


```

MenuItem('      KOLOR GRACZY',2,8,2, '');
SubMenu(16,6,1,Vertical,Frame1,Color2, '');
  MenuItem('      BIALE - CZARNE      ',1,6,21, '');
  MenuItem('      CZARNE - BIALE',2,6,22, '');
  PopSublevel;
MenuItem('      KTO ZACZYNA',3,12,3, '');
SubMenu(16,7,1,Vertical,Frame1,Color2, '');
  MenuItem('      BIALE      ',1,6,31, '');
  MenuItem('      CZARNE',2,6,32, '');
  PopSublevel;
MenuItem('      POZIOM GRY',4,8,4, '');
SubMenu(16,8,1,Vertical,Frame1,Color2, '');
  MenuItem('      NISKI      ',1,6,41, '');
  MenuItem('      SREDNI',2,6,42, '');
  MenuItem('      DOBRY',3,6,43, '');
  MenuItem('      WYSOKI',4,6,44, '');
  MenuItem('      SUPER',5,6,45, '');
  PopSublevel;
MenuItem('      PODPOWIEDZ',5,9,5, '');
SubMenu(16,9,1,Vertical,Frame1,Color2, '');
  MenuItem('      TAK      ',1,6,51, '');
  MenuItem('      NIE',2,6,52, '');
  PopSublevel;
MenuItem('      GRAMY',6,8,6, '');
MenuItem('      KONIEC',7,10,7, '');
PopSublevel;

ResetMenu(M);
end;

type tab=array[0..9,0..9] of shortint;
  tab1=array[1..8,1..8] of shortint;
  buf=array[0..32,1..3] of integer;
const a='@';
  rg1=' KOMPUPER - KOMPUPER';
  rg2=' KOMPUPER - GRACZ  ';
  rg3=' GRACZ      - KOMPUPER';
  rg4=' GRACZ      - GRACZ  ';
  kg1=' BIALE      - CZARNE';
  kg2=' CZARNE     - BIALE  ';
  zg1=' BIALE     ';
  zg2=' CZARNE   ';
  pg1=' NISKI    ';
  pg2=' SREDNI  ';
  pg3=' DOBRY   ';
  pg4=' WYSOKI  ';
  pg5=' SUPER   ';
  og1=' T A K      ';
  og2=' N I E      ';

const  ro=60;
        br=25;
        bb=30;
        kr=-50;
        kr1=25;
        rs=15;
        sr=8;
        po=-15;
const punkt:tab1=((ro,kr1,br,br,br,br,kr1,ro),
                  (kr1,kr,po,po,po,po,kr,kr1),
                  (br,po,rs,sr,sr,rs,po,br),

```

```

        (br,po,sr,sr,sr,sr,po,br),
        (br,po,sr,sr,sr,sr,po,br),
        (br,po,rs,sr,sr,rs,po,br),
        (kr1,kr,po,po,po,po,kr,kr1),
        (ro,kr1,br,br,br,br,kr1,ro));

    tablica:tab=((0,0,0,0,0,0,0,0,0,0),
                (0,0,0,0,0,0,0,0,0,0),
                (0,0,0,0,0,0,0,0,0,0),
                (0,0,0,2,2,2,2,0,0,0),
                (0,0,0,2,1,-1,2,0,0,0),
                (0,0,0,2,-1,1,2,0,0,0),
                (0,0,0,2,2,2,2,0,0,0),
                (0,0,0,0,0,0,0,0,0,0),
                (0,0,0,0,0,0,0,0,0,0),
                (0,0,0,0,0,0,0,0,0,0));

    koniec:array[1..2] of boolean=(false,false);
const wac1='***** WACLAW LIBRONT *****';
    covers:pointer=nil;
var zacz_g:shortint;
    ii:byte;
    wac:string;
    punktacja:tab1;
    status:boolean;
    biale,czarne,kto_rusza,i,j:shortint;
    znak:char;
    buf_p:buf;
    bufor:array[0..3000,1..3] of integer;
    kursor_x,kursor_y:integer;
    wydr:boolean;
    mysl,poz:byte;
    okno_obl:boolean;
    menu_g,rodzaj_g,kolor_g,zaczyna_g,poziom_g,okno_g:byte;
    r_g,k_g,z_g,p_g,o_g:string;

procedure zapamietaj_ekran;
var a:boolean;
begin
    a:=savewindow(1,1,80,25,true,covers)
end;

procedure odtworz_ekran;
begin
    restorewindow(1,1,80,25,true,covers)
end;

procedure cursor (status:boolean);
begin
    case status of
        true:normalcursor;
        false:hiddencursor
    end
end;

procedure beep(ile:byte);
var i:byte;
begin
    for i:=1 to ile do
        begin

