CzytZn;
until upcase(zn) in ['N','T'];
if upcase(zn)='N' then
  konwd:=true;
until konwd=true;
close(zpk1b);
End
else
  konpr:=false;
End;
```

```
PROCEDURE PiszWartPol;
{ Wypisuje na ekranie wartosci pol rekordu }
```

```
Begin
  textbackground(white);
  textcolor(black);
  lowvideo;

  with zrk1b do
    case lpol of
      1:write(kk1:3);
      2:write(dldst:5);
      3:write(szdst:3);
      4:write(srddk:5);
      5:write(lstr:2);
      6:write(ls_h:2);
      7:write(lobw:2);
    end;
  textbackground(black);
  textcolor(white);
  lowvideo;
End;
```

```
PROCEDURE ZaznRek;
{ Zaznacza rekordy do usuniecia }
```

```
Begin
  if wyb=4 then
    Begin
      zrk1b.kk1:=#232#232#232;
      seek(zpk1b,FilePos(zpk1b)-1);
      write(zpk1b,zrk1b);
      kprek:=true;
      usun:=true;
    End;
End;
```

```
End;
gotoxy(knp,wherey);
```

PROCEDURE StrzGora;

```
Begin
  pekrk:=pekrk-1;
  if pekrk<1 then
    Begin
      pekrk:=1;
      write(#7);
    End
  else
    Begin
      if pekrk<ppole then
        Begin
          ppole:=ppole-1;
          ostp:=ostp-1;
          gotoxy(1,1);
          InsLine;
          gotoxy(knp,1);
          lpol:=ppole;
          PiszNazPol;
          gotoxy(kwp,wherey);
          PiszWartPol;
          gotoxy(kwp,wherey);
        End
      else
        Begin
          wekrk:=wekrk-1;
          gotoxy(wherex,wherey-1);
        End;
      End;
    End;
  End;
```

PROCEDURE StrzDol;

```
Begin
  pekrk:=pekrk+1;
  if pekrk>ilpol then
    Begin
      pekrk:=ilpol;
      write(#7);
    End
  else
    Begin
      if pekrk>ostp then
        Begin
          ppole:=ppole+1;
          ostp:=ostp+1;
          gotoxy(1,1);
          DelLine;
          gotoxy(knp,ilwe);
          lpol:=ostp;
          PiszNazPol;
          gotoxy(kwp,wherey);
          PiszWartPol;
          gotoxy(kwp,wherey);
        End
      else
        Begin
          wekrk:=wekrk+1;
          gotoxy(wherex,wherey+1);
        End;
      End;
    End;
  End;
```

```
End;
End;
End;
PROCEDURE AktPol;
Begin
  if wyb=3 then
    Begin
      lpol:=pekrk;
      jestakt:=true;
      WczytajPole;
      if jestakt then
        Begin
          WczytWartPol;
          seek(zpk1b,FilePos(zpk1b)-1);
          write(zpk1b,zrk1b);
        End
      else
        Begin
          gotoxy(kwp,wherey);
          PiszWartPol;
        End;
        gotoxy(kwp,wherey);
      End;
    End;
End;

PROCEDURE PrzeglRek;
Begin
  if ilpol>ilwe then
    rob:=ilwe
  else
    rob:=ilpol;

  kpzb:=false;

  window(2,pwp,78,owp);

  while (not Eof(zpk1b)) and (not kpzb) do
    Begin
      clrscr;
      ppole:=1;
      ostp:=ilwe;
      wekrk:=1;
      pekrk:=1;

      read(zpk1b,zrk1b);
      for lpol:=1 to rob do
        Begin
          gotoxy(knp,1+lpol-1);
          PiszNazPol;
          gotoxy(kwp,wherey);
          PiszWartPol;
          gotoxy(kwp,1);
        End;
        kprek:=false;
        while not kprek do
          Begin
            zn:=readkey;
```

```
case zn of
  #27 : Begin ( Esc )
    kprek:=true;
    kpzb:=true;
  End;
  #13 : AktPol; ( Enter )
  #0 : Begin
    zn:=readkey;
    case zn of
      #22 : ZaznRek; ( Alt-U )
      #72 : StrzGora;
      #80 : StrzDol;
      #81 : kprek:=true; ( PgDn )
    end;
  End; ( #0 )
end;
End; ( while not kprek )
End;
close(zpk1b);
End;

PROCEDURE PrzeglZbioru;
{ Służy do przeglądania zbioru KLASA LOTNISK BAZOWANIA bez możliwości
zmian w danych };
Begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Przeglądanie zbioru KLASA LOTNISK BAZOWANIA ');
      gotoxy(2,23);
      write('Esc : Wyjście PgDn : Następny rekord');
      write('#24 : Góra #25 : Dol');
      PrzeglRek;
    End; ( jestzb )
  End;

PROCEDURE UsuwRekDanych;
{ Usuwa ze zbioru zaznaczone rekordy };
Begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Usuwanie rekordów ze zbioru KLASA LOTNISK BAZOWANIA');
      gotoxy(2,23);
      write('Esc : Wyjście PgDn : Następny rekord Alt-U : Usunięcie');
      gotoxy(2,24);
      write('#24 : Góra #25 : Dol');
      usun:=false;
      PrzeglRek;
      if usun then
        Begin
          assign(zpk1b,'KLLB.LOT');
          reset(zpk1b);
          assign(zpk1b1,'KLLB01.LOT');
```

```
rewrite(zpk1b1);
while not Eof(zpk1b) do
  Begin
    read(zpk1b,zrklb);
    if zrklb.kk1<>#232#232#232 then
      write(zpk1b1,zrklb);
    End;
  close(zpk1b);
  close(zpk1b1);
  erase(zpk1b);
  rename(zpk1b1,'KLLB.LOT');
End;
```

```
End;
End;
```

```
PROCEDURE AktRekDanych;
```

```
Begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Aktualizacja zbioru KLASA LDTNISK BAZOWANIA');
      gotoxy(2,23);
      write('Esc : Wyjscie PgDn : Nastepny rekord Del : Kasowanie ');
      write('Alt-R : Rezygnacja');
      gotoxy(2,24);
      write('Enter : Poczatek,Koniec '#27' : Lewo ');
      write('#26' : Prawo '#24' : Gora '#25' : Dol');
      Przeg1Rek;
    End;
End;
```

```
End;
```

```
PROCEDURE WyprDanych(var nurz:text; turz:char);
```

```
Begin
  writeln(nurz,nagl1);
  writeln(nurz);
  write(nurz,#32);
  ZnPoziom(nurz,64);
  writeln(nurz);
  writeln(nurz,nagl2);
  writeln(nurz,nagl3);
  writeln(nurz,nagl4);
  writeln(nurz,nagl6);
  writeln(nurz,nagl5);
  writeln(nurz,nagl6);

  if upcase(turz)='M' then
    window(1,10,60,25);
    kpzb:=false;
    lrek:=0;

    while (not eof(zpk1b)) and (not kpzb) do
      Begin
        read(zpk1b,zrklb);
        with zrklb do
          Begin
            writeln(nurz,'|',kk1:3,'|',dldst:5,'|',szdst:3,+

```

```
ls_h:2, ' ', lobw:2, ' ');
End;
lrek:=lrek+1;
if (lrek>12) and (upcase(turz)='M') and (FileSize(zpk1b)<>lrek) then
Begin
  writeln;
  write('K-Kontynuacja R-Rezygnacja');
  repeat
    CzytZn;
  until upcase(zn) in ['K','R'];
  if upcase(zn)='K' then
  Begin
    clrscr;
    lrek:=0;
  End
  else
    kpzb:=true;
  End;
End;

if (upcase(turz)='M') and (upcase(zn)<>'R') then
Begin
  write(nurz,#32);
  ZnPoziom(nurz,64);
  writeln;
  writeln;
  write('Koniec zbioru Nacisnij klawisz Enter');
  repeat
    zn:=readkey;
  until zn=#13;
End;
if upcase(turz)='D' then
Begin
  write(nurz,#32);
  ZnPoziom(nurz,64);
  writeln(nurz);
End;
window(1,1,80,25);
End;

PROCEDURE WyprZbioru;

Begin
  BadDosZb;
  if jestzb then
  Begin
    clrscr;
    gotoxy(10,3);
    write('Wyprowadzanie danych ze zbioru KLASA LOTNISK BAZOWANIA');
    gotoxy(10,5);
    write('Podaj urzadzenie wyprowadzania : [D/M] ');
    repeat
      CzytZn;
    until upcase(zn) in ['D','M'];
    zn2:=zn;
    if upcase(zn)='M' then
    Begin
      clrscr;
      WyprDanych(Output,zn2);
    End;
  End;
End;
```

```
End
else
  Begin
    gotoxy(10,7);
    write('Wlacz drukarke i nacisnij Enter');
    repeat
      zn:=readkey;
    until zn=#13;
    WyprDanych(Lst,zn2);
  End;
close(zpk1b);
End;
End;

PROCEDURE Sort1;
var
  i,j,ilrek:integer;
Begin
  ilrek:=FileSize(zpk1b)-1;
  for i:=1 to ilrek do
    for j:=ilrek downto i do
      Begin
        seek(zpk1b,j);
        read(zpk1b,zrklb);
        seek(zpk1b,j-1);
        read(zpk1b,zrklb1);
        if zrklb1.kkl>zrklb.kkl then
          Begin
            seek(zpk1b,j-1);
            write(zpk1b,zrklb);
            write(zpk1b,zrklb1);
          End;
        End;
      End;
    End;
  End;

PROCEDURE Sortowanie;
Begin
  BadDosZb;
  if jestzb then
    Begin
      clrscr;
      writeln('Sortowanie rekordow w zbiorze KLLB.LDT odbywa sie');
      writeln('wedlug kodu klasyfikacyjnego lotniska');
      writeln;
      write(' ':15,'Nacisnij klawisz Enter');
      repeat
        zn:=readkey;
      until zn=#13;
      Sort1;
      close(zpk1b);
    End;
  End;

PROCEDURE WczytHaslo;
var
  whaslo:boolean;
  Begin
```

```
Begin
  whaslo:=false;
  pozk:=1;textbackground(white);
  textcolor(black);
  lowvideo;
  FillChar(bhas,dlh,' ');
  write(bhas);
  gotoxy(wherex-dlh,wherey);
  while not whaslo do
    Begin
      zn:=readkey;
      case zn of
        #13 : whaslo:=true;
        #0  : zn:=readkey;
      else
        Begin
          if (zn>=' ') and (zn<='z') then
            Begin
              write(zn);
              bhas[pozk]:=zn;
              pozk:=pozk+1;
            End;
          End;
        end;
      if pozk>dlh then
        Begin
          gotoxy(wherex-1,wherey);
          pozk:=pozk-1;
        End;
      End;
    textbackground(black);
    textcolor(white);
    lowvideo;
  End;
```

PROCEDURE Haslo;

```
const
  has1:string[dlh]='SZYM';
```

```
var
  rob,
  robl,
  hour,
  min,
  sec,
  s100 :word;
  kbl   :integer;
```

```
Begin
  whas1:=false;
  whas2:=false;
  clrscr;
  gotoxy(10,3);
  write('Podaj haslo i naciśnij klawisz Enter : ');
  WczytHaslo;
  if has1=bhas then
    whas1:=true
  else
    Begin
```

```
GetTime(hour,min,sec,s100);
Val(copy(bhas,1,1),rob,kbl);
if kbl=0 then
  Begin
    robl:=min div 10;
    if robl=rob then
      Begin
        Val(copy(bhas,2,1),rob,kbl);
        if kbl=0 then
          Begin
            if rob=0 then
              Begin
                Val(copy(bhas,3,2),rob,kbl);
                if kbl=0 then
                  Begin
                    if hour=rob then
                      whas2:=true;
                  End;
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End;
End;
```

```
if (not whas1) and (not whas2) then
  Begin
    gotoxy(10,5);
    write('Bledne haslo Nacisnij klawisz Enter');
    readln;
    konpr:=true;
  End;
```

End;

PROCEDURE DostepPelny;

```
Begin
  clrscr;
  gotoxy(20,5);
  write('KLASA LOTNISK BAZOWANIA');
  gotoxy(22,7);
  write('1.Wprowadzanie');
  gotoxy(22,8);
  write('2.Przeglądanie');
  gotoxy(22,9);
  write('3.Aktualizacja');
  gotoxy(22,10);
  write('4.Usuwanie');
  gotoxy(22,11);
  write('5.Wyprowadzanie');
  gotoxy(22,12);
  write('6.Sortowanie');
  gotoxy(22,13);
  write('7.Wyjście');
  gotoxy(22,15);
  write('Wybierz numer: ');
```

```
repeat
  CzytZn;
until zn in ['1'..'7'];
```

```
wyb:=ord(zn)-48;
lowvideo;
case wyb of
  1:WproRekDanych;
  2:PrzeGlZbioru;
  3:AktRekDanych;
  4:UsuwRekDanych;
  5:WyprZbioru;
  6:Sortowanie;
  7:konpr:=true;
End; (case)
End;
```

```
PROCEDURE DostepOgr;
Begin
  clrscr;
  gotoxy(20,5);
  write('KLASA LOTNISK BAZOWANIA');
  gotoxy(22,7);
  write('1.Przeglądanie');
  gotoxy(22,8);
  write('2.Wyprowadzanie');
  gotoxy(22,9);
  write('3.Sortowanie');
  gotoxy(22,10);
  write('4.Wyjście');
  gotoxy(22,12);
  write('Wybierz numer: ');

  repeat
    CzytZn;
  until zn in ['1'..'4'];

  wyb:=ord(zn)-48;

  case wyb of
    1:PrzeGlZbioru;
    2:WyprZbioru;
    3:Sortowanie;
    4:konpr:=true;
  End; (case)
End;
```

( P O C Z A T E K      P R O G R A M U )

```
BEGIN
  TextMode(2);
  textbackground(black);
  textcolor(white);
  lowvideo;
  konpr:=false;
  Haslo;
  while not konpr do
    Begin
      window(1,1,80,25);
      textbackground(black);
```

```

CORNA LOT textcolor(white);
          lowvideo;

          if whas1 then
            Dostep0gr
          else
            DostepFelny;
          End; (while)
END.

          (liczba pol w rekordzie)
          (dlugosc nazwy pola)
          (max dlugosc pola w rekordzie)
          (pozycja-poczetek nazwy pola)
          (pozycja-poczetek wartosci pola)
          (nr wiersza ekranu od ktorego
           rozpoczyna sie wyswietlanie pol)
          (nr wiersza ekranu na ktorym
           konczy sie wyswietlanie pol)
          (liczba wierszy ekranu w ktorych
           wyswietlane sa pola)
          (dlugosc hasla)

          (nazwa pola)
          (typ tablicy z nazwami pol)
          (typ rekordu z nazwa LOTNISKA BAZOWANIA)
          wrd: byte;
          naz1: array[1..12] of char;
          naz2: array[1..3] of char;
          w1: string[1];
          w2: string[1];
          w3: word;
          w4: word;
          wrd;
          (typ pilki)

          (stringi pol)
          t1: array[1..11] of byte;
          t2: array[1..4] of string[2];
          t3: array[1..4] of char;
          t4: string[4];
          t5: string[4];
          t6: string[4];
          t7: string[4];

          (nazwa lotniska bazowania)
          (nazwa lotniska bazowania)
          (kod klasyfikacyjny lotniska)
          (wspolrzadna X polozenia sredka lotniska)
          (wspolrzadna Y polozenia sredka lotniska)
          (kierunek nr1 kursu startu-ladownia)
          (kierunek nr2 kursu startu-ladownia)

          nag1:15= 'lotnisko bazowania';
          nag2:16= 'KLASA WSPOLRZEDNE POLOZE-';
          nag3:16= 'KIERUNEK KURSU-';
          nag4:16= 'LOT- I NAZWA LOTNISKI SRODKA LOTNISKI-';
          nag5:16= 'STARTU-LADOWNIA';
          nag6:16= 'LOTNISKI LOTNISKI';
          nag7:16= '1 2 3 4 5 6';
          nag8:16= '1 2 3 4 5 6';

```

```
PROGRAM LOT_BAZ;
```

```
uses crt,dos,printer;
```

```
const
```

```
  ilpol=7;           {ilosc pol w rekordzie}
  dlnp=39;           {dlugosc nazwy pola}
  mdp=12;            {max dlugosc pola w znakach}
  knp=10;            {nr kolumny-początek nazwy pola}
  kwp=52;            {nr kolumny-początek wartosci pola}
  pwp=4;             {nr wiersza ekranu od ktorego
                    rozpoczyna sie wyswietlanie pol}
  owp=19;            {nr wiersza ekranu na ktorym
                    konczy sie wyswietlanie pol}
  ilwe=owp-pwp+1;   {ilosc wierszy ekranu w ktorych
                    wyswietlane sa pola}
  dlh=4;             {dlugosc hasla}
```

```
type
```

```
  nazp=string[dlnp];           {nazwa pola}
  tnazp=array[1..ilpol] of nazp; {typ tablicy z nazwami pol}
  trklb=record                  {typ rekordu z pliku LOTNISKA BAZOWANIA}
    nr1 :byte;
    naz1 :array[1..12] of char;
    kkl  :array[1..3] of char;
    xl   :longint;
    yl   :longint;
    ks_l1:word;
    ks_l2:word;
  end;
  tpkib=file of trklb;         {typ pliku}

  t1=string[ilpol];           {zawieszona tablica}
  t2=array[1..ilpol] of byte; {zawieszona tablica}
  t3=array[1..4] of string[22]; {zawieszona tablica}
  t4=array[1..mdp] of char;   {zawieszona tablica}
  t5=string[44];              {zawieszona tablica}
  t6=string[64];              {zawieszona tablica}
  t15=string[45];             {zawieszona tablica}
  t16=string[58];             {zawieszona tablica}
```

```
const
```

```
  tnazp1:tnazp=('Numer lotniska bazowania',
               'Nazwa lotniska bazowania',
               'Kod klasyfikacyjny lotniska',
               'Wspolrzedna X polozenia srodka lotniska',
               'Wspolrzedna Y polozenia srodka lotniska',
               'Kierunek nr1 kursu startu-ladowania',
               'Kierunek nr2 kursu startu-ladowania');

  nagl1:t5='          Zestawienie lotnisk bazowania';
  nagl2:t6='!NUMER!          !KLASA!WSPOLRZEDNE POLOZE-!'+
           'KIERUNEK KURSU  !';
  nagl3:t6='!LOT- ! NAZWA          !LOT- !NIA SRODKA LOTNISKA!'+
           'STARTU-LADOWANIA !';
  nagl4:t6='!NISKA! LOTNISKA  !NISKA!-----!'+
           '-----!';
  nagl5:t6='!      !      !      !      X      !      Y      !'+
           '      !      !      !      !      !      !';
  nagl6:t6='! 1  !  !      2  !      ! 3  !      4  !      5  !'+
           '      !      !      !      !      !      !';
```

```
6 | 7 | ;  
nagl7:t6=' |-----|-----|-----|-----|'+  
|-----|';
```

```
nagl11:t15=' Wykaz lotnisk bazowania o klasie ' ;  
nagl12:t16=' ;NUMER: |WSPOLRZEDNE POLOZE-|KIERUNEK KURSU |'  
nagl13:t16=' ;LOT- | NAZWA |NIA SRODKA LOTNISKA|STARTU-LADOWANIA |'  
nagl14:t16=' ;NISKA| LOTNISKA |-----|-----|'  
nagl15:t16=' | | | X | Y | 1 | 2 |'  
nagl16:t16=' | 1 | 2 | 3 | 4 | 5 | 6 |'  
nagl17:t16=' |-----|-----|-----|-----|-----|'
```

```
ttp : t1='BACLLWW'; {tablica typow pol}  
tdp : t2=(3,12,3,5,5,3,3); {tablica dlugosci pol}  
tk : t3=(' Oczekiwana cyfra ', {tablica komunikatow}  
' Bledny znak ',  
' Za duza liczba ',  
' Bledna postac liczby ');
```

```
var  
ppole : integer; {nr pierwszego pola na ekranie}  
ostp : integer; {nr ostatniego pola na ekranie}  
wekrk : integer; {nr wiersza w oknie wyswietlania pol  
rekordu ,w ktorym znajduje sie kursor}  
pekrk : integer; {nr pola wyswietlonego na ekranie ,  
w ktorym znajduje sie kursor}  
rob : integer;  
lrek : integer; {licznik rekordow}  
zrk1b, zrk1b : trklb; {zmienna rekordowa}  
zpk1b, zpk1b : tpk1b; {zmienna plikowa}  
whas1, whas2, {wskazniki hasel}  
bl, kpzb, {koniec przegladania zbioru}  
kprek, {koniec przegladania rekordu}  
usun, jestzb, jestakt, konwd, konpr : boolean;  
zn1, zn2, zn3, zn : char;  
wyb : integer;  
milrek : integer;  
xk,yk : integer;  
bhas : array[1..dlh] of char; {bufor na haslo}  
buf : t4; {bufor dla znakow z klawiatury}  
bufkl:array[1..3] of char;  
lpol : integer; {licznik pol}  
pozsk : integer; {wskaznik pozycji kursora w "buf"}  
wczytkl, wczytzn, wczytpol : boolean;
```

```
nrk : integer;           {numer komunikatu}
lzm : longint;          {dla procedury VAL}
kbl : integer;
lcyfr: integer;         {licznik cyfr w "buf"}
pcb : integer;         {pozycja pierwszej cyfry w "buf"}
```

```
PROCEDURE WczytKlucz(dkl:integer);
```

```
Begin
```

```
  wczytkl:=false;
  pozk:=1;textbackground(white);
  textcolor(black);
  lowvideo;
  FillChar(bufkl,dkl,' ');
  write(copy(bufkl,1,dkl));
  gotoxy(wherex-dkl,wherey);
  while not wczytkl do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytkl:=true;
        #0  : zn:=readkey;
      else
        Begin
          if (zn>=' ') and (zn<='~') then
            Begin
              write(zn);
              bufkl[pozk]:=zn;
              pozk:=pozk+1;
            End;
          End;
        end;
        if pozk>dkl then
          Begin
            gotoxy(wherex-1,wherey);
            pozk:=pozk-1;
          End;
        End;
      textbackground(black);
      textcolor(white);
      lowvideo;
    End;
```

```
PROCEDURE CzytZn;
```

```
Begin
```

```
  zn:=readkey;
  if zn=#0 then
    zn:=readkey;
  if zn<>#13 then
    Begin
      write(zn);
      gotoxy(wherex-1,wherey);
    End;
  End;
```

```
PROCEDURE ZnPoziom(var nu:text; m:integer);
```

```
var  
  j:integer;
```

```
Begin  
  for j:=1 to m do  
    write(nu, '-');  
End;
```

```
PROCEDURE ZnPoz;
```

```
var  
  j:integer;
```

```
Begin  
  for j:=1 to 77 do  
    write(#196);  
End;
```

```
PROCEDURE SzabEkr;  
( Rysuje obrys okien )
```

```
var  
  k,i:integer;  
const  
  tabl : array[1..5] of integer=(1,3,20,22,25);
```

```
PROCEDURE Pion;
```

```
var  
  i:integer;  
Begin  
  for i:=1 to 23 do  
    Begin  
      write(#179);  
      gotoxy(wherex-1,wherey+1);  
    End;  
End;
```

```
Begin  
  clrscr;  
  write(#218,#191:78);  
  gotoxy(1,25);  
  write(#192,#217:78);  
  gotoxy(1,2);  
  Pion;  
  gotoxy(79,2);  
  Pion;  
  
  for k:=2 to 4 do  
    Begin  
      gotoxy(1,tabl[k]);  
      write(#195,#180:78);  
    End;  
  
  for k:=1 to 5 do  
    Begin  
      gotoxy(2,tabl[k]);  
      ZnPoz;  
    End;
```

End;

PROCEDURE PiszNazPol;  
{ Pisz nazwy pol }

Begin  
  case lpol of  
    1..ilpol : write(tnazpi[lpol]);

  end;  
End;

PROCEDURE WczytWartPol;  
{ Wczytuje wartosci pol }

Begin  
  with zrk1b do  
    case lpol of  
      1:nrl:=lzam;  
      2:move(buf,nazl,tdp[lpol]);  
      3:move(buf,kkl,tdp[lpol]);  
      4:xl:=lzam;  
      5:yl:=lzam;  
      6:ks\_l1:=lzam;  
      7:ks\_l2:=lzam;  
    end;

End;

PROCEDURE StrzLewo;

Begin  
  if pozk=1 then  
    write(chr(7))  
  else  
    Begin  
      pozk:=poz-1;  
      gotoxy((wherex-1),wherey);  
    End;

End;

PROCEDURE StrzPrawo;

Begin  
  if pozk=tdp[lpol] then  
    write(chr(7))  
  else  
    Begin  
      pozk:=poz+1;  
      gotoxy((wherex+1),wherey);  
    End;

End;

PROCEDURE KasZn;  
{ Kasuje znak na ktorym stoi kursor }

Begin  
  if (tdp[lpol]-poz)>0 then  
    move(buf[poz+1],buf[poz],(tdp[lpol]-poz));  
    buf[tdp[lpol]]:= ' ';

```
write(copy(buf,pozka,(tdp[lpol]-pozka+1)));  
gotoxy((wherex-(tdp[lpol]-pozka+1)),wherey);  
End;
```

```
PROCEDURE WypKom(a:integer);  
{ Wypisuje w wierszu 21 komunikat nr 'a',  
wysyla sygnal dzwiekowy,ustawia kursor  
tam gdzie byl przed wywołaniem procedury }
```

```
var  
x,y:integer;  
Begin  
x:=wherex;  
y:=wherey;  
window(2,21,78,21);  
write(tk[a]);  
gotoxy(wherex-1,wherey);  
write(chr(7));  
delay(1000);  
  
textbackground(black);  
textcolor(white);  
lowvideo;  
  
gotoxy(1,1);  
clreol;  
window(2,pwp,78,owp);  
gotoxy(x,y);  
  
textbackground(white);  
textcolor(black);  
lowvideo;  
End;
```

```
PROCEDURE SprPol;  
{ Sprawdza czy wczytany znak jest odpowiedni dla danego typu pola }
```

```
Begin  
bl:=false;  
case ttp[lpol] of  
'L':  
'C':  
'W':  
'B': if (not(zn in ['0'..'9'])) then  
Begin  
bl:=true;WypKom(1);  
nrk:=1; End; (case zn of przy 0)  
End;  
'A': if not((zn>=' ') and (zn<='~')) then  
Begin  
bl:=true;  
nrk:=2; End;  
End; if not bl then  
end; (case)  
  
if bl then  
WypKom(nrk);  
End;
```

```
PROCEDURE WypKom1;
Begin
  if ttp[1pol] in ['W','B','L','C'] then
    nrk:=1
  else
    nrk:=2;
  WypKom(nrk);
End;

PROCEDURE AktRez;
Begin
  if wyb=3 then
    Begin
      jestakt:=false;
      wczytzn:=true;
    End
  else
    WypKom1;
End;

PROCEDURE WczytZnPol;
( Wczytuje znaki do pol )
Begin
  pozk:=1;
  wczytzn:=false;
  while not wczytzn do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytzn:=true;
        #0 : Begin
              zn:=readkey;
              case zn of
                #19 : AktRez;           ( Alt-R )
                #75 : StrzLewo;
                #77 : StrzPrawo;
                #83 : KasZn;           ( Del )
              end
            End
          else
            WypKom1;
          End; (case zn of przy 0)
        End {#0}
      else (case)
        Begin
          SprPol;
          if not bl then
            Begin
              write(zn);
              buf[pozk]:=zn;
              pozk:=pozk+1;
            End
          End;
        End;
      End;
    End;
  End;
End;
```

```
        if pozk>tdp[lipol] then
            Begin
                gotoxy((wherex-1),wherey);
                pozk:=poz-1;
            End;
        End; {else od case}
    End; {case}
End; {while}
```

End;

```
PROCEDURE SprWartPol;
{ Sprawdza czy wczytana wartosc jest odpowiednia
dla danego typu pola }
```

```
var
    i:integer;
    wcyfr:boolean;
```

Begin

```
    lcyfr:=0;
    pcb:=1;
    wcyfr:=false;
```

```
case ttp[lipol] of
```

```
    'L',
    'W',
```

```
    'B' :Begin
```

```
        for i:=1 to tdp[lipol] do
```

```
            Begin
```

```
                if buf[i] in ['0'..'9'] then
```

```
                    Begin
```

```
                        lcyfr:=lcyfr+1;
```

```
                        wcyfr:=true;
```

```
                    End;
```

```
                if not wcyfr then
```

```
                    pcb:=pcb+1;
```

```
            End;
```

```
        if lcyfr=0 then
```

```
            Begin
```

```
                kbl:=0;
```

```
                lzam:=0;
```

```
            End
```

```
        else
```

```
            val(copy(buf,pcb,lcyfr),lzam,kbl);
```

```
        if (kbl=0) then
```

```
            Begin
```

```
                if (lzam>65535) and (ttp[lipol]='W') then
```

```
                    Begin
```

```
                        nrk:=3;
```

```
                        WypKom(nrk);
```

```
                        wczytpol:=false;
```

```
                    End
```

```
                else
```

```
        wczytpol:=true;
    if (lzam>255) and (ttp[lpoll]='B') then
        Begin
            nrk:=3;
            WypKom(nrk);
            wczytpol:=false;
        End
    else
        if ttp[lpoll]='B' then
            wczytpol:=true;
        End
    else
        Begin
            nrk:=4;
            WypKom(nrk);
        End;
    End;
    'C',
    'A' : wczytpol:=true;
End; { case }
End;
```

PROCEDURE WczytajPole;

```
Begin
    wczytpol:=false;
    repeat
        FillChar(buf,tdp[lpoll],' ');
        gotoxy (kwp,wherey);
        textbackground(white);
        textcolor(black);
        write(copy(buf,1,tdp[lpoll]));
        gotoxy(kwp,wherey);
        WczytZnPol;
        if (wyb=3) and (jestakt=false) then
            wczytpol:=true
        else
            SprWartPol;
    until wczytpol;

    textbackground(black);
    textcolor(white);
    lowvideo;
End;
```

PROCEDURE BadDoszB;

{ Bada dostep do zbioru LOTBAZ.LOT }

```
Begin
    window(1,1,80,25);
    clrscr;
    assign(zpk1b,'LOTBAZ.LOT');
    {$I-}
    reset(zpk1b);
    {$I+}
    if IOResult<>0 then
        Begin
```

```
    gotoxy(20,5);
    write('Zbior LOTBAZ.LOT niedostepny');
    gotoxy(20,wherey+2);
    write('Nacisnij klawisz Enter');
    readln;
    jestzb:=false
end
else
  Begin
    if (FileSize(zpk1b)=0) and (wyb<>1) then
      Begin
        gotoxy(10,1);
        write('Zbior LOTBAZ.LOT jest pusty');
        delay(1500);
        jestzb:=false;
        close(zpk1b);
      End
    else
      jestzb:=true;
    End;
  End;
End;

PROCEDURE WproRekDanych;
{ Siuzy do wprowadzania danych }

Begin
  BadDosZb;
  if not jestzb then
    Begin
      gotoxy(20,wherey+1);
      write('Czy utworzyc nowy zbior ? : [T/N] ');
      repeat
        CzytZn;
      until upcase(zn) in ['N','T'];

      if upcase(zn)='T' then
        rewrite(zpk1b)
      else
        konpr:=true;
    End;

    if not konpr then
      Begin

        milrek:=filesize(zpk1b);
        seek(zpk1b,milrek);

        konwd:=false;
        SzabEkr;
        gotoxy(2,2);
        write('Wprowadzanie danych do zbioru LOTNISKA BAZOWANIA');
        gotoxy(2,23);
        write('Enter : Koniec Del : Kasowanie');
        write(' #27 : Lewo #26 : Prawo');
        window(2,pwp,78,owp);
        repeat
          textbackground(black);
          textcolor(white);
          lowvideo;
        end repeat
      End;
    End;
  End;
End;
```

```
clrscr;

for lpol:=1 to ilpol do
  Begin
    gotoxy(knp,wherey);
    PiszNazPol;
    WczytajPole;
    WczytWartPol;
    writeln;
  End;
write(zpklb,zrklb);

write ('Czy kontynuujesz wprowadzanie ? : [T/N] ');

repeat
  CzytZn;
until upcase(zn) in ['N','T'];
if upcase(zn)='N' then
  konwd:=true;
until konwd=true;
close(zpklb);
End
else
konpr:=false;
End;
```

```
PROCEDURE PiszWartPol;
{ Wypisuje na ekranie wartosci pol rekordu }
```

```
Begin
  textbackground(white);
  textcolor(black);
  lowvideo;

  with zrklb do
    case lpol of
      1:write(nr1:3);
      2:write(naz1:12);
      3:write(kk1:3);
      4:write(x1:5);
      5:write(y1:5);
      6:write(ks_l1:3);
      7:write(ks_l2:3);
    end;
  textbackground(black);
  textcolor(white);
  lowvideo;
End;
```

```
PROCEDURE ZaznRek;
{ Zaznacza rekordy do usuniecia }
```

```
Begin
  if wyb=4 then
    Begin
      zrklb.kk1:=#232#232#232;
      seek(zpklb,FilePos(zpklb)-1);
      write(zpklb,zrklb);
      kprek:=true;
      usun:=true;
    End;
```

```
End;  
End;  
PROCEDURE StrzGora;  
Begin  
  pekrk:=pekrk-1;  
  if pekrk<1 then  
    Begin  
      pekrk:=1;  
      write(#7);  
    End  
  else  
    Begin  
      if pekrk<ppole then  
        Begin  
          ppole:=ppole-1;  
          ostp:=ostp-1;  
          gotoxy(1,1);  
          InsLine;  
          gotoxy(knp,1);  
          lpol:=ppole;  
          PiszNazPol;  
          gotoxy(kwp,wherey);  
          PiszWartPol;  
          gotoxy(kwp,wherey);  
        End  
      else  
        Begin  
          wekrk:=wekrk-1;  
          gotoxy(wherex,wherey-1);  
        End;  
      End;  
    End;  
End;  
End;
```

```
PROCEDURE StrzDol;  
Begin  
  pekrk:=pekrk+1;  
  if pekrk>ilpol then  
    Begin  
      pekrk:=ilpol;  
      write(#7);  
    End  
  else  
    Begin  
      if pekrk>ostp then  
        Begin  
          ppole:=ppole+1;  
          ostp:=ostp+1;  
          gotoxy(1,1);  
          DelLine;  
          gotoxy(knp,ilwe);  
          lpol:=ostp;  
          PiszNazPol;  
          gotoxy(kwp,wherey);  
          PiszWartPol;  
          gotoxy(kwp,wherey);  
        End  
      else  
        End;  
      End;  
    End;  
  End;  
End;
```

```
Begin
  wekrk:=wekrk+1;
  gotoxy(wherex,wherey+1);
End;
End;
End;
PROCEDURE AktPol;
Begin
  if wyb=3 then
    Begin
      lpol:=pekrk;
      jestakt:=true;
      WczytajPole;
      if jestakt then
        Begin
          WczytWartPol;
          seek(zpk1b,FilePos(zpk1b)-1);
          write(zpk1b,zrk1b);
        End
      else
        Begin
          gotoxy(kwp,wherey);
          PiszWartPol;
        End;
        gotoxy(kwp,wherey);
      End;
    End;
End;
PROCEDURE PrzeglRek;
Begin
  if ilpol>ilwe then
    rob:=ilwe
  else
    rob:=ilpol;
  kpzb:=false;
  window(2,pwp,78,owp);
  while (not Eof(zpk1b)) and (not kpzb) do
    Begin
      clrscr;
      ppole:=1;
      ostp:=ilwe;
      wekrk:=1;
      pekrk:=1;
      read(zpk1b,zrk1b);
      for lpol:=1 to rob do
        Begin
          gotoxy(knp,1+lpol-1);
          PiszNazPol;
          gotoxy(kwp,wherey);
          PiszWartPol;
          gotoxy(kwp,1);
        End;
      End;
    End;
  End;
End;
```

```
kprek:=false;
while not kprek do
  Begin
    zn:=readkey;
    case zn of
      #27 : Begin
        kprek:=true;
        kpzb:=true;
      End;
      #13 : AktPol;
      #0 : Begin
        zn:=readkey;
        case zn of
          #22 : ZaznRek;
          #72 : StrzGora;
          #80 : StrzDol;
          #81 : kprek:=true;
        end;
      End;
    end;
  End;
End;
close(zpk1b);
End;
```

PROCEDURE PrzeglZbioru;

{ Służy do przeglądania zbioru LOTNISKA BAZOWANIA bez możliwości zmian w danych }

```
Begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Przeglądanie zbioru LOTNISKA BAZOWANIA ');
      gotoxy(2,23);
      write('Esc : Wyjście PgDn : Następny rekord');
      write(' #24 : Góra #25 : Dol');
      PrzeglRek;
    End;
  End;
End;
```

PROCEDURE UsuwRekDanych;

{ Usuwa ze zbioru zaznaczone rekordy }

```
Begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Usuwanie rekordów ze zbioru LOTNISKA BAZOWANIA');
      gotoxy(2,23);
      write('Esc : Wyjście PgDn : Następny rekord Alt-U : Usunięcie');
      gotoxy(2,24);
      write('#24 : Góra #25 : Dol');
      usun:=false;
      PrzeglRek;
    End;
  End;
End;
```

```
if usun then
  Begin
    assign(zpk1b,'LOTBAZ.LOT');
    reset(zpk1b);
    assign(zpk1b1,'LOTBAZ01.LOT');
    rewrite(zpk1b1);
    while not Eof(zpk1b) do
      Begin
        read(zpk1b,zrklb);
        if zrklb.kk1<>#232#232#232 then
          write(zpk1b1,zrklb);
        End;
      close(zpk1b);
      close(zpk1b1);
      erase(zpk1b);
      rename(zpk1b1,'LOTBAZ.LOT');
    End;
  End;
End;

PROCEDURE AktRekDanych;

Begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Aktualizacja zbioru LOTNISKA BAZOWANIA');
      gotoxy(2,23);
      write('Esc : Wyjscie PgDn : Nastepny rekord Del : Kasowanie ');
      write('Alt-R : Rezygnacja');
      gotoxy(2,24);
      write('Enter : Poczatek,Koniec '#27' : Lewo ');
      write('#26' : Prawo '#24' : Gora '#25' : Dol');
      PrzeglRek;
    End;
  End;

PROCEDURE WyprDanych(var nurz:text; turz:char);

Begin
  if zn3='1' then
    Begin
      writeln(nurz,nagl1);
      writeln(nurz);
      write(nurz,#32);
      ZnPoziom(nurz,62);
      writeln(nurz);
      writeln(nurz,nagl2);
      writeln(nurz,nagl3);
      writeln(nurz,nagl4);
      writeln(nurz,nagl5);
      writeln(nurz,nagl7);
      writeln(nurz,nagl6);
      writeln(nurz,nagl7);
    end
  else
    Begin
      write(nurz,nagl11);
    End;
  End;
End;
```

```
writeln(nurz,bufkl:3);
writeln(nurz);
write(nurz,#32);
ZnPoziom(nurz,56);
writeln(nurz);
writeln(nurz,nagl12);
writeln(nurz,nagl13);
writeln(nurz,nagl14);
writeln(nurz,nagl15);
writeln(nurz,nagl17);
writeln(nurz,nagl16);
writeln(nurz,nagl17);
  End;

if upcase(turz)='M' then
  window(1,11,80,25);
  kpzb:=false;
  lrek:=0;

  while (not eof(zpk1b)) and (not kpzb) do
    Begin
      read(zpk1b,zrk1b);
      with zrk1b do
        Begin
          if zn3='1' then
            Begin
              writeln(nurz,'1',nrl:3,'1',nazl:12,'1',kkl:3,'1',xl:5,+
                '1',yl:5,'1',ks_11:3,'1',ks_12:3,'1');
              lrek:=lrek+1;
            End
          else
            Begin
              if kkl=bufkl then
                Begin
                  writeln(nurz,'1',nrl:3,'1',nazl:12,'1',xl:5,'1',+
                    yl:5,'1',ks_11:3,'1',ks_12:3,'1');
                  lrek:=lrek+1;
                End;
            End;
          End;
        End;
      if (lrek>11) and (upcase(turz)='M') and (FileSize(zpk1b)<>lrek) then
        Begin
          writeln;
          write('K-Kontynuacja R-Rezygnacja');
          repeat
            CzytZn;
          until upcase(zn) in ['K','R'];
          if upcase(zn)='K' then
            Begin
              clrscr;
              lrek:=0;
            End
          else
            kpzb:=true;
          End;
        End;
      End;
    End;

  if (upcase(turz)='M') and (upcase(zn)<>'R') then
```

```
Begin
  write(nurz,#32);
  if zn3='1' then
    ZnPoziom(nurz,62)
  else
    ZnPoziom(nurz,56);
  writeln;
  writeln;
  write('Koniec zbioru  Nacisnij klawisz Enter');
  repeat
    zn:=readkey;
  until zn=#13;
End;
if upcase(turz)='D' then
Begin
  write(nurz,#32);
  if zn3='1' then
    ZnPoziom(nurz,62)
  else
    ZnPoziom(nurz,56);
  writeln(nurz);
End;
window(1,1,80,25);
end;

PROCEDURE WyprZbioru;

begin
  BadDosZb;
  if jestzb then
  Begin
    clrscr;
    gotoxy(15,1);
    write('Wyprowadzanie danych ze zbioru LOTNISKA BAZOWANIA');
    gotoxy(10,3);
    write('1.Wszystkie lotniska bazowania. ');
    gotoxy(10,4);
    write('2.Wybrana klasa lotnisk bazowania. ');
    gotoxy(10,6);
    write('Wybierz numer: ');
    repeat
      CzytZn;
    until zn in ['1','2'];
    zn3:=zn;
    if zn3='2' then
      Begin
        gotoxy(10,8);
        write('Podaj klase lotnisk bazowania i nacisnij Enter : ');
        WczytKlucz(3);
        End;
        gotoxy(10,11);
        write('Podaj urządzenie wyprowadzania : [D/M] ');
        repeat
          CzytZn;
        until upcase(zn) in ['D','M'];
        zn2:=zn;
        if upcase(zn)='M' then
          Begin
            clrscr;
```

```
    WyprDanych(Output,zn2);  
End  
else  
  Begin  
    gotoxy(10,13);  
    write('Wlacz drukarke i nacisnij Enter');  
    repeat  
      zn:=readkey;  
    until zn=#13;  
    WyprDanych(Lst,zn2);  
  End;  
close(zpk1b);  
End;  
End;
```

```
PROCEDURE Sort1;
```

```
var  
  i,j,ilrek:integer;  
Begin  
  ilrek:=FileSize(zpk1b)-1;  
  for i:=1 to ilrek do  
    for j:=ilrek downto i do  
      Begin  
        seek(zpk1b,j);  
        read(zpk1b,zrklb);  
        seek(zpk1b,j-1);  
        read(zpk1b,zrklb1);  
        if zrklb1.nrl>zrklb.nrl then  
          Begin  
            seek(zpk1b,j-1);  
            write(zpk1b,zrklb);  
            write(zpk1b,zrklb1);  
          End;  
        End;  
      End;  
    End;  
  End;
```

```
PROCEDURE Sortowanie;
```

```
Begin  
  BadDosZb;  
  if jestzb then  
    Begin  
      clrscr;  
      writeln('Sortowanie rekordow w zbiorze LOTBAZ.LOT odbywa sie');  
      writeln('wedlug numeru lotniska bazowania');  
      writeln;  
      write(' ':15,'Nacisnij klawisz Enter');  
      repeat  
        zn:=readkey;  
      until zn=#13;  
      Sort1;  
      close(zpk1b);  
    End;  
  End;  
End;
```

```
PROCEDURE WczytHaslo;
```

```
var
```

```
whaslo:boolean;  
  
Begin  
  whaslo:=false;  
  pozk:=1;textbackground(white);  
  textcolor(black);  
  lowvideo;  
  FillChar(bhas,dlh,' ');  
  write(bhas);  
  gotoxy(wherex-dlh,wherey);  
  while not whaslo do  
    Begin  
      zn:=readkey;  
      case zn of  
        #13 : whaslo:=true;  
        #0  : zn:=readkey;  
      else  
        Begin  
          if (zn>=' ') and (zn<='~') then  
            Begin  
              write(zn);  
              bhas[pozk]:=zn;  
              pozk:=pozk+1;  
            End;  
          End;  
        end;  
      if pozk>dlh then  
        Begin  
          gotoxy(wherex-1,wherey);  
          pozk:=pozk-1;  
        End;  
      End;  
    textbackground(black);  
    textcolor(white);  
    lowvideo;  
  End;
```

PROCEDURE Haslo;

```
const  
  has1:string[dlh]='SZYM';  
var  
  rob,  
  rob1,  
  hour,  
  min,  
  sec,  
  s100 :word;  
  kbl  :integer;
```

```
Begin  
  whas1:=false;  
  whas2:=false;  
  clrscr;  
  gotoxy(10,3);  
  write('Podaj haslo i nacisnij klawisz Enter : ');  
  WczytHaslo;  
  if has1=bhas then  
    whas1:=true
```

```
else
  Begin
    GetTime(hour,min,sec,s100);
    Val(copy(bhas,1,1),rob,kbl);
    if kbl=0 then
      Begin
        robi:=min div 10;
        if robi=rob then
          Begin
            Val(copy(bhas,2,1),rob,kbl);
            if kbl=0 then
              Begin
                if rob=0 then
                  Begin
                    Val(copy(bhas,3,2),rob,kbl);
                    if kbl=0 then
                      Begin
                        if hour=rob then
                          whas2:=true;
                      End;
                    End;
                  End;
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End;
End;
```

```
if (not whas1) and (not whas2) then
  Begin
    gotoxy(10,5);
    write('Bledne haslo  Nacisnij klawisz Enter');
    readln;
    konpr:=true;
  End;
```

End;

PROCEDURE DostepPelny;

```
Begin
  clrscr;
  gotoxy(20,5);
  write('LOTNISKA BAZDWANIA');
  gotoxy(22,7);
  write('1.Wprowadzanie');
  gotoxy(22,8);
  write('2.Przeglądanie');
  gotoxy(22,9);
  write('3.Aktualizacja');
  gotoxy(22,10);
  write('4.Usuwanie');
  gotoxy(22,11);
  write('5.Wyprowadzanie');
  gotoxy(22,12);
  write('6.Sortowanie');
  gotoxy(22,13);
  write('7.Wyjście');
  gotoxy(22,15);
  write('Wybierz numer: ');
```

```
repeat
  until not konpr do
  CzytZn;
```

```
until zn in ['1'..'7'];
```

```
wyb:=ord(zn)-48;
```

```
case wyb of  
  1:WproRekDanych;  
  2:PrzeGlZbioru;  
  3:AktRekDanych;  
  4:UsuwRekDanych;  
  5:WyprZbioru;  
  6:Sortowanie;  
  7:konpr:=true;
```

```
End; {case}
```

```
End;
```

```
PROCEDURE DostepOgr;
```

```
begin
```

```
  clrscr;
```

```
  gotoxy(20,5);
```

```
  write('LOTNISKA BAZOWANIA');
```

```
  gotoxy(22,7);
```

```
  write('1.Przeglądanie');
```

```
  gotoxy(22,8);
```

```
  write('2.Wyprowadzanie');
```

```
  gotoxy(22,9);
```

```
  write('3.Sortowanie');
```

```
  gotoxy(22,10);
```

```
  write('4.Wyjście');
```

```
  gotoxy(22,12);
```

```
  write('Wybierz numer: ');
```

```
  repeat
```

```
    CzytZn;
```

```
  until zn in ['1'..'4'];
```

```
  wyb:=ord(zn)-48;
```

```
case wyb of  
  1:PrzeGlZbioru;  
  2:WyprZbioru;  
  3:Sortowanie;  
  4:konpr:=true;
```

```
End; {case}
```

```
End;
```

```
( P O C Z A T E K      P R O G R A M U )
```

```
BEGIN
```

```
  TextMode(2);
```

```
  textbackground(black);
```

```
  textcolor(white);
```

```
  lowvideo;
```

```
  konpr:=false;
```

```
  Haslo;
```

```
  while not konpr do
```

```
    Begin
```

```
window(1,1,80,25);  
textbackground(black);  
textcolor(white);  
lowvideo;
```

```
if whasl then  
  DostepOgr  
else  
  DostepPelny;
```

```
End; (while)
```

END.

```
PROGRAM LSRW;
uses crt,dos,printer;

const
  ilpol=2;           {ilosc pol w rekordzie}
  dlnp=30;          {dlugosc nazwy pola}
  mdp=17;           {max dlugosc pola w znakach}
  knp=10;           {nr kolumny-początek nazwy pola}
  kwp=42;           {nr kolumny-początek wartosci pola}
  pwp=4;            {nr wiersza ekranu od ktorego
                    rozpoczyna sie wyswietlanie pol}
  owp=19;           {nr wiersza ekranu na ktorym
                    konczy sie wyswietlanie pol}
  ilwe=owp-pwp+1;  {ilosc wierszy ekranu w ktorych
                    wyswietlane sa pola}
  dlh=4;            {dlugosc hasla}

type
  nazp=string[dlnp];           {nazwa pola}
  tnazp=array[1..ilpol] of nazp; {typ tablicy z nazwami pol}
  trklb=record                 {typ rekordu z pliku LOTNICZE SRODKI WALKI}
    nazlsw :array[1..17] of char;
    klsw   :integer;
  end;
  tpklb=file of trklb;         {typ pliku}

  t1=string[ilpol];
  t2=array[1..ilpol] of byte;
  t3=array[1..4] of string[22];
  t4=array[1..mdp] of char;
  t5=string[64];
  t6=string[31];

const
  tnazp1:tnazp=('Nazwa lotniczego srodka walki ',
               'Kod lotniczego srodka walki ');

  nagl1:t5='          Zestawienie lotniczych srodkow walki';
  nagl2:t6='!LOTNICZE SRODKI WALKI!KOD LSW!';
  nagl3:t6='!          1          | 2          !';

  nagl4:t6='!-----|-----!';
  ttp : t1='AI';           {tablica typow pol}
  tdp : t2=(17,3);         {tablica dlugosci pol}
  tk  : t3=(' Oczekiwana cyfra ',      {tablica komunikatow}
           ' Bledny znak ',
           ' Za duza liczba ',
           ' Bledna postac liczby ');

var
  ppole : integer;        {nr pierwszego pola na ekranie}
  ostp  : integer;        {nr ostatniego pola na ekranie}
  wekrk : integer;        {nr wiersza w oknie wyswietlania pol
                           rekordu ,w ktorym znajduje sie kursor}
  pekrk : integer;        {nr pola wyswietlonego na ekranie ,
                           w ktorym znajduje sie kursor}

  rob   :integer;
  irek  :integer;         {licznik rekordow}
  zrklbi,
```

```
zrklb :trklb;           {zmienna rekordowa}
zpk1b1,
zpk1b :tpklb;          {zmienna plikowa}
whas1,
whas2,                 {wskazniki hasel}
bl,
kpzb,                 {koniec przegladania zbioru}
kprek,                {koniec przegladania rekordu}
usun,
jestzb,
jestakt,
konwd,
konpr :boolean;
zn1,
zn2,
zn :char;
wyb :integer;
milrek :integer;
xk,yk :integer;
bhas : array[1..dlh] of char; {bufor na haslo}
buf : t4;             {bufor dla znakow z klawiatury}
lpol : integer;      {licznik pol}
pozsk : integer;     {wskaznik pozycji kursora w "buf"}
wczytzn,
wczytpol : boolean;
nrk : integer;       {numer komunikatu}
lzam : longint;     {dla procedury VAL}
kbl : integer;
lcyfr: integer;     {licznik cyfr w "buf"}
pcb : integer;      {pozycja pierwszej cyfry w "buf"}
```

PROCEDURE CzytZn;

```
Begin
  zn:=readkey;
  if zn=#0 then
    zn1:=readkey;
  if zn<>#13 then
    Begin
      write(zn);
      gotoxy(wherex-1,wherey);
    End;
End;
```

PROCEDURE ZnPoziom(var nu:text; m:integer);

```
var
  j:integer;
Begin
  for j:=1 to m do
    write(nu,'-');
End;
```

PROCEDURE ZnPoz;

```
var
  j:integer;
```

```
Begin
  for j:=1 to 77 do
    write(#196);
  End;

PROCEDURE SzabEkr;
  { Rysuje obrys okien }

var
  k,i:integer;
const
  tabl : array[1..5] of integer=(1,3,20,22,25);

PROCEDURE Pion;

var
  i:integer;
Begin
  for i:=1 to 23 do
    Begin
      write(#179);
      gotoxy(wherex-1,wherey+1);
    End;
  End;

Begin
  clrscr;
  write(#218,#191:78);
  gotoxy(1,25);
  write(#192,#217:78);
  gotoxy(1,2);
  Pion;
  gotoxy(79,2);
  Pion;

  for k:=2 to 4 do
    Begin
      gotoxy(1,tabl[k]);
      write(#195,#180:78);
    End;

  for k:=1 to 5 do
    Begin
      gotoxy(2,tabl[k]);
      ZnPoz;
    End;
  End;

PROCEDURE PiszNazPol;
  { Pisz nazwy pol }

Begin
  case lpol of
    1:write(tnazp1[1]);
    2:write(tnazp1[2]);
  end;
End;

PROCEDURE WczytWartPol;
  { Wczytuje wartosci pol }
```

```
Begin
  with zrk1b do
    case lpol of
      1:move(buf,naz1sw,tdp[lpol]);
      2:k1srw:=lzam;
    end;
End;
```

PROCEDURE StrzLewo;

```
Begin
  if pozk=1 then
    write(chr(7))
  else
    Begin
      pozk:=poz-1;
      gotoxy((wherex-1),wherey);
    End;
End;
```

PROCEDURE StrzPrawo;

```
Begin
  if pozk=tdp[lpol] then
    write(chr(7))
  else
    Begin
      pozk:=poz+1;
      gotoxy((wherex+1),wherey);
    End;
End;
```

PROCEDURE KasZn;

{ Kasuje znak na ktorym stoi kursor }

```
Begin
  if (tdp[lpol]-pozk)>0 then
    move(buf[poz+1],buf[pozk],(tdp[lpol]-pozk));
    buf[tdp[lpol]]:= ' ';
    write(copy(buf,pozk,(tdp[lpol]-poz+1)));
    gotoxy((wherex-(tdp[lpol]-poz+1)),wherey);
End;
```

PROCEDURE WypKom(a:integer);

{ Wypisuje w wierszu 21 komunikat nr 'a',  
wysyla sygnal dzwiekowy,ustawia kursor  
tam gdzie byl przed wywołaniem procedury }

var

x,y:integer;

```
Begin
  x:=wherex;
  y:=wherey;
  window(2,21,78,21);
  write(tk[a]);
  gotoxy(wherex-1,wherey);
  write(chr(7));
  delay(1000);
```

```
textbackground(black);
textcolor(white);
lowvideo;

gotoxy(1,1);
clreol;
window(2,pwp,78,owp);
gotoxy(x,y);

textbackground(white);
textcolor(black);
lowvideo;
End;

PROCEDURE SprPol;
( Sprawdza czy wczytany znak jest odpowiedni dla danego typu pola )
Begin
  bl:=false;
  case ttp[lpol] of
    'I',
    'L',
    'C',
    'W',
    'B': if (not(zn in ['0'..'9'])) then
      Begin
        bl:=true;
        nrk:=1;
      End;
    'A': if not((zn>=' ') and (zn<='~')) then
      Begin
        bl:=true;
        nrk:=2;
      End;
  end; {case}

  if bl then
    WypKom(nrk);
End;

PROCEDURE WypKom1;
Begin
  if ttp[lpol] in ['W','B','L','C','I'] then
    nrk:=1;
  else
    nrk:=2;
  WypKom(nrk);
End;

PROCEDURE AktRez;
Begin
  if wyb=3 then
    Begin
      jestakt:=false;
      wczytzn:=true;
    End
End;
```

```
else
  WypKom1;
End;

PROCEDURE WczytZnPol;
( Wczytuje znaki do pol )
Begin
  pozk:=1;
  wczytzn:=false;
  while not wczytzn do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytzn:=true;
        #0 : Begin
              wcy:zn:=readkey;
            End;
        if not case zn of
          #19 : AktRez;           ( Alt-R )
          #75 : StrzLewo;
          #77 : StrzPrawo;
          #83 : KasZn;           ( Del )
        else
          WypKom1;
        End; {case zn of przy 0}
      End (#0)
    else {case}
      Begin
        SprPol;
        if not bl then
          Begin
            write(zn);
            buf[pozk]:=zn;
            pozk:=pozk+1;
          End;
        if pozk>tdp[lipol] then
          Begin
            gotoxy((wherex-1),wherex);
            pozk:=pozk-1;
          End;
        End; {else od case}
      End; {case}
    End; {while}
  End;
End;
```

```
PROCEDURE SprWartPol;
( Sprawdza czy wczytana wartosc jest odpowiednia
  dla danego typu pola )
var
```

```
i:integer;
wcyfr:boolean;
begin
lcyfr:=0;
pcb:=1;
wcyfr:=false;
case ttp[lpoll] of
'I',
'L',
'W',
'B' :Begin
for i:=1 to tdp[lpoll] do
Begin
if buf[i] in ['0'..'9'] then
Begin
lcyfr:=lcyfr+1;
wcyfr:=true;
End;
if not wcyfr then
pcb:=pcb+1;
End;
if lcyfr=0 then
Begin
kbl:=0;
lzam:=0;
End
else
val(copy(buf,pcb,lcyfr),lzam,kbl);
if (kbl=0) then
Begin
if (lzam>65535) and (ttp[lpoll]='W') then
Begin
nrk:=3;
WypKom(nrk);
wczytpol:=false;
End
else
wczytpol:=true;
if (lzam>255) and (ttp[lpoll]='B') then
Begin
nrk:=3;
WypKom(nrk);
wczytpol:=false;
End
else
if ttp[lpoll]='B' then
wczytpol:=true;
End
else
Begin
nrk:=4;
WypKom(nrk);
End;
```

```
End;
  'C',
  'A' : zaczytpol:=true;
End; { case }
end;

PROCEDURE WczytajPole;

begin
  zaczytpol:=false;
  repeat
    FillChar(buf,tdp[lpol],' ');
    gotoxy (kwp,wherey);
    textbackground(white);
    textcolor(black);
    write(copy(buf,1,tdp[lpol]));
    gotoxy(kwp,wherey);
    WczytZnPoi;
    if (wyb=3) and (jestakt=false) then
      zaczytpol:=true
    else
      SprWartPoi;
  until zaczytpol;

  textbackground(black);
  textcolor(white);
  lowvideo;
end;

PROCEDURE BadDosZb;
{ Bada dostep do zbioru T_LSRW.LOT }
begin
  window(1,1,80,25);
  clrscr;
  assign(zpk1b,'T_LSRW.LOT');
  {$I-}
  reset(zpk1b);
  {$I+}
  if IOResult<>0 then
    Begin
      gotoxy(20,5);
      write('Zbior T_LSRW.LOT niedostepny');
      gotoxy(20,wherey+2);
      write('Nacisnij klawisz Enter');
      readln;
      jestzb:=false;
    end
  else
    Begin
      if (FileSize(zpk1b)=0) and (wyb<>1) then
        Begin
          gotoxy(10,1);
          write('Zbior T_LSRW.LOT jest pusty');
          delay(1500);
          jestzb:=false;
          close(zpk1b);
        end
      End
    else

```

```
        jestzb:=true;
    End;
End;

PROCEDURE WproRekDanych;
    Sluzy do wprowadzania danych )
Begin
    BadDosZb;
    if not jestzb then
        Begin
            gotoxy(20,wherey+1);
            write('Czy utworzyc nowy zbior ? : [T/N] ');
            repeat
                CzytZn;
            until upcase(zn) in ['N','T'];

            if upcase(zn)='T' then
                rewrite(zpk1b)
            else
                konpr:=true;
        End;

        if not konpr then
            Begin

            milrek:=filesize(zpk1b);
            seek(zpk1b,milrek);

            konwd:=false;
            SzabEkr;
            gotoxy(2,2);
            write('Wprowadzanie danych do zbioru  LOTNICZE SRODKI WALKI');
            gotoxy(2,23);
            write('Enter : Koniec  Del : Kasowanie');
            write('  #27 : Lewo  #26 : Prawo');
            window(2,pwp,78,owp);
            repeat
                textbackground(black);
                textcolor(white);
                lowvideo;

                clrscr;

                for lpol:=1 to ilpol do
                    Begin
                        gotoxy(knp,wherey);
                        PiszNazPol;
                        WczytajPole;
                        WczytWartPol;
                        writeln;
                    End;
                write(zpk1b,zrk1b);

                write ('Czy kontynuujesz wprowadzanie ? :[T/N] ');

                repeat
                    CzytZn;
                until upcase(zn) in ['N','T'];
                if upcase(zn)='N' then
```

```
        konwd:=true;
    until konwd=true;
    close(zpk1b);
End
else
    konpr:=false;
End;
```

```
PROCEDURE PiszWartPol;
    Wypisuje na ekranie wartosci pol rekordu }
```

```
begin
    textbackground(white);
    textcolor(black);
    lowvideo;

    with zrklb do
        case lpol of
            1:write(nazlsw:17);
            2:write(klsrw:3);
        end;
    textbackground(black);
    textcolor(white);
    lowvideo;
end;
```

```
PROCEDURE ZaznRek;
    Zaznacza rekordy do usuniecia }
```

```
begin
    if wyb=4 then
        begin
            zrklb.nazlsw[1]:=#232;
            seek(zpk1b,FilePos(zpk1b)-1);
            write(zpk1b,zrklb);
            kprek:=true;
            usun:=true;
        end;
    end;
```

```
PROCEDURE StrzBora;
```

```
begin
    pekrk:=pekrk-1;
    if pekrk<1 then
        begin
            pekrk:=1;
            write(#7);
        end
    else
        begin
            if pekrk<ppole then
                begin
                    ppole:=ppole-1;
                    ostp:=ostp-1;
                    gotoxy(1,1);
                    InsLine;
                    gotoxy(knp,1);
                    lpol:=ppole;
                    FiszNazPol;
```

```
    gotoxy(kwp,wherey);
    PiszWartPol;
    gotoxy(kwp,wherey);
  End
else
  Begin
    wekrk:=wekrk-1;
    gotoxy(wherex,wherey-1);
  End;
End;
```

PROCEDURE StrzDol;

```
Begin
  pekrk:=pekrk+1;
  if pekrk>ilpol then
    Begin
      pekrk:=ilpol;
      write(#7);
    End
  else
    Begin
      if pekrk>ostp then
        Begin
          ppole:=ppole+1;
          ostp:=ostp+1;
          gotoxy(1,1);
          DelLine;
          gotoxy(knp,ilwe);
          lpol:=ostp;
          PiszNazPol;
          gotoxy(kwp,wherey);
          PiszWartPol;
          gotoxy(kwp,wherey);
        End
      else
        Begin
          wekrk:=wekrk+1;
          gotoxy(wherex,wherey+1);
        End;
      End;
    End;
  End;
```

PROCEDURE AktPol;

```
Begin
  if wyb=3 then
    Begin
      lpol:=pekrk;
      jestakt:=true;
      WczytajPole;
      if jestakt then
        Begin
          WczytWartPol;
          seek(zpk1b,FilePos(zpk1b)-1);
          write(zpk1b,zrk1b);
        End
      else
        Begin
```

```
    gotoxy(kwp,wherey);
    PiszWartPol;
End;
gotoxy(kwp,wherey);
End;
end;

PROCEDURE PrzeglRek;
begin
    if ilpol>ilwe then
        rob:=ilwe
    else
        rob:=ilpol;
    kpzb:=false;

    window(2,pwp,78,owp);

    while (not Eof(zpk1b)) and (not kpzb) do
        Begin
            clrscr;
            ppole:=1;
            ostp:=ilwe;
            wekrk:=1;
            pekrk:=1;

            read(zpk1b,zrk1b);
            for lpol:=1 to rob do
                Begin
                    gotoxy(knp,1+lpol-1);
                    PiszNazPol;
                    gotoxy(kwp,wherey);
                    PiszWartPol;
                    gotoxy(kwp,1);
                End;
            kprek:=false;
            while not kprek do
                Begin
                    zn:=readkey;
                    case zn of
                        #27 : Begin { Esc }
                            kprek:=true;
                            kpzb:=true;
                        End;
                        #13 : AktPol; { Enter }
                        #0  : Begin
                            zn:=readkey;
                            case zn of
                                #22 : ZaznRek; { Alt-U }
                                #72 : StrzGora;
                                #80 : StrzDol;
                                #81 : kprek:=true; { PgDn }
                            end;
                        End; { #0 }
                    end;
                End; { while not kprek }
            End;
        close(zpk1b);
```

end;

PROCEDURE PrzeglZbioru;

Sluzy do przegladania zbioru LOTNICZE SRODKI WALKI bez mozliwosci zmian w danych )

begin

BadDosZb;

if jestzb then

Begin

SzabEkr;

gotoxy(2,2);

write('Przegladanie zbioru LOTNICZE SRODKI WALKI ');

gotoxy(2,23);

write('Esc : Wyjscie PgDn : Nastepny rekord');

write(' #24 : Gora #25 : Dol');

PrzeglRek;

End; { jestzb }

end;

PROCEDURE UsuwRekDanych;

Usuwa ze zbioru zaznaczone rekordy )

begin

BadDosZb;

if jestzb then

Begin

SzabEkr;

gotoxy(2,2);

write('Usuwanie rekordow ze zbioru LOTNICZE SRODKI WALKI');

gotoxy(2,23);

write('Esc : Wyjscie PgDn : Nastepny rekord Alt-U : Usuniecie');

gotoxy(2,24);

write('#24 : Gora #25 : Dol');

usun:=false;

PrzeglRek;

if usun then

Begin

assign(zpk1b, 'T\_LSRW.LOT');

reset(zpk1b);

assign(zpk1b1, 'T\_LSRW01.LOT');

rewrite(zpk1b1);

while not Eof(zpk1b) do

Begin

read(zpk1b, zrk1b);

if zrk1b.naz1sw[11]<>#232 then

write(zpk1b1, zrk1b);

End;

close(zpk1b);

close(zpk1b1);

erase(zpk1b);

rename(zpk1b1, 'T\_LSRW.LOT');

End;

End;

end;

PROCEDURE AktRekDanych;

begin

```
BadDosZb;
if jestzb then
  Begin
    SzabEkr;
    gotoxy(2,2);
    write('Aktualizacja zbioru LOTNICZE SRODKI WALKI');
    gotoxy(2,23);
    write('Esc : Wyjscie PgDn : Nastepny rekord Del : Kasowanie ');
    write('Alt-R : Rezygnacja');
    gotoxy(2,24);
    write('Enter : Poczatek,Koniec '#27' : Lewo ');
    write('#26' : Prawo '#24' : Gora '#25' : Dol');
    PrzeglRek;
  End;
end;

PROCEDURE WyprDanych(var nurz:text; turz:char);
begin
  writeln(nurz,nagl1);
  writeln(nurz);
  write(nurz,#32:18);
  ZnPoziom(nurz,29);
  writeln(nurz);
  write(nurz,#32:17);
  writeln(nurz,nagl2);
  write(nurz,#32:17);
  writeln(nurz,nagl4);
  write(nurz,#32:17);
  writeln(nurz,nagl3);
  write(nurz,#32:17);
  writeln(nurz,nagl4);

  if upcase(turz)='M' then
    window(1,8,80,25);
    kpzb:=false;
    lrek:=0;

    while (not eof(zpk1b)) and (not kpzb) do
      Begin
        read(zpk1b,zrk1b);
        with zrk1b do
          Begin
            write(nurz,#32:17);
            writeln(nurz,' ',naz1sw:17,' ',k1srw:3,' ');
          End;
        lrek:=lrek+1;
        if (lrek>14) and (upcase(turz)='M') and (FileSize(zpk1b)<>lrek) then
          Begin
            writeln;
            write(#32:17);
            write('K-Kontynuacja R-Rezygnacja');
            repeat
              CzytZn;
            until upcase(zn) in ['K','R'];
            if upcase(zn)='K' then
              Begin
                clrscr;
                lrek:=0;
              End
          End
      End
    End
  End
end;
```

```
        else
            kpzb:=true;
        End;
    End;

f (upcase(turz)='M') and (upcase(zn)<>'R') then
    Begin
        write(nurz,#32:18);
        ZnPoziom(nurz,29);
        writeln;
        writeln;
        write(#32:17);
        write('Koniec zbioru  Nacisnij klawisz Enter');
        repeat
            zn:=readkey;
        until zn=#13;
    End;
f upcase(turz)='D' then
    Begin
        write(nurz,#32:18);
        ZnPoziom(nurz,29);
        writeln(nurz);
    End;
window(1,1,80,25);
End;

PROCEDURE WyprZbioru;

Begin
    BadDosZb;
    if jestzb then
        Begin
            clrscr;
            gotoxy(10,3);
            write('Wyprowadzanie danych ze zbioru LOTNICZE SRODKI WALKI');
            gotoxy(10,5);
            write('Podaj urzadzenie wyprowadzania : [D/M] ');
            repeat
                CzytZn;
            until upcase(zn) in['D','M'];
            zn2:=zn;
            if upcase(zn)='M' then
                Begin
                    clrscr;
                    WyprDanych(Output,zn2);
                End
            else
                Begin
                    gotoxy(10,7);
                    write('Wlacz drukarke i nacisnij Enter');
                    repeat
                        zn:=readkey;
                    until zn=#13;
                    WyprDanych(Lst,zn2);
                End;
            close(zpklb);
        End;
    End;
End;
```

```
PROCEDURE Sort1;
```

```
var  
  i,j,ilrek:integer;  
begin  
  ilrek:=FileSize(zpklb)-1;  
  for i:=1 to ilrek do  
    for j:=ilrek downto i do  
      Begin  
        seek(zpklb,j);  
        read(zpklb,zrklb);  
        seek(zpklb,j-1);  
        read(zpklb,zrklb1);  
        if zrklb1.klsrw>zrklb.klsrw then  
          Begin  
            seek(zpklb,j-1);  
            write(zpklb,zrklb);  
            write(zpklb,zrklb1);  
          End;  
        End;  
      End;  
end;
```

```
PROCEDURE Sortowanie;
```

```
begin  
  BadDosZb;  
  if jestzb then  
    Begin  
      clrscr;  
      writeln('Sortowanie rekordow w zbiorze T_LSRW.LDT odbywa sie');  
      writeln('wedlug kodu lotniczego srodka walki');  
      writeln;  
      write(' ':15,'Nacisnij klawisz Enter');  
      repeat  
        zn:=readkey;  
      until zn=#13;  
      Sort1;  
      close(zpklb);  
    End;  
end;
```

```
PROCEDURE WczytHaslo;
```

```
var  
  whaslo:boolean;  
begin  
  whaslo:=false;  
  pozk:=1;textbackground(white);  
  textcolor(black);  
  lowvideo;  
  FillChar(bhas,dlh,' ');  
  write(bhas);  
  gotoxy(wherex-dlh,wherex);  
  while not whaslo do  
    Begin  
      zn:=readkey;  
      case zn of  
        ' ':  
          Val(copy(bhas,1,pozk),rob,rob1);  
        '0'..'9':  
          Val(copy(bhas,1,pozk),rob,rob1);  
        'A'..'Z':  
          Val(copy(bhas,1,pozk),rob,rob1);  
      end;  
      write(rob);  
      if rob=rob1 then  
        whaslo:=true;  
      end;  
      pozk:=pozk+1;  
    End;  
end;
```

```
#13 : whaslo:=true;
#0  : zn:=readkey;
else
  Begin
    if (zn>=' ') and (zn<='~') then
      Begin
        write(zn);
        bhas[pozki]:=zn;
        pozki:=pozki+1;
      End;
    End;
  end;
if pozki>dlh then
  Begin
    gotoxy(wherex-1,wherey);
    pozki:=pozki-1;
  End;
End;
textbackground(black);
textcolor(white);
lowvideo;
end;

PROCEDURE Haslo;

const
  hasl:string[dlh]='SZYM';
var
  rob,
  robi,
  hour,
  min,
  sec,
  s100 :word;
  kbl  :integer;

begin
  whas1:=false;
  whas2:=false;
  clrscr;
  gotoxy(10,3);
  write('Podaj haslo i naciśnij klawisz Enter : ');
  WczytHaslo;
  if has1=bhas then
    whas1:=true
  else
    Begin
      GetTime(hour,min,sec,s100);
      Val(copy(bhas,1,1),rob,kbl);
      if kbl=0 then
        Begin
          robi:=min div 10;
          if robi=rob then
            Begin
              Val(copy(bhas,2,1),rob,kbl);
              if kbl=0 then
                Begin
                  if rob=0 then
                    Begin
                      Val(copy(bhas,3,2),rob,kbl);
```

```
        if kbl=0 then
          Begin
            if hour=rob then
              whas2:=true;
            End;
          End;
        End;
      End;
    End;
  End;
```

```
if (not whas1) and (not whas2) then
  Begin
    gotoxy(10,5);
    write('Bledne haslo  Nacisnij klawisz Enter');
    readln;
    konpr:=true;
  End;
```

nd;

PROCEDURE DostepPelny;

egin

```
  clrscr;
  gotoxy(20,5);
  write('LOTNICZE SRODKI WALKI');
  gotoxy(22,7);
  write('1.Wprowadzanie');
  gotoxy(22,8);
  write('2.Przeglądanie');
  gotoxy(22,9);
  write('3.Aktualizacja');
  gotoxy(22,10);
  write('4.Usuwanie');
  gotoxy(22,11);
  write('5.Wyprowadzanie');
  gotoxy(22,12);
  write('6.Sortowanie');
  gotoxy(22,13);
  write('7.Wyjście');
  gotoxy(22,15);
  write('Wybierz numer: ');
```

```
  repeat
    CzytZn;
  until zn in ['1'..'7'];
```

```
  wyb:=ord(zn)-48;
```

```
  case wyb of
    1:WproRekDanych;
    2:PrzeGlZbioru;
    3:AktRekDanych;
    4:UsuwRekDanych;
    5:WyprZbioru;
    6:Sortowanie;
    7:konpr:=true;
```

```
  End; {case}
```

nd;

PROCEDURE DostepOgr;

```
begin
  clrscr;
  gotoxy(20,5);
  write('LOTNICZE SRODKI WALKI');
  gotoxy(22,7);
  write('1.Przeglądanie');
  gotoxy(22,8);
  write('2.Wyprowadzanie');
  gotoxy(22,9);
  write('3.Sortowanie');
  gotoxy(22,10);
  write('4.Wyjscie');
  gotoxy(22,12);
  write('Wybierz numer: ');
```

```
repeat
  CzytZn;
until zn in ['1'..'4'];

wyb:=ord(zn)-48;

case wyb of
  1:PrzeGlZbioru;
  2:WyprZbioru;
  3:Sortowanie;
  4:konpr:=true;
```

End; {case}

```
end;

P O C Z A T E K      P R O G R A M U }
```

```
EGIN
  TextMode(2);
  textbackground(black);
  textcolor(white);
  lowvideo;
  konpr:=false;
  Haslo;
  while not konpr do
    Begin
      window(1,1,80,25);
      textbackground(black);
      textcolor(white);
      lowvideo;

      if whasi then
        DostepOgr
      else
        DostepPelny;

    End; {while}
```

END.

PROGRAM OBUD;

uses crt,dos,printer;

```
const
  ilpol=5;           {ilosc pol w rekordzie}
  dlnp=17;          {dlugosc nazwy pola}
  mdp=58;           {max dlugosc pola w znakach}
  knp=1;            {nr kolumny-początek nazwy pola}
  kwp=19;           {nr kolumny-początek wartosci pola}
  pwp=4;            {nr wiersza ekranu od ktorego
                    rozpoczyna sie wyswietlanie pol}
  owp=19;           {nr wiersza ekranu na ktorym
                    konczy sie wyswietlanie pol}
  ilwe=owp-pwp+1;   {ilosc wierszy ekranu w ktorych
                    wyswietlane sa pola}
  dlh=4;            {dlugosc hasla}
```

```
type
  nazp=string[dlnp]; {nazwa pola}
  tnazp=array[1..ilpol] of nazp; {typ tablicy z nazwami pol}
  trklb=record        {typ rekordu z pliku OBIEKTY UDERZENIA}
    kobu :byte;
    naob :array[1..58] of char;
    dlob :integer;
    szob :integer;
    sorob :word;
  end;
```

```
tpklb=file of trklb; {typ pliku}
t1=string[ilpol];
t2=array[1..ilpol] of byte;
t3=array[1..4] of string[22];
t4=array[1..mdp] of char;
t5=string[36];
t6=string[72];
t15=string[46];
t16=string[64];
```

```
const
  tnazp1:tnazp=('Kod obiektu',
               'Nazwa obiektu',
               'Dlugosc obiektu',
               'Szerokosc obiektu',
               'Sred.odl.rozpozn.');
```

```
nagl1:t5='          Obiekty uderzenia LMB';
nagl2:t6='|KOD KLA-|          |DLUGOSC|SZERO-|'+
         '|SREDNIA ODLE-|';
nagl3:t6='|SYFIKA-|  NAZWA OBIEKTU UDERZENIA  |OBIEKTU|KOSC  |+
         '|GLOSC ROZPOZ-|';
nagl4:t6='|CYJNY  |          | [m] |OBIEKTU|'+
         '|NANIA OBIEKTU|';
nagl5:t6='|OBIEKTU |          |          | [m] |+
         '|          | [m] |';
nagl6:t6='| 1 | 2 | 3 | 4 |+
         '|... 5 of char|';
nagl7:t6='|-----|-----|-----|-----|'+
         '|-----|';
```

```
nagl11:t15='                               Obiekty uderzenia LMB o kodzie ' ;
nagl12:t16='|                               |DLUGOSC|SZERO-|'+
           'SREDNIA ODLE-|';
nagl13:t16='|   NAZWA OBIEKTU UDERZENIA   |OBIEKTU|KOSC   |'+
           'GLOSC ROZPOZ-|';
nagl14:t16='|                               | [m] |OBIEKTU|'+
           'NANIA OBIEKTU|';
nagl15:t16='|                               | [m] |'+
           ' [m] |';
nagl16:t16='|                               | 2 | 3 |'+
           ' 4 |';
nagl17:t16='|-----|-----|'+
           '-----|';
```

```
ttp : t1='BAIIW';                               {tablica typow pol}
tdp : t2=(2,58,4,3,5);                           {tablica dlugosci pol}
tk   : t3=(' Oczekiwana cyfra ',
           ' Bledny znak ',
           ' Za duza liczba ',
           ' Bledna postac liczby ');           {tablica komunikatow}
```

```
wart1 : byte;
ppole  : integer;                                {nr pierwszego pola na ekranie}
ostp   : integer;                                {nr ostatniego pola na ekranie}
wekrk  : integer;                                {nr wiersza w oknie wyswietlania pol
rekordu ,w ktorym znajduje sie kursor}
pekrk  : integer;                                {nr pola wyswietlonego na ekranie ,
w ktorym znajduje sie kursor}
rob    : integer;
lrek   : integer;                                {licznik rekordow}
zrklb1,
zrklb  :trklb;                                   {zmienna rekordowa}
zpk1b1,
zpk1b  :tpklb;                                   {zmienna plikowa}
whas1,
whas2,                                       {wskazniki hasel}
bl,
kpsz,                                       {koniec przegladania zbioru}
kprek,                                       {koniec przegladania rekordu}
usun,
jestzb,
jestakt,
konwd,
konpr  :boolean;
zn1,
zn2,
zn3,
zn     :char;
wyb    :integer;
milrek :integer;
xk,yk  :integer;
bk11   :string[2];
bufkl  :array[1..2] of char;
bhas   :array[1..dlh] of char;                {bufor na haslo}
buf    : t4;                                  {bufor dla znakow z klawiatury}
lpol   : integer;                                {licznik pol}
```

```
pozkl : integer;           {wskaznik pozycji kursora w "buf"}
wczytkl,
wczytzn,
wczytpol : boolean;
nrk : integer;             {numer komunikatu}
lzam : longint;           {dla procedury VAL}
kbl : integer;
lcyfr : integer;          {licznik cyfr w "buf"}
pcb : integer;            {pozycja pierwszej cyfry w "buf"}

PROCEDURE Wczytklucz(dkl:integer);
r
  ww:byte;
gin
  ww:=0;
  wczytkl:=false;
  pozkl:=1;textbackground(white);
  textcolor(black);
  lowvideo;
  FillChar(bufkl,dkl,' ');
  write(copy(bufkl,1,dkl));
  gotoxy(wherex-dkl,wherey);
  while not wczytkl do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytkl:=true;
        #0 : zn:=readkey;
        else
          Begin
            if (zn>=' ') and (zn<='~') then
              Begin
                write(zn);
                bufkl[pozkl]:=zn;
                pozkl:=pozkl+1;
              End;
            End;
          end;
        if pozkl>dkl then
          Begin
            ww:=1;
            gotoxy(wherex-1,wherey);
            pozkl:=pozkl-1;
          End;
        End;
      textbackground(black);
      textcolor(white);
      lowvideo;
      if ww=0 then
        pozkl:=pozkl-1;
      end;
    nd;
  PROCEDURE CzytZn;
  gin
    zn:=readkey;
    if zn=#0 then
      zn1:=readkey;
```

```
if zn<>#13 then
  Begin
    write(zn);
    gotoxy(wherex-1,wherey);
  End;
nd;

PROCEDURE ZnPoziom(var nu:text; m:integer);
var
  j:integer;
begin
  for j:=1 to m do
    write(nu,'-');
  nd;

PROCEDURE ZnPoz;
var
  j:integer;
begin
  for j:=1 to 77 do
    write(#196);
  nd;

PROCEDURE SzabEkr;
  Rysuje obrys okien }
var
  k,i:integer;
const
  tabl : array[1..5] of integer=(1,3,20,22,25);
PROCEDURE Pion;
var
  i:integer;
begin
  for i:=1 to 23 do
    Begin
      write(#179);
      gotoxy(wherex-1,wherey+1);
    End;
  nd;

begin
  clrscr;
  write(#218);
  gotoxy(79,wherey);
  write(#191);
  gotoxy(1,25);
  write(#192);
  gotoxy(79,wherey);
  write(#217);
  gotoxy(1,2);
  Pion;
  gotoxy(79,2);
```

```
Pion;  
for k:=1 to 5 do  
  Begin  
    gotoxy(2,tab1[k]);  
    ZnPoz;  
  End;  
for k:=2 to 4 do  
  Begin  
    gotoxy(1,tab1[k]);  
    write(#195);  
    gotoxy(79,tab1[k]);  
    write(#180);  
  End;  
end;  
PROCEDURE PiszNazPol;  
  Pisz nazwy pol )  
begin  
  case lpol of  
    1..ilpol : write(tnazp1[lpol]);  
  end;  
end;  
PROCEDURE WczytWartPol;  
  Wczytuje wartosci pol )  
begin  
  with zrk1b do  
    case lpol of  
      1:kobu:=lzam;  
      2:move(buf,naob,tdp[lpol]);  
      3:dlob:=lzam;  
      4:szob:=lzam;  
      5:sorob:=lzam;  
    end;  
end;  
PROCEDURE StrzLewo;  
begin  
  if pozk=1 then  
    write(chr(7))  
  else  
    Begin  
      pozk:=poz-1;  
      gotoxy((wherex-1),wherey);  
    End;  
end;  
PROCEDURE StrzPrawo;  
begin  
  if pozk=tdp[lpol] then  
    write(chr(7))  
  else
```

```
Begin
  pozk:=pozk+1;
  gotoxy((wherex+1),wherey);
End;
d;

PROCEDURE KasZn;
  Kasuje znak na ktorym stoi kursor }
begin
  if (tdp[lpoll]-pozk)>0 then
    move(buf[pozk+1],buf[pozk],(tdp[lpoll]-pozk));
  buf[tdp[lpoll]]:= ' ';
  write(copy(buf,pozk,(tdp[lpoll]-pozk+1)));
  gotoxy((wherex-(tdp[lpoll]-pozk+1)),wherey);
d;

PROCEDURE WypKom(a:integer);
  Wypisuje w wierszu 21 komunikat nr 'a',
  wysyla sygnal dzwiekowy,ustawia kursor
  tam gdzie byl przed wywołaniem procedury }
var
  x,y:integer;
begin
  x:=wherex;
  y:=wherey;
  window(2,21,78,21);
  write(tk[a]);
  gotoxy(wherex-1,wherey);
  write(chr(7));
  delay(1000);

  textbackground(black);
  textcolor(white);
  lowvideo;

  gotoxy(1,1);
  clreol;
  window(2,pwp,78,owp);
  gotoxy(x,y);

  textbackground(white);
  textcolor(black);
  lowvideo;
end;

PROCEDURE SprPol;
  Sprawdza czy wczytany znak jest odpowiedni dla danego typu pola }
begin
  bl:=false;
  case ttp[lpoll] of
    'I',
    'W',
    'B': if (not(zn in ['0'..'9'])) then
      Begin
        bl:=true;wypKom1;

```

```
        nrk:=1;
    End;
'A': if not((zn>=' ') and (zn<='~')) then
    Begin
        bl:=true;
        nrk:=2;
    End;
end; {case}

if bl then
WypKom(nrk);
end;

PROCEDURE WypKom1;
begin
if ttp[lpol] in ['W','B','I'] then
    nrk:=1
else
    nrk:=2;
WypKom(nrk);
end;

PROCEDURE AktRez;
begin
if wyb=3 then
    Begin
        jestakt:=false;
        wczytzn:=true;
    End
else
    WypKom1;
end;

PROCEDURE WczytZnPol;
Wczytuje znaki do pol )
begin
    pozk:=1;
    wczytzn:=false;
    while not wczytzn do
        Begin
            zn:=readkey;
            case zn of

                #13 : wczytzn:=true;

                #0 : Begin
                    zn:=readkey;

                    case zn of

                        #19 : AktRez;           { Alt-R }
                        #75 : StrzLewo;
                        #77 : StrzPrawo;
                        #83 : KasZn;           { Del }

                    end;

                else
                    WypKom1;
            end;
        end;
    end;
end;
```

```
End; {case zn of przy 0}
End {#0}
else {case}
Begin
SprPol;
if not bl then
Begin
write(zn);
buf[pozki]:=zn;
pozki:=pozki+1;
End;
if pozki>tdp[lpol] then
Begin
gotoxy((wherex-1),wherey);
pozki:=pozki-1;
End;
End; {else od case}
End; {case}
End; {while}
d;
```

PROCEDURE SprWartPol;  
Sprawdza czy wczytana wartosc jest odpowiednia  
dla danego typu pola ?

```
var
i:integer;
wcyfr:boolean;
begin
lcyfr:=0;
pcb:=1;
wcyfr:=false;
case ttp[lpol] of
'I',
'W',
'B' :Begin
for i:=1 to tdp[lpol] do
Begin
if buf[i] in ['0'..'9'] then
Begin
lcyfr:=lcyfr+1;
wcyfr:=true;
End;
if not wcyfr then
pcb:=pcb+1;
End;
if lcyfr=0 then
Begin
kbl:=0;
lzam:=0;
End
else
```

```
val(copy(buf,pcb,lcyfr),lzam,kbl);
if (kbl=0) then
  Begin
    if (lzam>65535) and (ttp[lpol]='W') then
      Begin
        nrk:=3;
        WypKom(nrk);
        zaczytpol:=false;
      End
    else
      zaczytpol:=true;
    if (lzam>255) and (ttp[lpol]='B') then
      Begin
        nrk:=3;
        WypKom(nrk);
        zaczytpol:=false;
      End
    else
      if ttp[lpol]='B' then
        zaczytpol:=true;
      End
    else
      Begin
        nrk:=4;
        WypKom(nrk);
      End;
  End;
  'C',
  'A' : zaczytpol:=true;
End; { case }
```

```
PROCEDURE WczytajPole;
```

```
begin
  zaczytpol:=false;
  repeat
    FillChar(buf,tdp[lpol],' ');
    gotoxy (kwp,wherey);
    textbackground(white);
    textcolor(black);
    write(copy(buf,1,tdp[lpol]));
    gotoxy(kwp,wherey);
    WczytZnPol;
    if (wyb=3) and (jestakt=false) then
      zaczytpol:=true
    else
      SprWartPol;
  until zaczytpol;

  textbackground(black);
  textcolor(white);
  lowvideo;
end;
```

```
PROCEDURE BadDosZb;  
{ Bada dostep do zbioru OBIEKTY.LOT }  
{ Sprawdza dostep do zbioru OBIEKTY.UWAZENIA }  
begin  
  window(1,1,80,25);  
  clrscr;  
  assign(zpk1b,'OBIEKTY.LOT');  
  ($I-)  
  reset(zpk1b);  
  ($I+)  
  if IOResult<>0 then  
    Begin  
      gotoxy(20,5);  
      write('Zbior OBIEKTY.LOT niedostepny');  
      gotoxy(20,wherey+2);  
      write('Nacisnij klawisz Enter');  
      readln;  
      jestzb:=false;  
    end  
  else  
    Begin  
      if (FileSize(zpk1b)=0) and (wyb<>1) then  
        Begin  
          gotoxy(10,1);  
          write('Zbior OBIEKTY.LOT jest pusty');  
          delay(1500);  
          jestzb:=false;  
          close(zpk1b);  
        end  
      else  
        jestzb:=true;  
    end;  
end;  
end;
```

```
PROCEDURE WproRekDanych;  
{ Sluzy do wprowadzania danych }
```

```
begin  
  BadDosZb;  
  if not jestzb then  
    Begin  
      gotoxy(20,wherey+1);  
      write('Czy utworzyc nowy zbior ? : [T/N] ');  
      repeat  
        CzytZn;  
        until upcase(zn) in ['N','T'];  
  
      if upcase(zn)='T' then  
        rewrite(zpk1b)  
      else  
        konpr:=true;  
    end;  
  
    if not konpr then  
      Begin  
  
        milrek:=filesize(zpk1b);  
        seek(zpk1b,milrek);  
  
        konwd:=false;
```

```
SzabEkr;
gotoxy(2,2);
write('Wprowadzanie danych do zbioru  OBIEKTY UDERZENIA');
gotoxy(2,23);
write('Enter : Koniec  Del : Kasowanie');
write('  #27 : Lewo  #26 : Prawo');
window(2,pwp,78,owp);
  repeat
    textbackground(black);
    textcolor(white);
    lowvideo;

    clrscr;

    for lpol:=1 to ilpol do
      Begin
        gotoxy(knp,wherey);
        PiszNazPol;
        WczytajPole;
        WczytWartPol;
        writeln;
      End;
    write(zpklb,zrklb);

    write ('Czy kontynuujesz wprowadzanie ? :[T/N] ');

    repeat
      CzytZn;
      until upcase(zn) in ['N','T'];
      if upcase(zn)='N' then
        konwd:=true;
      until konwd=true;
      close(zpklb);
    End
  else
    konpr:=false;
  End;

PROCEDURE PiszWartPol;
  Wypisuje na ekranie wartosci pol rekordu }
begin
  textbackground(white);
  textcolor(black);
  lowvideo;

  with zrklb do
    case lpol of
      1:write(kobu:2);
      2:write(naob:58);
      3:write(dlob:4);
      4:write(szob:3);
      5:write(sorob:5);
    end;
  textbackground(black);
  textcolor(white);
  lowvideo;
end;

PROCEDURE ZaznRek;
```

Zaznacza rekordy do usuniecia }

```
begin
  if wyb=4 then
    Begin
      zrk1b.naob[1]:=#232;
      seek(zpk1b,FilePos(zpk1b)-1);
      write(zpk1b,zrk1b);
      kprek:=true;
      usun:=true;
    End;
  end;
```

PROCEDURE StrzGora;

```
begin
  pekrk:=pekrk-1;
  if pekrk<1 then
    Begin
      pekrk:=1;
      write(#7);
    End
  else
    Begin
      if pekrk<ppole then
        Begin
          ppole:=ppole-1;
          ostp:=ostp-1;
          gotoxy(1,1);
          InSLine;
          gotoxy(knp,1);
          lpol:=ppole;
          PiszNazPol;
          gotoxy(kwp,wherey);
          PiszWartPol;
          gotoxy(kwp,wherey);
        End
      else
        Begin
          wekrk:=wekrk-1;
          gotoxy(wherex,wherey-1);
        End;
      End;
    End;
```

PROCEDURE StrzDol;

```
begin
  pekrk:=pekrk+1;
  if pekrk>ilpol then
    Begin
      pekrk:=ilpol;
      write(#7);
    End
  else
    Begin
      if pekrk>ostp then
        Begin
          ppole:=ppole+1;
          ostp:=ostp+1;
        End;
      End;
    End;
```

```
    gotoxy(1,1);
    DelLine;
    gotoxy(knp,ilwe);
    lpol:=ostp;
    PiszNazPol;
    gotoxy(kwp,wherey);
    PiszWartPol;
    gotoxy(kwp,wherey);
End
else
Begin
    wekrk:=wekrk+1;
    gotoxy(wherex,wherey+1);
End;
End;
end;
```

PROCEDURE AktPol;

```
begin
    if wyb=3 then
        Begin
            lpol:=pekrk;
            jestakt:=true;
            WczytajPole;
            if jestakt then
                Begin
                    WczytWartPol;
                    seek(zpk1b,FilePos(zpk1b)-1);
                    write(zpk1b,zrk1b);
                End
            else
                Begin
                    gotoxy(kwp,wherey);
                    PiszWartPol;
                End;
            gotoxy(kwp,wherey);
            End;
        end;
```

PROCEDURE PrzeglRek;

```
begin
    if ilpol>ilwe then
        rob:=ilwe
    else
        rob:=ilpol;
    kpzb:=false;
    window(2,pwp,78,owp);
    while (not Eof(zpk1b)) and (not kpzb) do
        Begin
            clrscr;
            ppole:=1;
            ostp:=ilwe;
            wekrk:=1;
            pekrk:=1;
```

```
read(zpk1b,zrklb);
for lpol:=1 to rob do
  Begin
    gotoxy(knp,1+lpol-1);
    PiszNazPol;
    gotoxy(kwp,wherey);
    PiszWartPol;
    gotoxy(kwp,1);
  End;
kprek:=false;
while not kprek do
  Begin
    zn:=readkey;
    case zn of
      #27 : Begin { Esc }
        kprek:=true;
        kpzb:=true;
      End;
      #13 : AktPol; { Enter }
      #0 : Begin
        zn:=readkey;
        case zn of
          #22 : ZaznRek; { Alt-U }
          #72 : StrzGora;
          #80 : StrzDol;
          #81 : kprek:=true; { PgDn }
        end;
      End; { #0 }
    end;
  End; { while not kprek }
End;
close(zpk1b);
nd;
```

PROCEDURE PrzeglZbioru;

Sluzy do przegladania zbioru OBIEKTY UDERZENIA bez mozliwosci zmian w danych }

```
begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Przegladanie zbioru OBIEKTY UDERZENIA ');
      gotoxy(2,23);
      write('Esc : Wyjscie PgDn : Nastepny rekord');
      write(' #24 : Gora #25 : Dol');
      PrzeglRek;
    End; { jestzb }
end;
```

PROCEDURE UsuwRekDanych;

Usuwa ze zbioru zaznaczone rekordy }

```
begin
  BadDosZb;
  if jestzb then
```

```
Begin
  SzabEkr;
  gotoxy(2,2);
  write('Usuwanie rekordow ze zbioru OBIEKTY UDERZENIA');
  gotoxy(2,23);
  write('Esc : Wyjscie PgDn : Nastepny rekord Alt-U : Usuniecie');
  gotoxy(2,24);
  write('#24' : Gora '#25' : Dol');
  usun:=false;
  PrzeglRek;
  if usun then
    Begin
      assign(zpk1b,'OBIEKTY.LOT');
      reset(zpk1b);
      assign(zpk1b1,'OBIEKTY1.LOT');
      rewrite(zpk1b1);
      while not Eof(zpk1b) do
        Begin
          read(zpk1b,zrklb);
          if zrklb.naob[11]<>#232 then
            write(zpk1b1,zrklb);
          End;
        End;
      close(zpk1b);
      close(zpk1b1);
      erase(zpk1b);
      rename(zpk1b1,'OBIEKTY.LOT');
    End;
  End;
end;
```

PROCEDURE AktRekDanych;

```
begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Aktualizacja zbioru OBIEKTY UDERZENIA');
      gotoxy(2,23);
      write('Esc : Wyjscie PgDn : Nastepny rekord Del : Kasowanie ');
      write('Alt-R : Rezygnacja');
      gotoxy(2,24);
      write('Enter : Poczatek,Koniec '#27' : Lewo ');
      write('#26' : Prawo '#24' : Gora '#25' : Dol');
      PrzeglRek;
    End;
  end;
```

PROCEDURE WyprDanych(var nurz:text; turz:char);

```
begin
  if zn3='1' then
    Begin
      writeln(nurz,nagl1);
      writeln(nurz);
      write(nurz,#32);
      ZnPoziom(nurz,70);
      writeln(nurz);
      writeln(nurz,nagl2);
    End;
  end;
```

```
writeln(nurz,nagl3);
writeln(nurz,nagl4);
writeln(nurz,nagl5);
writeln(nurz,nagl7);
writeln(nurz,nagl6);
writeln(nurz,nagl7);
  end
else
  Begin
write(nurz,nagl11);
writeln(nurz,bufkl:2);
writeln(nurz);
write(nurz,#32);
ZnPoziom(nurz,61);
writeln(nurz);
writeln(nurz,nagl12);
writeln(nurz,nagl13);
writeln(nurz,nagl14);
writeln(nurz,nagl15);
writeln(nurz,nagl17);
writeln(nurz,nagl16);
writeln(nurz,nagl17);
  End;

if upcase(turz)='M' then
  window(1,11,80,25);
  kpzb:=false;
  lrek:=0;

  while (not eof(zpk1b)) and (not kpzb) do
    Begin
      read(zpk1b,zrk1b);
      with zrk1b do
        Begin
          if zn3='1' then
            Begin
              writeln(nurz,'| ',kobu:2,' | ',copy(naob,1,29),' | ',+
                dlob:4,' | ',szob:3,' | ',sorob:5,' | ');
              writeln(nurz,'| ',copy(naob,30,29),' | ',+
                '| ');
              lrek:=lrek+1;
            End
          else
            Begin
              val(copy(bufkl,1,pozkl),lzam,kbl);
              if (kbl=0) and (lzam=kobu) then
                Begin
                  writeln(nurz,'| ',copy(naob,1,29),' | ',+
                    dlob:4,' | ',szob:3,' | ',sorob:5,' | ');
                  writeln(nurz,'| ',copy(naob,30,29),' | ',+
                    '| ');
                  lrek:=lrek+1;
                End;
            End;
          End;
        End;
      if (lrek>5) and (upcase(turz)='M') and (FileSize(zpk1b)<>lrek) then
        Begin
          writeln;
          write('K-Kontynuacja R-Rezygnacja');
```

```
repeat
  CzytZn;
until upcase(zn) in ['K', 'R'];
if upcase(zn)='K' then
  Begin
    clrscr;
    lrek:=0;
  End
else
  kpzb:=true;
End;
End;

if (upcase(turz)='M') and (upcase(zn) <> 'R') then
  Begin
    write(nurz, #32);
    if zn3='1' then
      ZnPoziom(nurz, 70)
    else
      ZnPoziom(nurz, 61);
    writeln;
    writeln;
    write('Koniec zbioru. Nacisnij klawisz Enter');
    repeat
      zn:=readkey;
    until zn=#13;
  End;
if upcase(turz)='D' then
  Begin
    write(nurz, #32);
    if zn3='1' then
      ZnPoziom(nurz, 70)
    else
      ZnPoziom(nurz, 61);
    writeln(nurz);
  End;
window(1, 1, 80, 25);
end;

PROCEDURE WyprZbioru;
begin
  BadDosZb;
  if jestzb then
    Begin
      clrscr;
      gotoxy(15, 1);
      write('Wyprowadzanie danych ze zbioru OBIEKTY UDERZENIA');
      gotoxy(10, 3);
      write('1. Wszystkie obiekty uderzenia. ');
      gotoxy(10, 4);
      write('2. Wybrane obiekty uderzenia. ');
      gotoxy(10, 6);
      write('Wybierz numer: ');
      repeat
        CzytZn;
      until zn in ['1', '2'];
      zn3:=zn;
      if zn3='2' then
        Begin
```

```
gotoxy(10,8);
write('Podaj kod obiektow uderzenia i naciśnij Enter : ');
WczytKlucz(2);
  End;
gotoxy(10,11);
write('Podaj urządzenie wyrowadzania : [D/M] ');
repeat
  CzytZn;
until upcase(zn) in ['D','M'];
zn2:=zn;
if upcase(zn)='M' then
  Begin
    clrscr;
    WyprDanych(Output,zn2);
  End
else
  Begin
    gotoxy(10,13);
    write('Włącz drukarke i naciśnij Enter');
    repeat
      zn:=readkey;
    until zn=#13;
    WyprDanych(Lst,zn2);
  End;
close(zpk1b);
End;
End;

PROCEDURE Sort1;
var
  i,j,ilrek:integer;
begin
  ilrek:=FileSize(zpk1b)-1;
  for i:=1 to ilrek do
    for j:=ilrek downto i do
      Begin
        seek(zpk1b,j);
        read(zpk1b,zrklb);
        seek(zpk1b,j-1);
        read(zpk1b,zrklb1);
        if zrklb1.kobu>zrklb.kobu then
          Begin
            seek(zpk1b,j-1);
            write(zpk1b,zrklb);
            write(zpk1b,zrklb1);
          End;
        End;
      End;
End;

PROCEDURE Sortowanie;
Begin
  BadDosZb;
  if jestzb then
    Begin
      clrscr;
      writeln('Sortowanie rekordow w zbiorze OBIEKTY.LDT odbywa sie');
      writeln('według kodu obiektu uderzenia');
    End;
End;
```

```
writeln;
write(' ':15,'Nacisnij klawisz Enter');
repeat
  zn:=readkey;
until zn=#13;
Sort1;
close(zpk1b);
End;
nd;
PROCEDURE WczytHaslo;
var
  whaslo:boolean;
begin
  whaslo:=false;
  pozk:=1;textbackground(white);
  textcolor(black);
  lowvideo;
  FillChar(bhas,dlh,' ');
  write(bhas);
  gotoxy(wherex-dlh,wherey);
  while not whaslo do
    Begin
      zn:=readkey;
      case zn of
        #13 : whaslo:=true;
        #0  : zn:=readkey;
      else
        Begin
          if (zn>=' ') and (zn<='9') then
            Begin
              write(zn);
              bhas[pozk]:=zn;
              pozk:=pozk+1;
            End;
          End;
        end;
      if pozk>dlh then
        Begin
          gotoxy(wherex-1,wherey);
          pozk:=pozk-1;
        End;
      End;
    textbackground(black);
    textcolor(white);
    lowvideo;
  nd;
PROCEDURE Haslo;
const
  has1:string[dlh]='SZYM';
var
  rob,
  rob1,
  hour,
  min,
```

```
sec;  
s100 :word;  
kbl :integer;  
  
begin  
whas1:=false;  
whas2:=false;  
clrscr;  
gotoxy(10,3);  
write('Podaj haslo i naciśnij klawisz Enter : ');  
WczytHaslo;  
  if has1=bhas then  
    whas1:=true  
  else  
    Begin  
      GetTime(hour,min,sec,s100);  
      Val(copy(bhas,1,1),rob,kbl);  
      if kbl=0 then  
        Begin  
          robi:=min div 10;  
          if robi=rob then  
            Begin  
              Val(copy(bhas,2,1),rob,kbl);  
              if kbl=0 then  
                Begin  
                  if rob=0 then  
                    Begin  
                      Val(copy(bhas,3,2),rob,kbl);  
                      if kbl=0 then  
                        Begin  
                          if hour=rob then  
                            whas2:=true;  
                        End;  
                      End;  
                    End;  
                  End;  
                End;  
              End;  
            End;  
          End;  
        End;  
      End;  
    End;  
  End;  
  
  if (not whas1) and (not whas2) then  
    Begin  
      gotoxy(10,5);  
      write('Bledne haslo Naciśnij klawisz Enter');  
      readln;  
      konpr:=true;  
    End;  
end;  
  
PROCEDURE DostepPelny;  
  
begin  
  clrscr;  
  gotoxy(20,5);  
  write('OBIEKTY UDERZENIA');  
  gotoxy(22,7);  
  write('1.Wprowadzanie');  
  gotoxy(22,8);  
  write('2.Przeglądanie');  
  gotoxy(22,9);  
  write('3.Aktualizacja');
```

```
gotoxy(22,10);
write('4.Usuwanie');
gotoxy(22,11);
write('5.Wyprowadzanie');
gotoxy(22,12);
write('6.Sortowanie');
gotoxy(22,13);
write('7.Wyjście');
gotoxy(22,15);
write('Wybierz numer: ');
```

```
repeat
  CzytZn;
until zn in ['1'..'7'];
```

```
wyb:=ord(zn)-48;
```

```
case wyb of
```

```
  1:WproRekDanych;
  2:PrzeGlZbioru;
  3:AktRekDanych;
  4:UsuwRekDanych;
  5:WyprZbioru;
  6:Sortowanie;
  7:konpr:=true;
```

```
End; {case}
```

```
nd;
```

```
PROCEDURE DostepOgr;
```

```
begin
```

```
  clrscr;
  gotoxy(20,5);
  write('OBIEKTY UDERZENIA');
  gotoxy(22,7);
  write('1.Przeglądanie');
  gotoxy(22,8);
  write('2.Wyprowadzanie');
  gotoxy(22,9);
  write('3.Sortowanie');
  gotoxy(22,10);
  write('4.Wyjście');
  gotoxy(22,12);
  write('Wybierz numer: ');
```

```
repeat
```

```
  CzytZn;
until zn in ['1'..'4'];
```

```
wyb:=ord(zn)-48;
```

```
case wyb of
```

```
  1:PrzeGlZbioru;
  2:WyprZbioru;
  3:Sortowanie;
  4:konpr:=true;
```

```
End; {case}
```

```
nd;
```

POCZATEK PROGRAMU }

IN

```

TextMode(2);
textbackground(black);
textcolor(white);
lowvideo;
konpr:=false;
Haslo;
while not konpr do
  Begin
    window(1,1,80,25);
    textbackground(black);
    textcolor(white);
    lowvideo;

    if whas1 then
      DostepOgr
    else
      DostepPelny;
  End; (while)

```

(Typ rekordu z pliku TAKTYCZNY PROMIEN) (DZIALANIA 36000000)

konpr: boolean;
whas1: integer;
whas2: array[1..4] of tro2;

plike: file of tro1; (typ pliku)

tro1: array[1..4] of bytes;
tro2: array[1..4] of string[25];
tro3: array[0..max] of char;

- Kod 36000000
- Kod wariantu w obronnie
- Problemy lotu w kierunku i mac. 36000000
- Problemy "pojedynczy-para" nr 300 m
- Problemy "pojedynczy-para" nr 3000 m
- Problemy "pojedynczy-para" nr 5000 m
- Problemy "pojedynczy-para" nr 10000 m
- Problemy "kider" nr 300 m
- Problemy "kider" nr 3000 m
- Problemy "kider" nr 5000 m
- Problemy "kider" nr 10000 m
- Problemy "eskadra" nr 300 m
- Problemy "eskadra" nr 3000 m
- Problemy "eskadra" nr 5000 m
- Problemy "eskadra" nr 10000 m
- Problemy lotu w kierunku i mac. 36000000
- Problemy "pojedynczy-para" nr 300 m
- Problemy "pojedynczy-para" nr 3000 m
- Problemy "pojedynczy-para" nr 5000 m

GRAM TPRD;

device crt,dos,printer;

```
begin
  ilpol=54;           {ilosc pol w rekordzie}
  dlnp=38;           {dlugosc nazwy pola}
  mdp=4;             {max dlugosc pola w znakach}
  knp=10;            {nr kolumny-poczatek nazwy pola}
  kwp=48;            {nr kolumny-poczatek wartosci pola}
  pwp=4;             {nr wiersza ekranu od ktorego
                    rozpoczyna sie wyswietlanie pol}
  owp=19;            {nr wiersza ekranu na ktorym
                    konczy sie wyswietlanie pol}
  ilwe=owp-pwp+1;    {ilosc wierszy ekranu w ktorych
                    wyswietlane sa pola}
  dlh=4;             {dlugosc hasla}

  nazp=string[dlnp]; {nazwa pola}
  tnazp=array[1..ilpol] of nazp; {typ tablicy z nazwami pol}
  trd2=record
    vl:integer;
    trd:array[1..12] of integer;
  end;
  trklb=record
    {typ rekordu z pliku TAKTYCZNY PROMIEN}
    {DZIALANIA SAMOLOTOW}
    ksam :char;
    kpwu :integer;
    trd1 :array[1..4] of trd2;
  end;
  tpklb=file of trklb; {typ pliku}

  t1=string[ilpol];
  t2=array[1..ilpol] of byte;
  t3=array[1..4] of string[22];
  t4=array[1..mdp] of char;
  t5=string[70];
  t6=string[79];
end;
```

```
begin
  tnazp1:tnazp=('Kod samolotu'
               'Kod wariantu uzbrojenia'
               'Predkosc lotu w [km/h] : max.zasieg'
               'Promien "pojedynczy-para" h= 300 m'
               'Promien "pojedynczy-para" h= 3000 m'
               'Promien "pojedynczy-para" h= 5000 m'
               'Promien "pojedynczy-para" h=10000 m'
               'Promien "klucz" h= 300 m'
               'Promien "klucz" h= 3000 m'
               'Promien "klucz" h= 5000 m'
               'Promien "klucz" h=10000 m'
               'Promien "eskadra" h= 300 m'
               'Promien "eskadra" h= 3000 m'
               'Promien "eskadra" h= 5000 m'
               'Promien "eskadra" h=10000 m'
               'Predkosc lotu w [km/h] : 800'
               'Promien "pojedynczy-para" h= 300 m'
               'Promien "pojedynczy-para" h= 3000 m'
               'Promien "pojedynczy-para" h= 5000 m'
               );
end;
```

```
'Promien "pojedynczy-para" h=10000 m
'Promien "klucz" h= 300 m
'Promien "klucz" h= 3000 m
'Promien "klucz" h= 5000 m
'Promien "klucz" h=10000 m
'Promien "eskadra" h= 300 m
'Promien "eskadra" h= 3000 m
'Promien "eskadra" h= 5000 m
'Promien "eskadra" h=10000 m
'Predkosc lotu w [km/h] : 900
'Promien "pojedynczy-para" h= 300 m
'Promien "pojedynczy-para" h= 3000 m
'Promien "pojedynczy-para" h= 5000 m
'Promien "pojedynczy-para" h=10000 m
'Promien "klucz" h= 300 m
'Promien "klucz" h= 3000 m
'Promien "klucz" h= 5000 m
'Promien "klucz" h=10000 m
'Promien "eskadra" h= 300 m
'Promien "eskadra" h= 3000 m
'Promien "eskadra" h= 5000 m
'Promien "eskadra" h=10000 m
'Predkosc lotu w [km/h] : 1000
'Promien "pojedynczy-para" h= 300 m
'Promien "pojedynczy-para" h= 3000 m
'Promien "pojedynczy-para" h= 5000 m
'Promien "pojedynczy-para" h=10000 m
'Promien "klucz" h= 300 m
'Promien "klucz" h= 3000 m
'Promien "klucz" h= 5000 m
'Promien "klucz" h=10000 m
'Promien "eskadra" h= 300 m
'Promien "eskadra" h= 3000 m
'Promien "eskadra" h= 5000 m
'Promien "eskadra" h=10000 m
```

nagl1:t5=' Taktyczne promienie dzialania samolotow myśliwsko'+  
'-bombowych';

nagl2:t6='IKOD IKOD WAI PRED I POJEDYNCZY - PARA I'+  
' KLUCZ I ESKADRA I';

nagl3:t6='ITYPU IRIANTU IKOSC I';

nagl4:t6='ISAMO I PODWIE I LOTU I'+  
' WYSOKOSC LOTU W [km] I';

nagl5:t6=' ILOTU I SZENIA I km/h I';

nagl6:t6=' I I I I 0.3 I 3 I 5 I 10 I'+  
' 0.3 I 3 I 5 I 10 I 0.3 I 3 I 5 I 10 I';

nagl7:t6=' I 1 I 2 I 3 I 4 I 5 I 6 I 7 I 8 I 9 I'+  
' 10 I 11 I 12 I 13 I 14 I 15 I';

nagl8:t6=' I-----I-----I-----I-----I-----I-----I-----I-----I'+  
'-----I-----I-----I';

ttp : t1='CIIIIIIIIIIIIIIII'+ (tablica typow pol)

'IIIIIIIIIIIIIIII'+ (upper komitatu)

'IIIIIIIIIIIIIIII'+ (tablica procedury VOT)

'IIIIIIIIIIIIIIII';

tdp : t2=(1,3,4,4,4,4,4,4,4,4,4,4,4,4,4,+ (tablica dlugosci pol)

4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,+

4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,+

```
4,4,4,4,4,4,4,4,4,4,4,4,4,4,4);
tk : t3=( ' Oczekiwana cyfra      ',      {tablica komunikatow}
        ' Bledny znak          ',
        ' Za duza liczba       ',
        ' Bledna postac liczby  ' );

r
ppole : integer;      {nr pierwszego pola na ekranie}
ostp  : integer;      {nr ostatniego pola na ekranie}
wekrk : integer;      {nr wiersza w oknie wyswietlania pol
                        rekordow ,w ktorym znajduje sie kursor}

pekrk : integer;      {nr pola wyswietlonego na ekranie ,
                        w ktorym znajduje sie kursor}

rob,
rob2,
robi  :integer;
lrek1,
lrek2,
lrek  :integer;      {licznik rekordow}
zrklb1,
zrklb :trklb;        {zmienna rekordowa}
zpk1b1,
zpk1b :tpklb;        {zmienna plikowa}
whas1,
whas2,
bl,
kpsz,
kprek,
usun,
jestzb,
jestakt,
konwd,
konpr :boolean;
zn1,
zn2,
zn3,
zn     :char;
wyb    :integer;
milrek :integer;
xk,yk  :integer;
bhas  : array[1..dlh] of char; {bufor na haslo}
buf    : t4;                 {bufor dla znakow z klawiatury}
bufk1  : array[1..4] of char;
bufk11 : char;
bufk12 : array[1..3] of char;
bufk13 : array[1..4] of char;
bk12   :string[3];
bk13   :string[4];
lpol   : integer;           {licznik pol}
ppozk,
pozka  : integer;          {wskaznik pozycji kursora w "buf"}
wczytk1,
wczytzn,
wczytpol : boolean;
nrk     : integer;          {numer komunikatu}
lzam   : longint;          {dla procedury VAL}
kbl    : integer;
lcyfr  : integer;          {licznik cyfr w "buf"}
pcb    : integer;          {pozycja pierwszej cyfry w "buf"}
```

```
PROCEDURE WczytKlucz(dkl:integer);
```

```
begin
  ww:=byte;
  ww:=0;
  wczytkl:=false;
  pozk:=1;textbackground(white);
  textcolor(black);
  lowvideo;
  FillChar(bufkl,dkl,' ');
  write(copy(bufkl,1,dkl));
  gotoxy(wherex-dkl,wherey);
  while not wczytkl do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytkl:=true;
        #0  : zn:=readkey;
      else
        Begin
          if (zn>=' ') and (zn<='~') then
            Begin
              write(zn);
              bufkl[pozk]:=zn;
              pozk:=pozk+1;
            End;
          End;
        end;
      if pozk>dkl then
        Begin
          ww:=1;
          gotoxy(wherex-1,wherey);
          pozk:=pozk-1;
        End;
      End;
    textbackground(black);
    textcolor(white);
    lowvideo;
    if ww=0 then
      pozk:=pozk-1;
    end;
end;
```

```
PROCEDURE CzytZn;
```

```
begin
  zn:=readkey;
  if zn=#0 then
    zn:=readkey;
  if zn<>#13 then
    Begin
      write(zn);
      gotoxy(wherex-1,wherey);
    End;
end;
```

```
PROCEDURE ZnPoziom(var nu:text; m:integer);
```

```
j:integer;
begin
for j:=1 to m do
write(nu, '-');
d;

PROCEDURE ZnPoz;
r
j:integer;
begin
for j:=1 to 77 do
write(#196);
d;

PROCEDURE SzabEkr;
Rysuje obrys okien }
r
k,i:integer;
const
tabl : array[1..5] of integer=(1,3,20,22,25);
PROCEDURE Pion;
r
i:integer;
begin
for i:=1 to 23 do
begin
write(#179);
gotoxy(wherex-1,wherey+1);
end;
d;

begin
clrscr;
write(#218,#191:78);
gotoxy(1,25);
write(#192,#217:78);
gotoxy(1,2);
Pion;
gotoxy(79,2);
Pion;

for k:=2 to 4 do
begin
gotoxy(1,tabl[k]);
write(#195,#180:78);
end;

for k:=1 to 5 do
begin
gotoxy(2,tabl[k]);
ZnPoz;
end;
```

```
d;
PROCEDURE PiszNazPol;
Pisze nazwy pol }
begin
case lpol of
1..11pol : write(tnazpi[lpol]);
end;
d;

PROCEDURE WczytWartPol;
Wczytuje wartosci pol }
begin
with zrk1b do
case lpol of
1: move(buf,ksam,tdp[lpol]);
2: kpwu:=lzam;
3,16,29,42 :
Begin
rob:=((lpol-3) div 13)+1;
trd1[rob].vl:=lzam;
End;
4..15,
17..28,
30..41,
43..54 :
Begin
rob:=((lpol-3) div 13)+1;
rob1:=lpol-3-13*((lpol-3) div 13);
trd1[rob].trd[rob1]:=lzam;
End;
end;
d;

PROCEDURE StrzLewo;
begin
if pozk=1 then
write(chr(7))
else
Begin
pozki:=pozki-1;
gotoxy((wherex-1),wherey);
End;
d;

PROCEDURE StrzPrawo;
begin
if pozk=tdp[lpol] then
write(chr(7))
else
Begin
pozki:=pozki+1;
gotoxy((wherex+1),wherey);
```

```
End;  
d;  
PROCEDURE KasZn;  
Kasuje znak na ktorym stoi kursor }
```

```
begin  
if (tdp[lpoll]-pozsk)>0 then  
move(buf[pozsk+1],buf[pozsk],(tdp[lpoll]-pozsk));  
buf[tdp[lpoll]]:= ' ';  
write(copy(buf,pozsk,(tdp[lpoll]-pozsk+1)));  
gotoxy((wherex-(tdp[lpoll]-pozsk+1)),wherey);  
d;
```

```
PROCEDURE WypKom(a:integer);  
Wypisuje w wierszu 21 komunikat nr 'a',  
wysyla sygnal dzwiekowy,ustawia kursor  
tam gdzie byl przed wywołaniem procedury }
```

```
var  
x,y:integer;  
begin  
x:=wherex;  
y:=wherey;  
window(2,21,78,21);  
write(tk[a]);  
gotoxy(wherex-1,wherey);  
write(chr(7));  
delay(1000);  
  
textbackground(black);  
textcolor(white);  
lowvideo;  
  
gotoxy(1,1);  
clreol;  
window(2,pwp,78,owp);  
gotoxy(x,y);  
  
textbackground(white);  
textcolor(black);  
lowvideo;  
d;
```

```
PROCEDURE SprPol;  
Sprawdza czy wczytany znak jest odpowiedni dla danego typu pola }
```

```
begin  
bl:=false;  
case ttp[lpoll] of  
'I',  
'C': if (not(zn in ['0'..'9'])) then  
Begin  
bl:=true;  
nrk:=1;  
End;  
'A': if not((zn>=' ') and (zn<='~')) then  
Begin
```

```
        bl:=true;
        nrk:=2;
    End;
end; {case}
if bl then
WypKom(nrk);
d;
PROCEDURE WypKom1;
begin
if ttp[1pol] in ['C', 'I'] then
    nrk:=1
else
    nrk:=2;
WypKom(nrk);
d;
PROCEDURE AktRez;
begin
if wyb=3 then
    Begin
        jestakt:=false;
        wczytzn:=true;
    End
else
    WypKom1;
d;
PROCEDURE WczytznPol;
Wczytuje znaki do pol }
begin
    pozk:=1;
    wczytzn:=false;
    while not wczytzn do
        Begin
            zn:=readkey;
            case zn of
                #13 : wczytzn:=true;
                #0 : Begin
                    zn:=readkey;
                    case zn of
                        #19 : AktRez;           { Alt-R }
                        #75 : StrzLewo;
                        #77 : StrzPrawo;
                        #83 : KasZn;           { Del }
                    end
                else
                    WypKom1;
            End; {case zn of przy 0}
        End {#0}
    else {case}
```

```
Begin
  SprPol;
  if not bl then
    Begin
      write(zn);
      buf[pozki]:=zn;
      pozki:=pozki+1;
    End;

    if pozki>tdp[lpoll] then
      Begin
        gotoxy((wherex-1),wherey);
        pozki:=pozki-1;
      End;

    End; {else od case}
  End; {case}
End; {while}
```

PROCEDURE SprWartPol;  
Sprawdza czy wczytana wartosc jest odpowiednia  
dla danego typu pola

```
var
  i:integer;
  wcyfr:boolean;

begin
  lcyfr:=0;
  pcb:=1;
  wcyfr:=false;

  case ttp[lpoll] of
    'I',
    'B':Begin
      for i:=1 to tdp[lpoll] do
        Begin
          if buf[i] in ['0'..'9'] then
            Begin
              lcyfr:=lcyfr+1;
              wcyfr:=true;
            End;
          if not wcyfr then
            pcb:=pcb+1;
        End;

        if lcyfr=0 then
          Begin
            kbl:=0;
            lzam:=0;
          End
        else
          val(copy(buf,pcb,lcyfr),lzam,kbl);

        if (kbl=0) then
          Begin
```

```
if (lzam>65535) and (ttp[lpol]='W') then
  Begin
    nrk:=3;
    WypKom(nrk);
    zaczytpol:=false;
  End
else
  zaczytpol:=true;

if (lzam>255) and (ttp[lpol]='B') then
  Begin
    nrk:=3;
    WypKom(nrk);
    zaczytpol:=false;
  End
else
  if ttp[lpol]='B' then
    zaczytpol:=true;
  End
else
  Begin
    nrk:=4;
    WypKom(nrk);
  End;
End;
'C' : zaczytpol:=true;
End; { case }
```

nd;

PROCEDURE WczytajPole;

```
begin
  zaczytpol:=false;
  repeat
    FillChar(buf,tdp[lpol],' ');
    gotoxy (kwp,wherey);
    textbackground(white);
    textcolor(black);
    write(copy(buf,1,tdp[lpol]));
    gotoxy(kwp,wherey);
    WczytZnPole;
    if (wyb=3) and (jestakt=false) then
      zaczytpol:=true
    else
      SprWartPol;
  until zaczytpol;

  textbackground(black);
  textcolor(white);
  lowvideo;
end;
```

PROCEDURE BadDoszB;

Bada dostep do zbioru TPRDS.LOT }

```
begin
  window(1,1,80,25);
  clrscr;
```

```
assign(zpk1b, 'TPRDS.LOT');
($I-)
reset(zpk1b);
($I+)
if IOResult<>0 then
  Begin
    gotoxy(20,5);
    write('Zbiór TPRDS.LOT niedostępny');
    gotoxy(20,wherey+2);
    write('Nacisnij klawisz Enter');
    repeat
      zn:=readkey;
    until zn=#13;
    jestzb:=false;
  end
else
  Begin
    if (FileSize(zpk1b)=0) and (wyb<>1) then
      Begin
        gotoxy(10,1);
        write('Zbiór TPRDS.LOT jest pusty');
        delay(1500);
        jestzb:=false;
        close(zpk1b);
      End
    else
      jestzb:=true;
    End;
  End;
end;

PROCEDURE WproRekDanych;
{Sluzy do wprowadzania danych }
begin
  BadDosZb;
  if not jestzb then
    Begin
      gotoxy(20,wherey+1);
      write('Czy utworzyc nowy zbior ? : [T/N] ');
      repeat
        CzytZn;
      until upcase(zn) in ['N','T'];

      if upcase(zn)='T' then
        rewrite(zpk1b)
      else
        konpr:=true;
    End;

    if not konpr then
      Begin

        milrek:=filesize(zpk1b);
        seek(zpk1b,milrek);

        konwd:=false;
        SzabEkr;
        gotoxy(2,2);
        write('Wprowadzanie danych do zbioru TAKTYCZNY PROMIEN'+
          ' DZIALANIA SAMOLOTOW');
```

```
gotoxy(2,23);
write('Enter : Koniec Del : Kasowanie');
write(' #27 : Lewo #26 : Prawo');
window(2,pwp,78,owp);
repeat
  textbackground(black);
  textcolor(white);
  lowvideo;

  clrscr;

  for lpol:=1 to ilpol do
    Begin
      gotoxy(knp,wherey);
      PiszNazPol;
      WczytajPole;
      WczytWartPol;
      writeln;
    End;
  write(zpklb,zrklb);

  write ('Czy kontynuujesz wprowadzanie ? : [T/N] ');

  repeat
    CzytZn;
    until upcase(zn) in ['N','T'];
    if upcase(zn)='N' then
      konwd:=true;
    until konwd=true;
    close(zpklb);
  End
else
  konpr:=false;
End;

PROCEDURE PiszWartPol;
Wypisuje na ekranie wartosci pol rekordu
Begin
  textbackground(white);
  textcolor(black);
  lowvideo;

  with zrklb do
    case lpol of
      1:write(ksam:1);
      2:write(kpwu:3);
      3,16,29,42 :
        Begin
          rob:=((lpol-3) div 13)+1;
          write(trd1[rob].v1:4);
        End;

      4..15,
      17..28,
      30..41,
      43..54 :
        Begin
          rob:=((lpol-3) div 13)+1;
          rob1:=lpol-3-13*((lpol-3) div 13);
          write(trd1[rob].trd[rob1]:4);
        End;
    end;
  end;
end;
```

End;

end;

textbackground(black);

textcolor(white);

lowvideo;

end;

PROCEDURE ZaznRek;

Zaznacza rekordy do usuniecia }

begin

if wyb=4 then

Begin

zrk1b.ksam:=#232;

seek(zpk1b,FilePos(zpk1b)-1);

write(zpk1b,zrk1b);

kprek:=true;

usun:=true;

End;

end;

PROCEDURE StrzGora;

begin

pekrk:=pekrk-1;

if pekrk<1 then

Begin

pekrk:=1;

write(#7);

End

else

Begin

if pekrk<ppole then

Begin

ppole:=ppole-1;

ostp:=ostp-1;

gotoxy(1,1);

InsLine;

gotoxy(knp,1);

lpol:=ppole;

PiszNazPol;

gotoxy(kwp,wherey);

PiszWartPol;

gotoxy(kwp,wherey);

End

else

Begin

wekrk:=wekrk-1;

gotoxy(wherex,wherey-1);

End;

End;

end;

PROCEDURE StrzDol;

begin

pekrk:=pekrk+1;

if pekrk>ilpol then

Begin

pekrk:=ilpol;

```
write(#7);
End
else
Begin
if pekrk>ostp then
Begin
ppole:=ppole+1;
ostp:=ostp+1;
gotoxy(1,1);
DelLine;
gotoxy(knp,ilwe);
lpol:=ostp;
PiszNazPol;
gotoxy(kwp,wherey);
PiszWartPol;
gotoxy(kwp,wherey);
End
else
Begin
wekrk:=wekrk+1;
gotoxy(wherex,wherey+1);
End;
End;
d;
PROCEDURE AktPol;
Begin
if wyb=3 then
Begin
lpol:=pekrk;
jestakt:=true;
WczytajPole;
if jestakt then
Begin
WczytWartPol;
seek(zpk1b,FilePos(zpk1b)-1);
write(zpk1b,zrk1b);
End
else
Begin
gotoxy(kwp,wherey);
PiszWartPol;
End;
gotoxy(kwp,wherey);
End;
d;
PROCEDURE Przeg1Rek;
Begin
if ilpol>ilwe then
rob2:=ilwe
else
rob2:=ilpol;
kpsz:=false;
window(2,pwp,78,owp);
```

```
while (not Eof(zpk1b)) and (not kpzb) do
  Begin
    clrscr;
    ppole:=1;
    ostp:=ilwe;
    wekrk:=1;
    pekrk:=1;

    read(zpk1b,zrklb);
    for lpol:=1 to rob2 do
      Begin
        gotoxy(knp,1+lpol-1);
        PiszNazPol;
        gotoxy(kwp,wherey);
        PiszWartPol;
        gotoxy(kwp,1);
      End;
    kprek:=false;
    while not kprek do
      Begin
        zn:=readkey;
        case zn of
          #27 : Begin { Esc }
            kprek:=true;
            kpzb:=true;
          End;
          #13 : AktPol; { Enter }
          #0 : Begin
            zn:=readkey;
            case zn of
              #22 : ZaznRek; { Alt-U }
              #72 : StrzGora;
              #80 : StrzDol;
              #81 : kprek:=true; { PgDn }
            end;
          End; { #0 }
        end;
      End; { while not kprek }
    End;
  close(zpk1b);
end;
```

PROCEDURE PrzeglZbioru;

Sluzy do przegladania zbioru bez mozliwosci zmian w danych

```
Begin
  BadDosZb;
  if jestzb then
    Begin
      SzabEkr;
      gotoxy(2,2);
      write('Przegladanie zbioru TAKTYCZNY PROMIEN'+
        ' DZIALANIA SAMOLOTOW');
      gotoxy(2,23);
      write('Esc : Wyjscie PgDn : Nastepny rekord Del : kasowanie');
      write(' #24 : Gora #25 : Dol');
      PrzeglRek;
    End; { jestzb }
  end;
```

```
nd;  
PROCEDURE UsuwRekDanych;  
Usuwa ze zbioru zaznaczone rekordy )  
  
begin  
BadDosZb;  
if jestzb then  
Begin  
SzabEkr;  
gotoxy(2,2);  
write('Usuwanie rekordow ze zbioru TAKTYCZNY PROMIEN'+  
      ' DZIALANIA SAMOLOTOW');  
gotoxy(2,23);  
write('Esc : Wyjscie PgDn : Nastepny rekord Alt-U : Usuniecie');  
gotoxy(2,24);  
write('#24' : Gora '#25' : Dol');  
usun:=false;  
PrzeglRek;  
if usun then  
Begin  
assign(zpk1b,'TPRDS.LOT');  
reset(zpk1b);  
assign(zpk1b1,'TPRDS01.LOT');  
rewrite(zpk1b1);  
while not Eof(zpk1b) do  
Begin  
read(zpk1b,zrklb);  
if zrklb.ksam<>#232 then  
write(zpk1b1,zrklb);  
End;  
close(zpk1b);  
close(zpk1b1);  
erase(zpk1b);  
rename(zpk1b1,'TPRDS.LOT');  
End;  
End;  
end;
```

```
PROCEDURE AktRekDanych;  
  
begin  
BadDosZb;  
if jestzb then  
Begin  
SzabEkr;  
gotoxy(2,2);  
write('Aktualizacja zbioru TAKTYCZNY PROMIEN'+  
      ' DZIALANIA SAMOLOTOW');  
gotoxy(2,23);  
write('Esc : Wyjscie PgDn : Nastepny rekord Del : Kasowanie ');  
write('Alt-R : Rezygnacja');  
gotoxy(2,24);  
write('Enter : Poczatek,Koniec '#27' : Lewo ');  
write('#26' : Prawo '#24' : Gora '#25' : Dol');  
PrzeglRek;  
End;  
end;
```

```
PROCEDURE WyprDanych(var nurz:text; turz:char);
```



```
else
  write(nurz,'|   |   |');
  write(nurz,trd[i].vl:4,'|');
  for j:=1 to 12 do
    write(nurz,trd[i].trd[j]:4,'|');
  writeln(nurz);
End;
Begin
  lrek:=lrek+1;
  lrek2:=lrek2+1;
  if (upcase(turz)='M') and (lrek1<>lrek2) then
    writeln(nurz,nagl8);
  if (upcase(turz)='D') and (lrek1<>lrek) then
    writeln(nurz,nagl8);
End;
End;

if zn3='3' then
  Begin
    if bufk1=ksam then
      Begin
        val(copy(bufk12,1,ppozk),lzam,kbl);
        if (kbl=0) and (lzam=kpwu) then
          Begin
            for i:=1 to 4 do
              Begin
                val(copy(bufk13,1,ppozk),lzam,kbl);
                if (kbl=0) and (lzam=trd[i].vl) then
                  Begin
                    write(nurz,'|   ',ksam:1,'|   ',kpwu:3,'|   ');
                    write(nurz,trd[i].vl:4,'|');
                    for j:=1 to 12 do
                      write(nurz,trd[i].trd[j]:4,'|');
                    writeln(nurz);
                  End;
                End;
              End;
            End;
          End;
        End;
      End;
    End;
  End;

End;

if (lrek>1) and (upcase(turz)='M')
and (FileSize(zpk1b)<>lrek2) and (zn3='1') then
  Begin
    writeln;
    write('K-Kontynuacja R-Rezygnacja');
    repeat
      CzytZn;
    until upcase(zn) in ['K','R'];
    if upcase(zn)='K' then
      Begin
        clrscr;
        lrek:=0;
      End;
    else
      kpzb:=true;
    End;
  End;

if (lrek>1) and (upcase(turz)='M')
and (lrek1<>lrek2) and (zn3='2') then
```

```
Begin
  writeln;
  write('K-Kontynuacja R-Rezygnacja');
  repeat
    CzytZn;
  until upcase(zn) in ['K','R'];
  if upcase(zn)='K' then
    Begin
      clrscr;
      lrek:=0;
    End
  else
    kpzb:=true;
  End;
End;

(upcase(turz)='M') and (upcase(zn) <> 'R') then
Begin
  write(nurz,#32);
  ZnPoziom(nurz,77);
  writeln(nurz,#32);
  writeln;
  write('Koniec zbioru Nacisnij klawisz Enter');
  repeat
    zn:=readkey;
  until zn=#13;
End;
upcase(turz)='D' then
Begin
  write(nurz,#32);
  ZnPoziom(nurz,77);
  writeln(nurz);
End;
ndow(1,1,80,25);
d;

PROCEDURE WyprZbioru;
Begin
  clrscr;
  gotoxy(2,1);
  write('Wyprowadzanie danych ze zbioru TAKTYCZNY PROMIEN'+
    ' DZIALANIA SAMOLOTOW');
  gotoxy(10,3);
  write('1.Taktyczne promienie dzialania wszystkich samolotow. ');
  gotoxy(10,4);
  write('2.Taktyczne promienie dzialania wybranego samolotu. ');
  gotoxy(10,5);
  write('3.Taktyczne promienie dzialania wybranego samolotu ');
  gotoxy(10,6);
  write(' z podanym wariantem uzbrojenia i predkoscia lotu. ');
  gotoxy(10,8);
  write('Wybierz numer: ');
  repeat
    CzytZn;
  until zn in ['1'..'3'];
End;
```

```
zn3:=zn;
if zn3 in ['2','3'] then
  Begin
    gotoxy(10,10);
    write('Podaj kod samolotu i naciśnij klawisz Enter : ');
    WczytKlucz(1);
    move(bufkl,bufkl1,1);
    irek1:=0;
    while not eof(zpk1b) do
      Begin
        read(zpk1b,zrklb);
        if bufkl1=zrklb.k$am then
          irek1:=irek1+1;
        End;
      seek(zpk1b,0);
      End;
    if zn3='3' then
      Begin
        gotoxy(10,11);
        write('Podaj kod wariantu uzbrojenia i naciśnij Enter : ');
        WczytKlucz(3);
        move(bufkl,bufkl2,3);
        ppoz:=ppoz;
        gotoxy(10,12);
        write('Podaj predkosć lotu i naciśnij Enter : ');
        WczytKlucz(4);
        move(bufkl,bufkl3,4);
        End;
      gotoxy(10,15);
      write('Podaj urządzenie wyprawdzania : [D/M] ');
      repeat
        CzytZn;
        until upcase(zn) in ['D','M'];
        zn2:=zn;
        if upcase(zn)='M' then
          Begin
            clrscr;
            WyprDanych(Output,zn2);
          End;
        else
          Begin
            gotoxy(10,17);
            write('Włącz drukarkę i naciśnij Enter');
            repeat
              zn:=readkey;
              until zn=#13;
              WyprDanych(Lst,zn2);
            End;
          close(zpk1b);
          End;
        End;
      PROCEDURE Sort1;
      var
        i,j,ilrek:integer;
```

```
begin
ilrek:=FileSize(zpk1b)-1;
for i:=1 to ilrek do
  for j:=ilrek downto i do
    Begin
      seek(zpk1b,j);
      read(zpk1b,zrklb);
      seek(zpk1b,j-1);
      read(zpk1b,zrklb1);
      if zrklb1.ksam>zrklb.ksam then
        Begin
          seek(zpk1b,j-1);
          write(zpk1b,zrklb);
          write(zpk1b,zrklb1);
        End;
      End;
    End;
end;
```

PROCEDURE Sortowanie;

```
begin
BadDosZb;
if jestzb then
  Begin
    clrscr;
    writeln('Sortowanie rekordow w zbiorze TAKTYCZNY PROMIEN'+
      ' DZIALANIA SAMOLOTOW odbywa sie');
    writeln('wedlug kodu samolotu');
    writeln;
    write(' ':15,'Nacisnij klawisz Enter');
    repeat
      zn:=readkey;
    until zn=#13;
    Sort1;
    close(zpk1b);
  End;
end;
```

PROCEDURE WczytHaslo;

```
var
  whaslo:boolean;
begin
  whaslo:=false;
  pozk:=1;textbackground(white);
  textcolor(black);
  lowvideo;
  FillChar(bhas,dlh,' ');
  write(bhas);
  gotoxy(wherex-dlh,wherey);
  while not whaslo do
    Begin
      zn:=readkey;
      case zn of
        #13 : whaslo:=true;
        #0 : zn:=readkey;
      else
        Begin

```

```
if (zn>=' ') and (zn<='~') then
  Begin
    write(zn);
    bhas[pozki]:=zn;
    pozki:=pozki+1;
  End;
End;
end;
if pozki>dlh then
  Begin
    gotoxy(wherex-1,wherey);
    pozki:=pozki-1;
  End;
End;
textbackground(black);
textcolor(white);
lowvideo;
d;
PROCEDURE Haslo;
const
  has1:string[dlh]='SZYM';
var
  rob,
  robi,
  hour,
  min,
  sec,
  s100 :word;
  kbl :integer;
begin
  whas1:=false;
  whas2:=false;
  clrscr;
  gotoxy(10,3);
  write('Podaj haslo i nacisnij klawisz Enter : ');
  WczytHaslo;
  if has1=bhas then
    whas1:=true
  else
    Begin
      GetTime(hour,min,sec,s100);
      Val(copy(bhas,1,1),rob,kbl);
      if kbl=0 then
        Begin
          robi:=min div 10;
          if robi=rob then
            Begin
              Val(copy(bhas,2,1),rob,kbl);
              if kbl=0 then
                Begin
                  if rob=0 then
                    Begin
                      Val(copy(bhas,3,2),rob,kbl);
                      if kbl=0 then
                        Begin
                          if hour=rob then
                            whas2:=true;

```

```
End;  
End;  
End;  
End;  
End;  
if (not whas1) and (not whas2) then  
Begin  
  gotoxy(10,5);  
  write('Bledne haslo Nacisnij klawisz Enter');  
  readln;  
  konpr:=true;  
End;  
end;
```

PROCEDURE DostepFelny;

```
begin  
  clrscr;  
  gotoxy(20,5);  
  write('TAKTYCZNY PROMIEN DZIALANIA SAMOLOTOW');  
  gotoxy(22,7);  
  write('1.Wprowadzanie');  
  gotoxy(22,8);  
  write('2.Przeglądanie');  
  gotoxy(22,9);  
  write('3.Aktualizacja');  
  gotoxy(22,10);  
  write('4.Usuwanie');  
  gotoxy(22,11);  
  write('5.Wyprowadzanie');  
  gotoxy(22,12);  
  write('6.Sortowanie');  
  gotoxy(22,13);  
  write('7.Wyjście');  
  gotoxy(22,15);  
  write('Wybierz numer: ');
```

```
repeat  
  CzytZn;  
until zn in ['1'..'7'];
```

```
wyb:=ord(zn)-48;
```

```
case wyb of  
  1:WproRekDanych;  
  2:PrzeglZbioru;  
  3:AktRekDanych;  
  4:UsuwRekDanych;  
  5:WyprZbioru;  
  6:Sortowanie;  
  7:konpr:=true;
```

```
End; {case}
```

PROCEDURE DostepOgr;

```
begin
```

```
clrscr;  
gotoxy(20,5);  
write('TAKTYCZNY PROMIEN DZIALANIA SAMOLOTOW');  
gotoxy(22,7);  
write('1.Przeglądanie');  
gotoxy(22,8);  
write('2.Wyprowadzanie');  
gotoxy(22,9);  
write('3.Sortowanie');  
gotoxy(22,10);  
write('4.Wyjście');  
gotoxy(22,12);  
write('Wybierz numer: ');
```

```
repeat  
  CzytZn;  
until zn in ['1'..'4'];
```

```
wyb:=ord(zn)-48;
```

```
case wyb of  
  1:PrzeGlZbioru;  
  2:WyprZbioru;  
  3:Sortowanie;  
  4:konpr:=true;
```

```
End; {case}
```

```
d;  
  
POCZATEK PROGRAMU }
```

GIN

```
TextMode(2);  
textbackground(black);  
textcolor(white);  
lowvideo;  
konpr:=false;  
Haslo;  
while not konpr do  
  . Begin  
    window(1,1,80,25);  
    textbackground(black);  
    textcolor(white);  
    lowvideo;  
  
    if whas1 then  
      DostepOgr  
    else  
      DostepPelny;  
  
  End; {while}
```

MD.



```
nagl1:t5='          Warianty uzbrojenia samolotow';
nagl2:t6=' |KOD      |ILOSC|          |'+
        ' |KOD      |ILOSC |          |'+
nagl3:t6=' |WARIANTU|RODZA|          NAZWA LSW          |'+
        ' |LSW      |SRODKOW|          |'+
nagl4:t6=' |UZBRO-  |JOW|          |'+
        ' |RAZENIA|          |'+
nagl5:t6=' |      JENIA| LSW |          |'+
        ' |      |      |          |'+
nagl6:t6=' |      1      | 2 |          |'+
        ' |      4      | 5 |          |'+
nagl7:t6=' |-----|-----|-----|-----|'+
        ' |-----|-----|-----|-----|'
```

```
ttp : t1='WBAWBABWBAWBABWB' ; {tablica typow pol}
tdp : t2=(3,1,35,3,3,35,3,3,35,3,3,+ {tablica dlugosci pol}
        35,3,3,35,3,3);
tk : t3=(' Oczekiwana cyfra      ', {tablica komunikatow}
        ' Bledny znak          ',
        ' Za duza liczba       ',
        ' Bledna postac liczby   ');
```

```
ppole : integer; {nr pierwszego pola na ekranie}
ostp  : integer; {nr ostatniego pola na ekranie}
wekrk : integer; {nr wiersza w oknie wyswietlania pol
                 rekordow ,w ktorzym znajduje sie kursor}
pekrk : integer; {nr pola wyswietlonego na ekranie ,
                 w ktorzym znajduje sie kursor}

rob,
rob2,
rob1 :integer;
lrek1,
lrek2,
lrek :integer; {licznik rekordow}
zrklb1,
zrklb :trklb; {zmienna rekordowa}
zpk1b1,
zpk1b :tpklb; {zmienna plikowa}
whas1,
whas2,
bl,
kpzb, {koniec przegladania zbioru}
kprek, {koniec przegladania rekordu}
usun,
jestzb,
jestakt,
konwd,
konpr :boolean;
zn1,
zn2,
zn3,
zn :char;
wyb :integer;
milrek :integer;
xk,yk :integer;
bhas : array[1..dlh] of char; {bufor na haslo}
buf : t4; {bufor dla znakow z klawiatury}
bufkl : array[1..3] of char;
```

```
bufkl1 : string[3];
bufkl2 : array[1..3] of char;
lpol : integer;           {licznik pol}
ppozk,
poz : integer;           {wskaznik pozycji kursora w "buf"}
wczytkl,
wczytzn,
wczytpol : boolean;
nrk : integer;           {numer komunikatu}
lzam : longint;         {dla procedury VAL}
kbl : integer;
lcyfr : integer;         {licznik cyfr w "buf"}
pcb : integer;           {pozycja pierwszej cyfry w "buf"}
```

```
PROCEDURE WczytKlucz(dkl:integer);
```

```
  ww:byte;
  begin
    ww:=0;
    wczytkl:=false;
    poz:=1;textbackground(white);
    textcolor(black);
    lowvideo;
    FillChar(bufkl,dkl,' ');
    write(copy(bufkl,1,dkl));
    gotoxy(wherex-dkl,wherey);
    while not wczytkl do
      Begin
        zn:=readkey;
        case zn of
          #13 : wczytkl:=true;
          #0  : zn:=readkey;
          else
            Begin
              if (zn>=' ') and (zn<='~') then
                Begin
                  write(zn);
                  bufkl[poz]:=zn;
                  poz:=poz+1;
                End;
              End;
            end;
          if poz>dkl then
            Begin
              ww:=1;
              gotoxy(wherex-1,wherey);
              poz:=poz-1;
            End;
          End;
        textbackground(black);
        textcolor(white);
        lowvideo;
        if ww=0 then
          poz:=poz-1;
        end;
```

```
PROCEDURE CzytZn;
```

```
begin
zn:=readkey;
if zn=#0 then
  zn1:=readkey;
if zn<>#13 then
  Begin
  write(zn);
  gotoxy(wherex-1,wherey);
  End;
d;
PROCEDURE ZnPoziom(var nu:text; m:integer);
var
j:integer;
begin
for j:=1 to m do
write(nu,'-');
d;
PROCEDURE ZnPoz;
var
j:integer;
begin
for j:=1 to 77 do
write(#196);
d;
PROCEDURE SzabEkr;
Rysuje obrys okien )
var
k,i:integer;
const
tabl : array[1..5] of integer=(1,3,20,22,25);
PROCEDURE Pion;
var
i:integer;
begin
for i:=1 to 23 do
  Begin
  write(#179);
  gotoxy(wherex-1,wherey+1);
  End;
d;
begin
clrscr;
write(#218,#191:78);
gotoxy(1,25);
write(#192,#217:78);
gotoxy(1,2);
Pion;
```

```
gotoxy(79,2);  
Pion;
```

```
for k:=2 to 4 do  
  Begin  
    gotoxy(1,tabl[k]);  
    write(#175,#180:78);  
  End;
```

```
for k:=1 to 5 do  
  Begin  
    gotoxy(2,tabl[k]);  
    ZnFoz;  
  End;
```

```
PROCEDURE PiszNazPol;  
Pisze nazwy pol ;
```

```
begin  
  case lpol of  
    1..ilpol : write(tnazpi[lpol]);  
  end;  
end;
```

```
PROCEDURE WczytWartPol;  
Wczytuje wartosci pol ;
```

```
begin  
  with zrklb do  
    case lpol of  
      1: kwu:=lzam;  
      2: ilrsw:=lzam;  
      3,6,9,12,15 :  
        Begin  
          rob:=((lpol-3) div 3)+1;  
          move(buf,trdi[rob].nlr,tdp[lpol]);  
        End;  
      4,7,10,13,16 :  
        Begin  
          rob:=((lpol-3) div 3)+1;  
          trdi[rob].klr:=lzam;  
        End;  
      5,8,11,14,17 :  
        Begin  
          rob:=((lpol-3) div 3)+1;  
          trdi[rob].ilsr:=lzam;  
        End;  
    end;  
end;
```

```
PROCEDURE StrzLewo;
```

```
begin  
  if pozk=1 then  
    write(chr(7))  
  else
```

```
Begin
  pozk:=poz-1;
  gotoxy((wherex-1),wherey);
End;
```

```
PROCEDURE StrzPrawo;
```

```
Begin
  if pozk=tdp[lpol] then
    write(chr(7))
  else
    Begin
      pozk:=poz+1;
      gotoxy((wherex+1),wherey);
    End;
End;
```

```
PROCEDURE KasZn;
  Kasuje znak na ktorym stoi kursor ;
```

```
Begin
  if (tdp[lpol]-poz)>0 then
    move(buf[poz+1],buf[poz],(tdp[lpol]-poz));
    buf[tdp[lpol]]:= ' ';
    write(copy(buf,poz,(tdp[lpol]-poz+1));
    gotoxy((wherex-(tdp[lpol]-poz+1)),wherey);
End;
```

```
PROCEDURE WypKom(a:integer);
  Wypisuje w wierszu 21 komunikat nr 'a',
  wysyla sygnal dzwiekowy,ustawia kursor
  tam gdzie byl przed wywołaniem procedury ;
```

```
var
  x,y:integer;
Begin
  x:=wherex;
  y:=wherey;
  window(2,21,78,21);
  write(tk[a]);
  gotoxy(wherex-1,wherey);
  write(chr(7));
  delay(1000);
```

```
  textbackground(black);
  textcolor(white);
  lowvideo;
```

```
  gotoxy(1,1);
  clreol;
  window(2,pwp,78,owp);
  gotoxy(x,y);
```

```
  textbackground(white);
  textcolor(black);
  lowvideo;
```

```
End;
```

PROCEDURE SprPol;  
Sprawdza czy wczytany znak jest odpowiedni dla danego typu pola )

```
begin
  bl:=false;
  case ttp[lpol] of
    'W':
      'B': if (not(zn in ['0'..'9'])) then
        Begin
          bl:=true;
          nrk:=1;
        End;
    'A': if not((zn>=' ') and (zn<='~')) then
      Begin
        bl:=true;
        nrk:=2;
      End;
  end; (case)
```

```
if bl then
  WypKom(nrk);
d;
```

PROCEDURE WypKom1;

```
begin
  if ttp[lpol] in ['W','B'] then
    nrk:=1;
  else
    nrk:=2;
  WypKom(nrk);
d;
```

PROCEDURE AktRez;

```
begin
  if wyb=3 then
    Begin
      jestakt:=false;
      wczytzn:=true;
    End
  else
    WypKom1;
d;
```

PROCEDURE WczytZnPol;  
Wczytuje znaki do pol )

```
begin
  pozk:=1;
  wczytzn:=false;
  while not wczytzn do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytzn:=true;
        #0 : Begin
```

```
if not zn:=readkey;
  case zn of
    #19 : AktRez;           { Alt-R }
    #75 : StrzLewo;
    #77 : StrzPrawo;
    #83 : KasZn;           { Del }
  else
    WypKom1;
  End; {case zn of przy 0}
End {#0}
else {case}

Begin
  SprPol;
  if not bl then
    Begin
      write(zn);
      buf[pozki]:=zn;
      pozki:=pozki+1;
    End;
  if pozki>tdp[lpoll] then
    Begin
      gotoxy((wherex-1),wherey);
      pozki:=pozki-1;
    End;
  End; {else od case}
End; {case}
End; {while}
```

PROCEDURE SprWartPol;  
Sprawdza czy wczytana wartosc jest odpowiednia  
dla danego typu pola }

```
var
  i:integer;
  wcyfr:boolean;

begin
  lcyfr:=0;
  pcb:=1;
  wcyfr:=false;

  case ttp[lpoll] of
    'W',
    'B':Begin
      for i:=1 to tdp[lpoll] do
        Begin
          if buf[i] in ['0'..'9'] then
            Begin
              lcyfr:=lcyfr+1;
              wcyfr:=true;
            End;
          else
            End;
        End;
      End;
    End;
  End;
```

```
if not wcyfr then  
  pcb:=pcb+1;  
End;
```

```
if lcyfr=0 then  
  Begin  
    kbl:=0;  
    lzam:=0;  
  End  
else  
  val(copy(buf,pcb,lcyfr),lzam,kbl);
```

```
if (kbl=0) then  
  Begin
```

```
    if (lzam>65535) and (ttp[lpol]='W') then  
      Begin  
        nrk:=3;  
        WypKom(nrk);  
        wczytpol:=false;  
      End
```

```
    else  
      wczytpol:=true;
```

```
    if (lzam>255) and (ttp[lpol]='B') then  
      Begin
```

```
        nrk:=3;  
        WypKom(nrk);  
        wczytpol:=false;  
      End
```

```
    else  
      if ttp[lpol]='B' then  
        wczytpol:=true;
```

```
    End  
  else  
    Begin  
      nrk:=4;  
      WypKom(nrk);  
    End;
```

```
End;  
'A' : wczytpol:=true;  
End; { case }
```

```
PROCEDURE WczytajPole;
```

```
Begin  
  wczytpol:=false;  
  repeat  
    FillChar(buf,tdp[lpol],' ');  
    gotoxy (kwp,wherey);  
    textbackground(white);  
    textcolor(black);  
    write(copy(buf,1,tdp[lpol]));  
    gotoxy(kwp,wherey);  
    WczytZnPol;  
    if (wyb=3) and (jestakt=false) then  
      wczytpol:=true  
    else
```

```
SprWartPol;  
until wczytpol;  
  
textbackground(black);  
textcolor(white);  
lowvideo;  
end;  
  
PROCEDURE BadDosZb;  
Bada dostep do zbioru WAR_UZB.LOT }  
  
begin  
window(1,1,80,25);  
clrscr;  
assign(zpk1b,'WAR_UZB.LOT');  
($I-)  
reset(zpk1b);  
($I+)  
if IOResult<>0 then  
begin  
gotoxy(20,5);  
write('Zbior WAR_UZB.LOT niedostepny');  
gotoxy(20,wherey+2);  
write('Nacisnij klawisz Enter');  
repeat  
zn:=readkey;  
until zn=#13;  
jestzb:=false;  
end  
else  
begin  
if (FileSize(zpk1b)=0) and (wyb<>1) then  
begin  
gotoxy(10,1);  
write('Zbior WAR_UZB.LOT jest pusty');  
delay(1500);  
jestzb:=false;  
close(zpk1b);  
end  
else  
jestzb:=true;  
end;  
end;  
  
PROCEDURE WproRekDanych;  
Sluzy do wprowadzania danych }  
  
begin  
BadDosZb;  
if not jestzb then  
begin  
gotoxy(20,wherey+1);  
write('Czy utworzyc nowy zbior ? : [T/N] ');  
repeat  
CzytZn;  
until upcase(zn) in ['N','T'];  
  
if upcase(zn)='T' then  
rewrite(zpk1b)
```

```
else
  konpr:=true;
End;

if not konpr then
  Begin
    milrek:=filesize(zpklb);
    seek(zpklb,milrek);

    konwd:=false;
    SzabEkr;
    gotoxy(2,2);
    write('Wprowadzanie danych do zbioru  WARIANTY UZBROJENIA'+
      ' SAMOLOTOW');
    gotoxy(2,23);
    write('Enter : Koniec  Del : Kasowanie');
    write('  #27 : Lewo  #26 : Prawo');
    window(2,wpw,78,owp);
    repeat
      textbackground(black);
      textcolor(white);
      lowvideo;

      clrscr;
      for lpol:=1 to ilpol do
        Begin
          gotoxy(knp,wherey);
          PiszNazPol;
          WczytajPole;
          WczytWartPol;
          writeln;
        End;
      write(zpklb,zrklb);

      write ('Czy kontynuujesz wprowadzanie ? : [T/N] ');

      repeat
        CzytZn;
      until upcase(zn) in ['N','T'];
      if upcase(zn)='N' then
        konwd:=true;
      until konwd=true;
      close(zpklb);
    End
  else
    konpr:=false;
  End;
end;

PROCEDURE PiszWartPol;
Wypisuje na ekranie wartosci pol rekordu
Begin
  textbackground(white);
  textcolor(black);
  lowvideo;

  with zrklb do
    case lpol of
```

```
1:write(kwu:tdp[lpol]);
2:write(ilrsw:tdp[lpol]);
3,6,9,12,15 :
    Begin
        rob:=((lpol-3) div 3)+1;
        write(trd1[rob].nlr:tdp[lpol]);
    End;
4,7,10,13,16 :
    Begin
        rob:=((lpol-3) div 3)+1;
        write(trd1[rob].klr:tdp[lpol]);
    End;

5,8,11,14,17 :
    Begin
        rob:=((lpol-3) div 3)+1;
        write(trd1[rob].ilsr:tdp[lpol]);
    End;
end;
textbackground(black);
textcolor(white);
lowvideo;
d;

PROCEDURE ZaznRek;
Zaznacza rekordy do usuniecia
Begin
    if wyb=4 then
        Begin
            zrklb.trd1[1].nlr[1]:=#232;
            seek(zpklb,FilePos(zpklb)-1);
            write(zpklb,zrklb);
            kprek:=true;
            usun:=true;
        End;
d;

PROCEDURE StrzGora;
Begin
    pekrk:=pekrk-1;
    if pekrk<1 then
        Begin
            pekrk:=1;
            write(#7);
        End
    else
        Begin
            if pekrk<ppole then
                Begin
                    ppole:=ppole-1;
                    ostp:=ostp-1;
                    gotoxy(1,1);
                    InsLine;
                    gotoxy(knp,1);
                    lpol:=ppole;
                    PiszNazPol;
                    gotoxy(kwp,wherey);
                    PiszWartPol;
                End;
            else
                Begin
                    kwp:=kwp-1;
                    gotoxy(kwp,wherey);
                    PiszWartPol;
                End;
            end;
        End;
end;
```

```
    gotoxy(kwp,wherey);  
End  
else  
    Begin  
        wekrk:=wekrk-1;  
        gotoxy(wherex,wherey-1);  
    End;  
End;  
d;
```

PROCEDURE StrzDol;

```
Begin  
    pekrk:=pekrk+1;  
    if pekrk>ilpol then  
        Begin  
            pekrk:=ilpol;  
            write(#7);  
        End  
    else  
        Begin  
            if pekrk>ostp then  
                Begin  
                    ppole:=ppole+1;  
                    ostp:=ostp+1;  
                    gotoxy(1,1);  
                    DelLine;  
                    gotoxy(knp,ilwe);  
                    lpol:=ostp;  
                    PiszNazPol;  
                    gotoxy(kwp,wherey);  
                    PiszWartPol;  
                    gotoxy(kwp,wherey);  
                End  
            else  
                Begin  
                    wekrk:=wekrk+1;  
                    gotoxy(wherex,wherey+1);  
                End;  
            End;  
        End;  
d;
```

PROCEDURE AktPol;

```
Begin  
    if wyb=3 then  
        Begin  
            lpol:=pekrk;  
            jestakt:=true;  
            WczytajPole;  
            if jestakt then  
                Begin  
                    WczytWartPol;  
                    seek(zpk1b,FilePos(zpk1b)-1);  
                    write(zpk1b,zrk1b);  
                End  
            else  
                Begin  
                    gotoxy(kwp,wherey);  
                    PiszWartPol;  
                End  
            End;  
        End;  
d;
```

```
End;
gotoxy(kwp,wherey);
End;
PROCEDURE Przeg1Rek;
Begin
if ilpol>ilwe then
  rob2:=ilwe
else
  rob2:=ilpol;
kpzb:=false;
window(2,pwp,78,owp);
while (not Eof(zpk1b)) and (not kpzb) do
  Begin
  clrscr;
  ppole:=1;
  ostp:=ilwe;
  wekrk:=1;
  pekrk:=1;
  read(zpk1b,zrk1b);
  for lpol:=1 to rob2 do
    Begin
      gotoxy(knp,1+lpol-1);
      PiszNazPol;
      gotoxy(kwp,wherey);
      PiszWartPol;
      gotoxy(kwp,1);
    End;
    kprek:=false;
    while not kprek do
      Begin
        zn:=readkey;
        case zn of
          #27 : Begin ( Esc )
            kprek:=true;
            kpzb:=true;
          End;
          #13 : AktPol; ( Enter )
          #0 : Begin
            zn:=readkey;
            case zn of
              #22 : ZaznRek; ( Alt-U )
              #72 : StrzGora;
              #80 : StrzDol;
              #81 : kprek:=true; ( PgDn )
            end;
          End; ( #0 )
        end;
      End; ( while not kprek )
    End;
  close(zpk1b);
```

PROCEDURE PrzeglZbioru;

Bluzy do przegladania zbioru bez mozliwosci zmian w danych )

```
begin
BadDosZb;
if jestzb then
  Begin
  SzabEkr;
  gotoxy(2,2);
  write('Przegladanie zbioru WARIANTY UZBROJENIA'+
        ' SAMOLOTOW');
  gotoxy(2,23);
  write('Esc : Wyjscie PgDn : Nastepny rekord');
  write(' #24 : Gora #25 : Dol');
  PrzeglRek;
  End; { jestzb }
d;
```

PROCEDURE UsuwRekDanych;

Usuwa ze zbioru zaznaczone rekordy )

```
begin
BadDosZb;
if jestzb then
  Begin
  SzabEkr;
  gotoxy(2,2);
  write('Usuwanie rekordow ze zbioru WARIANTY UZBROJENIA'+
        ' SAMOLOTOW');
  gotoxy(2,23);
  write('Esc : Wyjscie PgDn : Nastepny rekord Alt-U : Usuniecie');
  gotoxy(2,24);
  write('#24 : Gora #25 : Dol');
  usun:=false;
  PrzeglRek;
  if usun then
    Begin
    assign(zpk1b,'WAR_UZB.LOT');
    reset(zpk1b);
    assign(zpk1b1,'WAR_UZB1.LOT');
    rewrite(zpk1b1);
    while not Eof(zpk1b) do
      Begin
      read(zpk1b,zrk1b);
      if zrk1b.trd1[1].nlstr[1]<>#232 then
        write(zpk1b1,zrk1b);
      End;
    close(zpk1b);
    close(zpk1b1);
    erase(zpk1b);
    rename(zpk1b1,'WAR_UZB.LOT');
    End;
  End;
d;
```

PROCEDURE AktRekDanych;

```
begin
BadDosZb;
```

```
if jestzb then
Begin
  SzabEkr;
  gotoxy(2,2);
  write('Aktualizacja zbioru WARIANTY UZBROJENIA'+
        ' SAMOLOTOW');
  gotoxy(2,23);
  write('Esc : Wyjscie PgDn : Nastepny rekord Del : Kasowanie ');
  write('Alt-R : Rezygnacja');
  gotoxy(2,24);
  write('Enter : Poczatek,Koniec '#27' : Lewo ');
  write('#26' : Prawo '#24' : Gora '#25' : Dol');
  PrzeglRek;
End;
```

```
PROCEDURE WyprDanych(var nurz:text; turz:char);
```

```
var i,j:integer;
```

```
begin
  writeln(nurz,nagl1);
  writeln(nurz);
  write(nurz,#32);
  ZnPoziom(nurz,64);
  writeln(nurz);
  writeln(nurz,nagl2);
  writeln(nurz,nagl3);
  writeln(nurz,nagl4);
  writeln(nurz,nagl5);
  writeln(nurz,nagl7);
  writeln(nurz,nagl6);
  writeln(nurz,nagl7);
```

```
if upcase(turz)='M' then
```

```
  window(1,11,80,25);
  kpzb:=false;
  lrek:=0;
  lrek2:=0;
```

```
while (not eof(zpk1b)) and (not kpzb) do
```

```
  Begin
```

```
    read(zpk1b,zrklb);
```

```
    with zrklb do
```

```
      Begin
```

```
        if zn3='1' then
```

```
          Begin
```

```
            for i:=1 to 5 do
```

```
              Begin
```

```
                if i=1 then
```

```
                  write(nurz,'|',kww:tdp[1],+
                          '|',ilrsw:1,'|');
```

```
                else
```

```
                  write(nurz,'|',tdp[2],+
                          '|',tdp[3],'+
                          trd1[i].klsr:tdp[4],'+
                          trd1[i].ilsr:tdp[5],'+
                          '|');
```

```
                writeln(nurz);
```

```
              End;
```

```
            lrek:=lrek+1;
```

```
until lrek2:=lrek2+1;
if FileSize(zpk1b)<>lrek2 then
  Begin
    writeln(nurz,nagl7);
  End;
if zn3='2' then
  Begin
    str(kwu,bufkl1);
    if bufkl1[1]=bufkl1[1] then
      Begin
        for i:=1 to 5 do
          Begin
            if i=1 then
              write(nurz,'| ',kwu:tdp[1],+
                '| ',ilrsw:tdp[2],'| ');
            else
              write(nurz,'| ',tdp[i],'| ');
            write(nurz,trd1[i].nlr:tdp[3], '| ',+
              trd1[i].klr:tdp[4], '| ',+
              trd1[i].ilsr:tdp[5],'| ');
            writeln(nurz);
          End;
        lrek:=lrek+1;
      End;
    lrek2:=lrek2+1;
    if (upcase(turz)='M') and (lrek1<>lrek2) then
      writeln(nurz,nagl7);
    if (upcase(turz)='D') and (lrek1<>lrek) then
      writeln(nurz,nagl7);
  End;
End;
if zn3='3' then
  Begin
    val(copy(bufkl2,1,ppozk),lzam,kbl);
    if (kbl=0) and (lzam=kwu) then
      Begin
        for i:=1 to 5 do
          Begin
            if i=1 then
              write(nurz,'| ',kwu:tdp[1],+
                '| ',ilrsw:tdp[2],'| ');
            else
              write(nurz,'| ',tdp[i],'| ');
            write(nurz,trd1[i].nlr:tdp[3], '| ',+
              trd1[i].klr:tdp[4], '| ',+
              trd1[i].ilsr:tdp[5],'| ');
            writeln(nurz);
          End;
        End;
      End;
  End;
End;
if (lrek>1) and (upcase(turz)='M')
and (FileSize(zpk1b)<>lrek2) and (zn3='1') then
  Begin
    writeln;
    write('K-Kontynuacja R-Rezygnacja');
    repeat
      CzytZn;
    until;
  End;
```

```
until upcase(zn) in ['K','R'];
if upcase(zn)='K' then
  Begin
    clrscr;
    lrek:=0;
  End
else
  kpzb:=true;
End;

if (lrek>1) and (upcase(turz)='M')
and (lrek1<>lrek2) and (zn3='2') then
  Begin
    writeln;
    write('K-Kontynuacja R-Rezygnacja');
    repeat
      CzytZn;
    until upcase(zn) in ['K','R'];
    if upcase(zn)='K' then
      Begin
        clrscr;
        lrek:=0;
      End
    else
      kpzb:=true;
    End;
  End;

if (upcase(turz)='M') and (upcase(zn)<>'R') then
  Begin
    write(nurz,#32);
    ZnPoziom(nurz,64);
    writeln(nurz,#32);
    writeln;
    write('Koniec zbioru Nacisnij klawisz Enter');
    repeat
      zn:=readkey;
    until zn=#13;
  End;

if upcase(turz)='D' then
  Begin
    write(nurz,#32);
    ZnPoziom(nurz,64);
    writeln(nurz);
  End;
Window(1,1,80,25);
end;

PROCEDURE WyprZbioru;
begin
  BadDosZb;
  if jestzb then
    Begin
      clrscr;
      gotoxy(2,1);
      write('Wyprowadzanie danych ze zbioru WARIANTY UZBROJENIA'+
        ' SAMOLOTOW');
      gotoxy(10,3);
    End;
end;
```

```
write('1.Wszystkie warianty uzbrojenia samolotow. ');
gotoxy(10,4);
write('2.Wybrana klasa wariantow uzbrojenia samolotow. ');
gotoxy(10,5);
write('3.Wybrany wariant uzbrojenia samolotow. ');
gotoxy(10,7);
write('Wybierz numer: ');
repeat
  CzytZn;
until zn in ['1'..'3'];
zn3:=zn;

if zn3='2' then
  Begin
  gotoxy(2,9);
write('Podaj kod klasy wariantow uzbrojenia samolotow'+
  ' i naciśnij klawisz Enter : ');
WczytKlucz(1);
lrek1:=0;
while not eof(zpk1b) do
  Begin
  read(zpk1b,zrk1b);
  str(zrk1b.kwu,bufk11);
  if bufk11[1]=bufk11[1] then
    lrek1:=lrek1+1;
  End;
seek(zpk1b,0);
End;

if zn3='3' then
  Begin
gotoxy(10,9);
write('Podaj kod wariantu uzbrojenia i naciśnij Enter : ');
WczytKlucz(3);
move(bufk1,bufk12,3);
ppozk:=poz;
End;
gotoxy(10,12);
write('Podaj urządzenie wyprowadzania : [D/M] ');
repeat
  CzytZn;
until upcase(zn) in ['D','M'];
zn2:=zn;
if upcase(zn)='M' then
  Begin
  clrscr;
  WyprDanych(Output,zn2);
  End
else
  Begin
  gotoxy(10,14);
write('Włącz drukarkę i naciśnij Enter');
repeat
  zn:=readkey;
  until zn=#13;
  WyprDanych(Lst,zn2);
  End;
close(zpk1b);
End;
```

```
PROCEDURE Sort1;
```

```
var  
  i,j,ilrek:integer;  
begin  
  ilrek:=FileSize(zpklb)-1;  
  for i:=1 to ilrek do  
    for j:=ilrek downto i do  
      Begin  
        seek(zpklb,j);  
        read(zpklb,zrklb);  
        seek(zpklb,j-1);  
        read(zpklb,zrklb1);  
        if zrklb1.kwu>zrklb.kwu then  
          Begin  
            seek(zpklb,j-1);  
            write(zpklb,zrklb);  
            write(zpklb,zrklb1);  
          End;  
        End;  
      End;  
end;
```

```
PROCEDURE Sortowanie;
```

```
var  
  BadDosZb;  
if jestzb then  
  Begin  
    clrscr;  
    writeln('Sortowanie rekordow w zbiorze WARIANTY UZBROJENIA'+  
            ' SAMOLOTOW odbywa sie');  
    writeln('wedlug kodu wariantu uzbrojenia');  
    writeln;  
    write(' ':15,'Nacisnij klawisz Enter');  
    repeat  
      zn:=readkey;  
    until zn=#13;  
    Sort1;  
    close(zpklb);  
  End;
```

```
PROCEDURE WczytHaslo;
```

```
var  
  whaslo:boolean;  
begin  
  whaslo:=false;  
  pozk:=1;textbackground(white);  
  textcolor(black);  
  lowvideo;  
  FillChar(bhas,dlh,' ');  
  write(bhas);  
  gotoxy(wherex-dlh,wherey);  
  while not whaslo do  
    Begin  
      zn:=readkey;
```

```
case zn of
  #13 : whaslo:=true;
  #0  : zn:=readkey;
else
  Begin
    if (zn>=' ') and (zn<='~') then
      Begin
        write(zn);
        bhas[pozki]:=zn;
        pozki:=pozki+1;
      End;
    End;
  end;
if pozki>dlh then
  Begin
    gotoxy(wherex-1,wherey);
    pozki:=pozki-1;
  End;
End;
textbackground(black);
textcolor(white);
lowvideo;
end;
```

PROCEDURE Haslo;

```
const
  hasl:string[dlh]='SZYM';
var
  rob,
  robi,
  hour,
  min,
  sec,
  sl00 :word;
  kbl :integer;
begin
  whas1:=false;
  whas2:=false;
  clrscr;
  gotoxy(10,3);
  write('Podaj haslo i naciśnij klawisz Enter : ');
  WczytHaslo;
  if hasl=bhas then
    whas1:=true
  else
    Begin
      GetTime(hour,min,sec,sl00);
      Val(copy(bhas,1,1),rob,kbl);
      if kbl=0 then
        Begin
          robi:=min div 10;
          if robi=rob then
            Begin
              Val(copy(bhas,2,1),rob,kbl);
              if kbl=0 then
                Begin
                  if rob=0 then
```

```
Begin
  Val(copy(bhas,3,2),rob,kbl);
  if kbl=0 then
    Begin
      if hour=rob then
        whas2:=true;
      End;
    End;
  End;
End;

if (not whas1) and (not whas2) then
  Begin
    gotoxy(10,5);
    write('Bledne haslo Nacisnij klawisz Enter');
    readln;
    konpr:=true;
  End;
End;
```

PROCEDURE DostepPelny;

```
begin
  clrscr;
  gotoxy(20,5);
  write('WARIANTY UZBROJENIA SAMOLOTOW');
  gotoxy(22,7);
  write('1.Wprowadzanie');
  gotoxy(22,8);
  write('2.Przeglądanie');
  gotoxy(22,9);
  write('3.Aktualizacja');
  gotoxy(22,10);
  write('4.Usuwanie');
  gotoxy(22,11);
  write('5.Wyprowadzanie');
  gotoxy(22,12);
  write('6.Sortowanie');
  gotoxy(22,13);
  write('7.Wyjście');
  gotoxy(22,15);
  write('Wybierz numer: ');

  repeat
    CzytZn;
  until zn in ['1'..'7'];

  wyb:=ord(zn)-48;

  case wyb of
    1:WproRekDanych;
    2:PrzeGlZbioru;
    3:AktRekDanych;
    4:UsuwRekDanych;
    5:WyprZbioru;
    6:Sortowanie;
    7:konpr:=true;
  end;
```

```
End; {case}
nd;

PROCEDURE DostepOgr;

begin
  clrscr;
  gotoxy(20,5);
  write('WARIANTY UZBROJENIA SAMOLOTOW');
  gotoxy(22,7);
  write('1.Przeglądanie');
  gotoxy(22,8);
  write('2.Wyprowadzanie');
  gotoxy(22,9);
  write('3.Sortowanie');
  gotoxy(22,10);
  write('4.Wyjście');
  gotoxy(22,12);
  write('Wybierz numer: ');

  repeat
    CzytZn;
  until zn in ['1'..'4'];

  wyb:=ord(zn)-48;

  case wyb of
    1:PrzeZbioru;
    2:WyprZbioru;
    3:Sortowanie;
    4:konpr:=true;

  End; {case}
nd;
```

POCZATEK PROGRAMU }

```
EGIN
  TextMode(2);
  textbackground(black);
  textcolor(white);
  lowvideo;
  konpr:=false;
  Haslo;
  while not konpr do
    Begin
      window(1,1,80,25);
      textbackground(black);
      textcolor(white);
      lowvideo;

      if whasi then
        DostepOgr
      else
        DostepPelny;

    End; {while}
ND.
```

```
PROGRAM DWFOLS;
uses crt,dos,printer;

const
  ilpol=158; {ilosc pol w rekordzie (bez pola)
              {kontrolnego)}
  dlnp=31;   {dlugosc nazwy pola}
  mdp=3;     {max dlugosc pola w znakach}
  knp=10;    {nr kolumny-początek nazwy pola}
  kwp=44;    {nr kolumny-początek wartosci pola}
  pwp=5;     {nr wiersza ekranu od ktorego
              rozpoczyna sie wyswietlanie pol}
  owp=19;    {nr wiersza ekranu na ktorym
              konczy sie wyswietlanie pol}
  ilwe=owp-pwp+1; {ilosc wierszy ekranu w ktorych
                  wyswietlane sa pola}
  dlh=4;     {dlugosc hasla}

type
  nazp=string[dlnp]; {nazwa pola}
  tnazp=array[1..ilpol] of nazp; {typ tablicy z nazwami pol}
  pls1=record
    p6 :word;
    p7 :array[1..3,1..3] of byte;
  end;
  pls2=record
    p8 :word;
    p9 :array[1..3] of byte;
  end;
  trklb=record {typ rekordu z pliku ORIENTACYJNA WARTOSC}
    {POTRZEBNEJ OGNIOWEJ (POLIGONOWEJ) LICZBY}
    {SAMOLOTOW M-B}
    p1 :byte;
    p2 :byte;
    p3 :array[1..10] of pls1;
    p4 :array[1..6] of pls2;
    p5 :array[1..7] of pls2;
    pk :char; {pole kontrolne}
  end;
  tpklb=file of trklb; {typ pliku}

  t1=string[ilpol];
  t2=array[1..ilpol] of byte;
  t3=array[1..4] of string[22];
  t4=array[1..mdp] of char;
  t6=string[76];
  t7=string[52];

const
  n1=' Zniszczenie ';
  n2=' Obezwladnienie ';
  n3=' Dezorganizacja ';
  n4='LN';
  n5='LP';
  n6='LW';
  n7='Kod wariantu uzbrojenia';
  tnazpl:tnazp=('Kod typu samolotu','Kod obiektu uderzenia');
```





```
bk13 :string[4];  
lpol : integer;           {licznik pol}  
ppozk,  
pozka : integer;        {wskaznik pozycji kursora w "buf"}  
wczytkl,  
wczytzn,  
wczytpol : boolean;  
nrk : integer;          {numer komunikatu}  
lzam,  
lzam1 : longint;       {dla procedury VAL}  
kbl,  
kbl1 : integer;  
lcyfr: integer;        {licznik cyfr w "buf"}  
pcb : integer;         {pozycja pierwszej cyfry w "buf"}
```

```
PROCEDURE Wczytklucz(dkl:integer);
```

```
var  
  ww:byte;  
begin  
  ww:=0;  
  wczytkl:=false;  
  pozka:=1;textbackground(white);  
  textcolor(black);  
  lowvideo;  
  FillChar(bufkl,dkl,' ');  
  write(copy(bufkl,1,dkl));  
  gotoxy(wherex-dkl,wherey);  
  while not wczytkl do  
    Begin  
      zn:=readkey;  
      case zn of  
        #13 : wczytkl:=true;  
        #0  : zn:=readkey;  
        else  
          Begin  
            if (zn>=' ') and (zn<='~') then  
              Begin  
                write(zn);  
                bufkl[pozka]:=zn;  
                pozka:=pozka+1;  
              End;  
            End;  
          end;  
          if pozka>dkl then  
            Begin  
              ww:=1;  
              gotoxy(wherex-1,wherey);  
              pozka:=pozka-1;  
            End;  
          End;  
        textbackground(black);  
        textcolor(white);  
        lowvideo;  
        if ww=0 then  
          pozka:=pozka-1;  
        end;  
      end;
```

```
PROCEDURE CzytZn;
```

```
begin  
  zn:=readkey;  
  if zn=#0 then  
    zn1:=readkey;  
  if zn<>#13 then  
    Begin  
      write(zn);  
      gotoxy(wherex-1,wherey);  
    End;
```

```
end;
```

```
PROCEDURE ZnPoziom(var nu:text; m:integer);
```

```
var  
  j:integer;  
begin  
  for j:=1 to m do  
    write(nu,'-');
```

```
end;
```

```
PROCEDURE ZnPoz;
```

```
var  
  j:integer;  
begin  
  for j:=1 to 77 do  
    write(#196);
```

```
end;
```

```
PROCEDURE SzabEkr;  
  Rysuje obrys okien )
```

```
var  
  k,i:integer;  
const  
  tabl : array[1..5] of integer=(1,4,20,22,25);
```

```
PROCEDURE Pion;
```

```
var  
  i:integer;  
begin  
  for i:=1 to 23 do  
    Begin  
      write(#179);  
      gotoxy(wherex-1,wherey+1);  
    End;
```

```
end;
```

```
begin  
  clrscr;  
  write(#218,#191:78);  
  gotoxy(1,25);  
  write(#192,#217:78);  
  gotoxy(1,2);
```

```
on;  
gotoxy(79,2);  
on;  
for k:=2 to 4 do  
Begin  
gotoxy(1,tab1[k]);  
write(#195,#180:78);  
End;
```

```
for k:=1 to 5 do  
Begin  
gotoxy(2,tab1[k]);  
ZnPoz;  
End;
```

```
PROCEDURE PiszNazPol;  
pisze nazwy pol )
```

```
in  
case lpol of  
1..1pol : write(tnazp1[lpol]);
```

nd;

```
PROCEDURE WczytWartPol;  
czytuje wartosci pol )
```

```
in  
with zrklb do  
case lpol of
```

```
1: p1:=lzam;  
2: p2:=lzam;  
4,14,24,34,44,  
54,64,74,84,94 :
```

```
Begin  
rob:=(lpol div 10)+1;  
p3[rob].p6:=lzam;  
End;
```

```
5..13,15..23,  
25..33,35..43,  
45..53,55..63,  
65..73,75..83,  
85..93,95..103 :
```

```
Begin  
rob:=((lpol-5) div 10);  
rob1:=rob+1;  
rob2:=(lpol-5-10*rob);  
rob3:=(rob2 div 3)+1;  
rob4:=rob2-3*(rob3-1)+1;  
p3[rob1].p7[rob3,rob4]:=lzam;  
End;
```

```
105,109,113,  
117,121,125 :
```

```
Begin  
rob:=((lpol-105) div 4)+1;
```

```
      p4[rob].p8:=lzam;
    End;
106..108,110..112,
114..116,118..120,
122..124,126..128 :
      Begin
        rob:=((lpol-105) div 4)+1;
        rob1:=lpol-105-4*((lpol-105) div 4);
        p4[rob].p9[rob1]:=lzam;
      End;

131,135,139,
143,147,151,
155 :
      Begin
        rob:=((lpol-131) div 4)+1;
        p5[rob].p8:=lzam;
      End;

132..134,136..138,
140..142,144..146,
148..150,152..154,
156..158 :
      Begin
        rob:=((lpol-131) div 4)+1;
        rob1:=lpol-131-4*((lpol-131) div 4);
        p5[rob].p9[rob1]:=lzam;
      End;
end;
```

PROCEDURE StrzLewo;

```
begin
  if pozk=1 then
    write(chr(7))
  else
    Begin
      pozk:=poz-1;
      gotoxy((wherex-1),wherey);
    End;
end;
```

PROCEDURE StrzPrawo;

```
begin
  if pozk=tdp[lpol] then
    write(chr(7))
  else
    Begin
      pozk:=poz+1;
      gotoxy((wherex+1),wherey);
    End;
end;
```

PROCEDURE KasZn;

```
  Kasuje znak na ktorym stoi kursor )
begin
  if (tdp[lpol]-pozk)>0 then
    move(buf[poz+1],buf[pozk],(tdp[lpol]-pozk));
  buf[tdp[lpol]]:= ' ';
end;
```

```
write(copy(buf, pozk, (tdp[lpoll]-pozk+1)));  
gotoxy((wherex-(tdp[lpoll]-pozk+1)), wherey);  
end;
```

```
PROCEDURE WypKom(a:integer);  
Wypisuje w wierszu 21 komunikat nr 'a',  
wysyla sygnal dzwiekowy, ustawia kursor  
tam gdzie byl przed wywołaniem procedury }
```

```
var  
x,y:integer;  
begin  
x:=wherex;  
y:=wherey;  
window(2,21,78,21);  
write(tk[a]);  
gotoxy(wherex-1,wherey);  
write(chr(7));  
delay(1000);  
  
textbackground(black);  
textcolor(white);  
lowvideo;  
  
gotoxy(1,1);  
clreol;  
window(2,pwp,78,owp);  
gotoxy(x,y);  
  
textbackground(white);  
textcolor(black);  
lowvideo;  
end;
```

```
PROCEDURE SprPol;  
Sprawdza czy wczytany znak jest odpowiedni dla danego typu pola }
```

```
begin  
bl:=false;  
case ttp[lpoll] of  
'W':  
'B': if (not(zn in ['0'..'9'])) then  
Begin  
bl:=true;  
nrk:=1;  
End;  
'A': if not((zn>=' ') and (zn<='~')) then  
Begin  
bl:=true;  
nrk:=2;  
End;  
end; (case)  
  
if bl then  
WypKom(nrk);  
end;
```

```
PROCEDURE WypKom1;
```

```
Begin
  gotoxy(wherex-1, wherex);
in
  f ttp[lpol] in ['W', 'B'] then
    nrk:=1
  lse
    nrk:=2;
  ypkom(nrk);
  ;
PROCEDURE AktRez;

in
  f wyb=3 then
    Begin
      jestakt:=false;
      wczytzn:=true;
    End
  lse
    WypKom1;
  ;
PROCEDURE WczytznPol;
czytuje znaki do pol ?

in
  oznk:=1;
  czytzn:=false;
  while not wczytzn do
    Begin
      zn:=readkey;
      case zn of
        #13 : wczytzn:=true;
        #0 : Begin
          zn:=readkey;
        End;
        case zn of
          #19 : AktRez;           { Alt-R }
          #75 : StrzLewo;
          #77 : StrzPrawo;
          #83 : KasZn;           { Del }
        end;
      else
        WypKom1;
      End; {case zn of przy 0}
    End {#0}
  else {case}

    Begin
      SprPol;
      if not bl then
        Begin
          write(zn);
          buf[pozki]:=zn;
          pozki:=pozki+1;
        End;
      if pozki>tdp[lpol] then
```

```
Begin
  gotoxy((wherex-1),wherey);
  pozk:=poz-1;
End;
End;
End; (else od case)
End; (case)
End; (while)
End;

PROCEDURE SprWartPol;
Sprawdza czy wczytana wartosc jest odpowiednia
dla danego typu pola )

var
  i:integer;
  wcyfr:boolean;

begin
  lcyfr:=0;
  pcb:=1;
  wcyfr:=false;

  case ttp[lpol] of
    'W',
    'B' :Begin
      for i:=1 to tdp[lpol] do
        Begin
          if buf[i] in ['0'..'9'] then
            Begin
              lcyfr:=lcyfr+1;
              wcyfr:=true;
            End;
          if not wcyfr then
            pcb:=pcb+1;
          End;

          if lcyfr=0 then
            Begin
              kbl:=0;
              lzam:=0;
            End
          else
            val(copy(buf,pcb,lcyfr),lzam,kbl);

            if (kbl=0) then
              Begin
                if (lzam>65535) and (ttp[lpol]='W') then
                  Begin
                    nrk:=3;
                    WypKom(nrk);
                    wczytpol:=false;
                  End
                else
                  wczytpol:=true;
                End
              End
            if (lzam>255) and (ttp[lpol]='B') then

```

```
Begin
  nrk:=3;
  WypKom(nrk);
  zaczytpol:=false;
  repeat
    End
  else
    if ttp[1pol]='B' then
      zaczytpol:=true;
    End
  else
    Begin
      nrk:=4;
      WypKom(nrk);
    End;
  End;
End; { case }
```

```
PROCEDURE WczytajPole;
```

```
Begin
  zaczytpol:=false;
  repeat
    if tdp[1pol]=0 then
      zaczytpol:=true
    else
      Begin
        FillChar(buf,tdp[1pol],' ');
        gotoxy(kwp,wherey);
        textbackground(white);
        textcolor(black);
        write(copy(buf,1,tdp[1pol]));
        gotoxy(kwp,wherey);
        WczytZnPol;
        if (wyb=3) and (jestakt=false) then
          zaczytpol:=true
        else
          SprWartPol;
        End;
      End;
  until zaczytpol;

  textbackground(black);
  textcolor(white);
  lowvideo;
```

```
PROCEDURE BadDosZb;
Bada dostep do zbioru OWPOLS.LOT )
```

```
Begin
  window(1,1,80,25);
  clrscr;
  assign(zpk1b,'OWPOLS.LOT');
  {$I-}
  reset(zpk1b);
  {$I+}
  if IOResult<>0 then
    Begin
```

```
gotoxy(20,5);
write('Zbior OWFOLS.LOT niedostepny');
gotoxy(20,wherey+2);
write('Nacisnij klawisz Enter');
repeat
  zn:=readkey;
until zn=#13;
jestzb:=false
end
else
Begin
  if (FileSize(zpklb)=0) and (wyb<>1) then
    Begin
      gotoxy(10,1);
      write('Zbior OWFOLS.LOT jest pusty');
      delay(1500);
      jestzb:=false;
      close(zpklb);
    End
  else
    jestzb:=true;
  End;
end;

PROCEDURE WproRekDanych;
  Sluzy do wprowadzania danych }

begin
  BadDosZb;
  if not jestzb then
    Begin
      gotoxy(20,wherey+1);
      write('Czy utworzyc nowy zbior ? : [T/N] ');
      repeat
        CzytZn;
      until upcase(zn) in ['N','T'];

      if upcase(zn)='T' then
        rewrite(zpklb)
      else
        konpr:=true;
    End;

    if not konpr then
      Begin

        milrek:=filesize(zpklb);
        seek(zpklb,milrek);

        konwd:=false;
        SzabEkr;
        gotoxy(2,2);
        write('Wprowadzanie danych do zbioru  ORIENTACYJNA WARTOSC POTRZEBNEJ');
        gotoxy(2,3);
        write('OGNIOWEJ (POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');
        gotoxy(2,23);
        write('Enter : Koniec  Del : Kasowanie');
        write('  #27 : Lewo  #26 : Prawo');
        window(2,pwp,78,owp);
        repeat
```

```
textbackground(black);
textcolor(white);
lowvideo;
```

```
clrscr;
```

```
for lpol:=1 to ilpol do
```

```
Begin
```

```
gotoxy(knp,wherey);
```

```
PiszNazPol;
```

```
WczytajPole;
```

```
WczytWartPol;
```

```
writeln;
```

```
End;
```

```
write(zpk1b,zrk1b);
```

```
write ('Czy kontynuujesz wprowadzanie ? : [T/N] ');
```

```
repeat
```

```
  CzytZn;
```

```
until upcase(zn) in ['N','T'];
```

```
if upcase(zn)='N' then
```

```
  konwd:=true;
```

```
until konwd=true;
```

```
close(zpk1b);
```

```
End
```

```
else
```

```
konpr:=false;
```

```
End;
```

```
PROCEDURE PiszWartPol;
```

```
Wypisuje na ekranie wartosci pol rekordu }
```

```
Begin
```

```
textbackground(white);
```

```
textcolor(black);
```

```
lowvideo;
```

```
with zrk1b do
```

```
  case lpol of
```

```
    1: write(p1:tdp[lpol]);
```

```
    2: write(p2:tdp[lpol]);
```

```
    4,14,24,34,44,
```

```
    54,64,74,84,94 :
```

```
      Begin
```

```
        rob:=(lpol div 10)+1;
```

```
        write(p3[rob].p6:tdp[lpol]);
```

```
      End;
```

```
    5..13,15..23,
```

```
    25..33,35..43,
```

```
    45..53,55..63,
```

```
    65..73,75..83,
```

```
    85..93,95..103 :
```

```
      Begin
```

```
        rob:=((lpol-5) div 10);
```

```
        rob1:=rob+1;
```

```
        rob2:=(lpol-5-10*rob);
```

```
        rob3:=(rob2 div 3)+1;
```

```
        rob4:=rob2-3*(rob3-1)+1;
```

```
        write(p3[rob1].p7[rob3,rob4]:tdp[lpol]);
    End;

105,109,113,
117,121,125 :
    Begin
        rob:=((lpol-105) div 4)+1;
        write(p4[rob].p8:tdp[lpol]);
    End;

106..108,110..112,
114..116,118..120,
122..124,126..128 :
    Begin
        rob:=((lpol-105) div 4)+1;
        rob1:=lpol-105-4*((lpol-105) div 4);
        write(p4[rob].p9[rob1]:tdp[lpol]);
    End;

131,135,139,
143,147,151,
155 :
    Begin
        rob:=((lpol-131) div 4)+1;
        write(p5[rob].p8:tdp[lpol]);
    End;

132..134,136..138,
140..142,144..146,
148..150,152..154,
156..158 :
    Begin
        rob:=((lpol-131) div 4)+1;
        rob1:=lpol-131-4*((lpol-131) div 4);
        write(p5[rob].p9[rob1]:tdp[lpol]);
    End;

end;
textbackground(black);
textcolor(white);
lowvideo;
end;

PROCEDURE ZaznRek;
Zaznacza rekordy do usuniecia
Begin
    if wyb=4 then
        Begin
            zrklb.pk:=#232;
            seek(zpklb,FilePos(zpklb)-1);
            write(zpklb,zrklb);
            kprek:=true;
            usun:=true;
        End;
    end;

PROCEDURE StrzGora;
Begin
    pekrk:=pekrk-1;
    if pekrk<1 then
        Begin
            pekrk:=1;
            write(#7);
        End;
    end;
end;
```

```
End  
else  
  Begin  
    if pekrk < ppole then  
      Begin  
        ppole := ppole - 1;  
        ostp := ostp - 1;  
        gotoxy(1, 1);  
        InSLine;  
        gotoxy(knp, 1);  
        lpol := ppole;  
        PiszNazPol;  
        gotoxy(kwp, wherey);  
        PiszWartPol;  
        gotoxy(kwp, wherey);  
      End  
    else  
      Begin  
        wekrk := wekrk - 1;  
        gotoxy(wherex, wherey - 1);  
      End;  
    End;  
  End;
```

```
PROCEDURE StrzDol;
```

```
Begin  
  pekrk := pekrk + 1;  
  if pekrk > ilpol then  
    Begin  
      pekrk := ilpol;  
      write(#7);  
    End  
  else  
    Begin  
      if pekrk > ostp then  
        Begin  
          ppole := ppole + 1;  
          ostp := ostp + 1;  
          gotoxy(1, 1);  
          DelLine;  
          gotoxy(knp, ilwe);  
          lpol := ostp;  
          PiszNazPol;  
          gotoxy(kwp, wherey);  
          PiszWartPol;  
          gotoxy(kwp, wherey);  
        End  
      else  
        Begin  
          wekrk := wekrk + 1;  
          gotoxy(wherex, wherey + 1);  
        End;  
      End;  
    End;  
  End;
```

```
PROCEDURE AktPol;
```

```
Begin  
  if wyb = 3 then
```

```
Begin
lpol:=pekrk;
jestakt:=true;
WczytajPole;
if jestakt then
  Begin
  WczytWartPol;
  seek(zpk1b,FilePos(zpk1b)-1);
  write(zpk1b,zrk1b);
  End
else
  Begin
  gotoxy(kwp,wherey);
  PiszWartPol;
  End;
  gotoxy(kwp,wherey);
  End;
nd;

PROCEDURE Przeg1Rek;

begin

  if ilpol>ilwe then
    rob5:=ilwe
  else
    rob5:=ilpol;

  kpzb:=false;

  window(2,pwp,78,owp);

  while (not Eof(zpk1b)) and (not kpzb) do
    Begin
      clrscr;
      ppole:=1;
      ostp:=ilwe;
      wekrk:=1;
      pekrk:=1;

      read(zpk1b,zrk1b);
      for lpol:=1 to rob5 do
        Begin
          gotoxy(knp,1+lpol-1);
          PiszNazPol;
          gotoxy(kwp,wherey);
          PiszWartPol;
          gotoxy(kwp,1);
        End;
      kprek:=false;
      while not kprek do
        Begin
          zn:=readkey;
          case zn of
            #27 : Begin { Esc }
                    kprek:=true;
                    kpzb:=true;
                  End;
            #13 : AktPol; { Enter }
            #0  : Begin
```

```
      read(zpk1b, zn:=readkey);
      case zn of
        #22 : ZaznRek;      ( Alt-U )
        #72 : StrzGora;
        #80 : StrzDol;
        #81 : kprek:=true; ( PgDn )
      end;
      End; ( #0 )
    end;
  End; ( while not kprek )
End;
close(zpk1b);
```

PROCEDURE PrzeglZbioru;

Sluzy do przegladania zbioru bez mozliwosci zmian w danych )

```
begin
BadDosZb;
if jestzb then
  Begin
  SzabEkr;
  gotoxy(2,2);
  write('Przegladanie zbioru ORIENTACYJNA WARTOSC POTRZEBNEJ');
  gotoxy(2,3);
  write('OGNIOWEJ(POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');
  gotoxy(2,23);
  write('Esc : Wyjscie PgDn : Nastepny rekord');
  write('#24' : Gora '#25' : Dol');
  PrzeglRek;
  End; ( jestzb )
```

PROCEDURE UsuwRekDanych;

Usuwa ze zbioru zaznaczone rekordy )

```
begin
BadDosZb;
if jestzb then
  Begin
  SzabEkr;
  gotoxy(2,2);
  write('Usuwanie rekordow ze zbioru ORIENTACYJNA WARTOSC POTRZEBNEJ');
  gotoxy(2,3);
  write('OGNIOWEJ(POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');
  gotoxy(2,23);
  write('Esc : Wyjscie PgDn : Nastepny rekord Alt-U : Usuniecie');
  gotoxy(2,24);
  write('#24' : Gora '#25' : Dol');
  usun:=false;
  PrzeglRek;
  if usun then
    Begin
      assign(zpk1b, 'OWPOL.S.LOT');
      reset(zpk1b);
      assign(zpk1b1, 'OWPOLSOI.LOT');
      rewrite(zpk1b1);
      while not Eof(zpk1b) do
        Begin
```

```
    read(zpk1b,zrk1b);
    if zrk1b.pk<>#232 then
        write(zpk1b1,zrk1b);
    End;
    close(zpk1b);
    close(zpk1b1);
    erase(zpk1b);
    rename(zpk1b1,'OWPOL.S.LOT');
End;
End;
nd;

PROCEDURE AktRekDanych;
begin
    BadDosZb;
    if jestzb then
        Begin
            SzabEkr;
            gotoxy(2,2);
            write('Aktualizacja zbioru  ORIENTACYJNA WARTOSC POTRZEBNEJ');
            gotoxy(2,3);
            write('OGNIOWEJ (POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');
            gotoxy(2,23);
            write('Esc : Wyjscie PgDn : Nastepny rekord Del : Kasowanie ');
            write('Alt-R : Rezygnacja');
            gotoxy(2,24);
            write('Enter : Poczatek,Koniec '#27' : Lewo ');
            write('#26' : Prawo '#24' : Gora '#25' : Dol');
            PrzeglRek;
        End;
    nd;

PROCEDURE WyprDanych(var nurz:text; turz:char);
var
    i,j,k:integer;
begin
    if zn4='1' then
        Begin
            writeln(nurz,' '+nagl1);
            writeln(nurz,' '+nagl2);
            writeln(nurz);

            writeln(nurz,'A.UZBROJENIE BOMBARDIERSKIE');
            write(nurz,#32);
            ZnPoziom(nurz,74);
            writeln(nurz);
            writeln(nurz,nagl7);
            writeln(nurz,nagl8);
            writeln(nurz,nagl9);
        End
    else
        Begin
            writeln(nurz,nagl1);
            writeln(nurz,' '+nagl2);
            writeln(nurz);

            if zn4='2' then
```



```
else
  write(nurz, ' ');
write(nurz, ' ', p4[i].p8:3, ' ');
for j:=1 to 3 do
  write(nurz, ' ', p4[i].p9[j]:3, ' ');
writeln(nurz);
End;
End; { zn4='2' }
```

```
if zn4='3' then
  Begin
    writeln(nurz, nagl14);
    for i:=1 to 7 do
      Begin
        if i=1 then
          write(nurz, ' ', p1:1, ' ', p2:2, ' ');
        else
          write(nurz, ' ');
          write(nurz, ' ', p5[i].p8:3, ' ');
          for j:=1 to 3 do
            write(nurz, ' ', p5[i].p9[j]:3, ' ');
          writeln(nurz);
        End;
      End;
    End; { zn4='3' }
```

```
if upcase(turz)='M' then
  Begin
    writeln;
    write('K-Kontynuacja R-Rezygnacja ');
    repeat
      CzytZn;
    until upcase(zn) in ['K', 'R'];
    if upcase(zn)='K' then
      clrscr
    else
      kpzb:=true;
    End;
  End;
```

```
End; { zn3 in ['1', '2', '3'] }
```

```
if zn3='4' then
  Begin
    if (kbl=0) and (kbl1=0) then
      Begin
        if lzam=p1 then
          Begin
            if zn4='1' then
              Begin
                for i:=1 to 10 do
                  Begin
                    if lzam1=p3[i].p6 then
                      Begin
                        writeln(nurz, nagl10);
                        write(nurz, ' ', p1:1, ' ', p2:2, ' ', p3[i].p6:3, ' ');
                        for j:=1 to 3 do
                          Begin
```

```
End;
for k:=1 to 3 do
  write(nurz,' ',p3[i].p7[j,k]:3,' ');
End;
writeln(nurz);
End;
End;
lr:=lr+1;
End; { zn4='1' }

if zn4='2' then
  Begin
    for i:=1 to 6 do
      Begin
        if lzam1=p4[i].p8 then
          Begin
            writeln(nurz,nagl14);
            write(nurz,' ',p1:1,' ',p2:2,' ',p4[i].p8:3,' ');
            for j:=1 to 3 do
              write(nurz,' ',p4[i].p9[j]:3,' ');
            writeln(nurz);
          End;
        End;
      End;
    lr:=lr+1;
  End; { zn4='2' }

if zn4='3' then
  Begin
    for i:=1 to 7 do
      Begin
        if lzam1=p5[i].p8 then
          Begin
            writeln(nurz,nagl14);
            write(nurz,' ',p1:1,' ',p2:2,' ',p5[i].p8:3,' ');
            for j:=1 to 3 do
              write(nurz,' ',p5[i].p9[j]:3,' ');
            writeln(nurz);
          End;
        End;
      End;
    lr:=lr+1;
  End; { zn4='3' }

End;
End;
End; { zn3='4' }

End; { with }

if (upcase(turz)='M') and (lr>5) and (zn3='4') then
  Begin
    writeln;
    write('K-Kontynuacja R-Rezygnacja ');
    repeat
      CzytZn;
    until upcase(zn) in ['K','R'];
    if upcase(zn)='K' then
      Begin
        clrscr;
        lr:=0;

```

```
End
else
  kpzb:=true;
End;

End; { while }

(upcase(turz)='M') and (upcase(zn)<>'R') then
Begin
  if zn4='1' then
    Begin
      write(nurz,#32);
      ZnPoziom(nurz,74);
      writeln(nurz,#32);
      writeln;
    End
  else
    Begin
      write(nurz,#32);
      ZnPoziom(nurz,50);
      writeln(nurz,#32);
      writeln;
    End;
  write('Koniec zbioru. Nacisnij klawisz Enter');
  repeat
    zn:=readkey;
  until zn=#13;
End;

upcase(turz)='D' then
Begin
  if zn4='1' then
    Begin
      write(nurz,#32);
      ZnPoziom(nurz,74);
      writeln(nurz);
    End
  else
    Begin
      write(nurz,#32);
      ZnPoziom(nurz,50);
      writeln(nurz);
    End;
  End;
ndow(1,1,80,25);
d;

PROCEDURE WyprZbioru;
Begin
  BadDosZb;
  if jestzb then
    Begin
      clrscr;
      gotoxy(2,1);
      write('Wyprowadzanie danych ze zbioru ORIENTACYJNA WARTOSC POTRZEBNEJ');
      gotoxy(2,2);
      write('OGNIOWEJ (POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');
    End;
  End;
End;
```

```
gotoxy(10,4);
write('1.Wszystkie samoloty. ');
gotoxy(10,5);
write('2.Wybrany obiekt uderzenia. ');
gotoxy(10,6);
write('3.Wybrany samolot. ');
gotoxy(10,7);
write('4.Wybrany samolot z podanym wariantem uzbrojenia. ');
gotoxy(10,9);
write('Wybierz numer: ');
repeat
  CzytZn;
until zn in ['1'..'4'];
zn3:=zn;

if zn3='2' then
  Begin
    gotoxy(10,11);
    write('Podaj kod obiektu uderzenia i naciśnij klawisz Enter : ');
    WczytKlucz(2);
  End;

if zn3='3' then
  Begin
    gotoxy(10,11);
    write('Podaj kod typu samolotu i naciśnij klawisz Enter : ');
    WczytKlucz(1);
  End;

if zn3='4' then
  Begin
    gotoxy(10,11);
    write('Podaj kod typu samolotu i naciśnij klawisz Enter : ');
    WczytKlucz(1);
    val(copy(bufkl,1,pozkl),lzan,kbl);
    gotoxy(10,12);
    write('Podaj kod wariantu uzbrojenia i naciśnij klawisz Enter : ');
    WczytKlucz(3);
    val(copy(bufkl,1,pozkl),lzanl,kbl1);
  End;

ruzbr:=false;
while not ruzbr do
  Begin
    clrscr;
    gotoxy(20,1);
    write('Rodzaje uzbrojenia. ');
    gotoxy(20,wherey+2);
    write('1.Uzbrojenie bombardierskie. ');
    gotoxy(20,wherey+1);
    write('2.Kierowane pociski rakietowe. ');
    gotoxy(20,wherey+1);
    write('3.Niekierowane pociski rakietowe. ');
    gotoxy(20,wherey+1);
    write(' i uzbrojenie artyleryjskie. ');
    gotoxy(20,wherey+1);
    write('4.Wyjście. ');
    gotoxy(20,wherey+2);
    write('Wybierz numer: ');
```

```
repeat
  CzytZn;
until zn in ['1'..'4'];

zn4:=zn;
if zn4 in ['1'..'3'] then
  Begin
    gotoxy(10,15);
    write('Podaj urządzenie wyprowadzania : [D/M] ');
    repeat
      CzytZn;
    until upcase(zn) in ['D','M'];
    zn2:=zn;
    seek(zpk1b,0);
    if upcase(zn2)='M' then
      Begin
        clrscr;
        WyprDanych(Output,zn2);
      End
    else
      Begin
        gotoxy(10,17);
        write('Włącz drukarkę i naciśnij Enter');
        repeat
          zn:=readkey;
        until zn=#13;
        WyprDanych(Lst,zn2);
      End;
    End
  End;
else
  ruzbr:=true;
End;
close(zpk1b);
End;
End;
```

PROCEDURE Sort1;

Var

```
  i,j,ilrek:integer;
Begin
  ilrek:=FileSize(zpk1b)-1;
  for i:=1 to ilrek do
    for j:=ilrek downto i do
      Begin
        seek(zpk1b,j);
        read(zpk1b,zrk1b);
        seek(zpk1b,j-1);
        read(zpk1b,zrk1b1);
        if zrk1b1.pl>zrk1b.pl then
          Begin
            seek(zpk1b,j-1);
            write(zpk1b,zrk1b);
            write(zpk1b,zrk1b1);
          End;
        End;
      End;
  End;
End;
```

```
PROCEDURE Sortowanie;
```

```
begin  
  BadDosZb;  
  if jestzb then  
    Begin  
      clrscr;  
      writeln('Sortowanie rekordow w zbiorze ORIENTACYJNA WARTOSC POTRZEBNEJ');  
      writeln('OGNIOWEJ(POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');  
      writeln('odbywa sie wedlug kodu typu samolotu');  
      writeln;  
      write('':15,'Nacisnij klawisz Enter');  
      repeat  
        zn:=readkey;  
      until zn=#13;  
      Sort1;  
      close(zpk1b);  
    End;  
end;
```

```
PROCEDURE WczytHaslo;
```

```
var  
  whaslo:boolean;  
  
begin  
  whaslo:=false;  
  pozk:=1;textbackground(white);  
  textcolor(black);  
  lowvideo;  
  FillChar(bhas,dlh,' ');  
  write(bhas);  
  gotoxy(wherex-dlh,wherey);  
  while not whaslo do  
    Begin  
      zn:=readkey;  
      case zn of  
  
        #13 : whaslo:=true;  
        #0  : zn:=readkey;  
      else  
        Begin  
          if (zn>=' ') and (zn<='~') then  
            Begin  
              write(zn);  
              bhas[pozk]:=zn;  
              pozk:=pozk+1;  
            End;  
          End;  
        end;  
      if pozk>dlh then  
        Begin  
          gotoxy(wherex-1,wherey);  
          pozk:=pozk-1;  
        End;  
      End;  
  textbackground(black);  
  textcolor(white);  
  lowvideo;  
end;
```



```
begin
  clrscr;
  gotoxy(20,5);
  write('ORIENTACYJNA WARTOSC POTRZEBNEJ OGNIOWEJ');
  gotoxy(20,6);
  write('(POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');
  gotoxy(22,8);
  write('1.Wprowadzanie');
  gotoxy(22,9);
  write('2.Przeglądanie');
  gotoxy(22,10);
  write('3.Aktualizacja');
  gotoxy(22,11);
  write('4.Usuwanie');
  gotoxy(22,12);
  write('5.Wyprowadzanie');
  gotoxy(22,13);
  write('6.Sortowanie');
  gotoxy(22,14);
  write('7.Wyjście');
  gotoxy(22,16);
  write('Wybierz numer: ');

  repeat
    CzytZn;
  until zn in ['1'..'7'];

  wyb:=ord(zn)-48;

  case wyb of
    1:WproRekDanych;
    2:PrzeGlZbioru;
    3:AktRekDanych;
    4:UsuwRekDanych;
    5:WyprZbioru;
    6:Sortowanie;
    7:konpr:=true;

  End; (case)
end;

PROCEDURE DostepOgr;
begin
  clrscr;
  gotoxy(20,5);
  write('ORIENTACYJNA WARTOSC POTRZEBNEJ OGNIOWEJ');
  gotoxy(20,6);
  write('(POLIGONOWEJ) LICZBY SAMOLOTOW MYSLIWSKO-BOMBOWYCH');
  gotoxy(22,8);
  write('1.Przeglądanie');
  gotoxy(22,9);
  write('2.Wyprowadzanie');
  gotoxy(22,10);
  write('3.Sortowanie');
  gotoxy(22,11);
  write('4.Wyjście');
  gotoxy(22,13);
  write('Wybierz numer: ');
```

```
repeat  
  CzytZn;  
until zn in ['1'..'4'];
```

```
wyb:=ord(zn)-48;
```

```
case wyb of  
  1:PrzeGlZbioru;  
  2:WyprZbioru;  
  3:Sortowanie;  
  4:konpr:=true;
```

```
End; {case}
```

```
;
```

```
P O C Z A T E K      P R O G R A M U 3
```

```
GIN
```

```
TextMode(2);  
textbackground(black);  
textcolor(white);  
lowvideo;  
konpr:=false;  
Haslo;  
while not konpr do  
  Begin  
    window(1,1,80,25);  
    textbackground(black);  
    textcolor(white);  
    lowvideo;
```

```
    if whasi then  
      DostepOgr  
    else  
      DostepPelny;
```

```
  End; {while}
```

```
0.
```