```

```

        sound(300);delay(100);
        nosound;delay(100)
    end
end;

procedure rysuj_pole(x,y,kolor:shortint);
var i,j,k,l,att,ile:shortint;
begin
    ile:=5;
    case kolor of
        1:att:=16+15;
        2,0:att:=16+12;
        -1:att:=16+0;
        5:att:=16+1;
        3:att:=16+1;
        4:att:=16+1
    end;
    k:=15+x*ile;
    l:=3+y*2;
    tablica[x,y]:=kolor;
    case kolor of
        2,-1,0,1,4:begin
            sound(100);
            putchar(k ,l,32,att);
            putchar(k+1,l,220,att);
            putchar(k+2,l,220,att);
            putchar(k+3,l,220,att);
            putchar(k+4,l,220,att);
            putchar(k ,l+1,32,att);
            putchar(k+1,l+1,219,att);
            putchar(k+2,l+1,219,att);
            putchar(k+3,l+1,219,att);
            putchar(k+4,l+1,219,att);
            delay(10);
            nosound;
        end;
        5:begin
            putchar(k,l,220,att);
            putchar(k+1,l,220,att);
            putchar(k+2,l,220,att);
            putchar(k+3,l,220,att);
            putchar(k+4,l,220,att);
        end;
        3:begin
            putchar(k,l,219,att);
            putchar(k,l+1,219,att);
        end
    end
end;

end;

procedure wylicz_pola;
var i,j,b,c:byte;
begin
    b:=0;c:=0;
    for i:=1 to 8 do
        for j:=1 to 8 do
            case tablica[i,j] of
                1:b:=b+1;
                -1:c:=c+1;
                2,0:begin end
            end;
        end;
    end;
end;

```

```

biale:=b;czarne:=c;
for i:=1 to 80 do putchar(i,1,32,80);
for i:=1 to 80 do putchar(i,2,32,80);
textattr:=80+15;
gotoxy(1,1);write('BIALE ',biale);
textattr:=80+0;
gotoxy(1,2);write('CZARNE ',czarne);
for i:=1 to biale do putchar(i+10,1,254,15+80);
for i:=1 to czarne do putchar(i+10,2,254,80)
end;

procedure sprawdz_pole(var nr_pola:shortint;var koniec:boolean;
                      x,y,kolor:shortint);
begin
  if (x<1) or (x>8) or (y<1) or (y>8) then nr_pola:=0;
  if (tablica[x,y]=0) or (tablica[x,y]=2) then nr_pola:=0;
  if tablica[x,y]=kolor then koniec:=true;
end;

procedure nowe_pole(kierunek,x,y,nr_pola:byte;var x1,y1:shortint);
begin
  case kierunek of
    1:begin x1:=x+nr_pola;y1:=y+nr_pola end;
    2:begin x1:=x+nr_pola;y1:=y          end;
    3:begin x1:=x+nr_pola;y1:=y-nr_pola end;
    4:begin x1:=x;          y1:=y-nr_pola end;
    5:begin x1:=x-nr_pola;y1:=y-nr_pola end;
    6:begin x1:=x-nr_pola;y1:=y          end;
    7:begin x1:=x-nr_pola;y1:=y+nr_pola end;
    8:begin x1:=x;          y1:=y+nr_pola end;
  end
end;

procedure nowe_pole_rek(kierunek,x,y,nr_pola,kolor:shortint);
var x1,y1:shortint;
begin
  nowe_pole(kierunek,x,y,nr_pola,x1,y1);
  case tablica[x1,y1] of
    2,0:buf_p[31,3]:=0;
    1:if kolor=1 then buf_p[31,3]:=1
      else begin nr_pola:=nr_pola+1;
                nowe_pole_rek(kierunek,x,y,nr_pola,kolor);
                if buf_p[31,3]=1 then rysuj_pole(x1,y1,kolor)
              end;
    -1:if kolor=-1 then buf_p[31,3]:=1
      else begin nr_pola:=nr_pola+1;
                nowe_pole_rek(kierunek,x,y,nr_pola,kolor);
                if buf_p[31,3]=1 then rysuj_pole(x1,y1,kolor)
              end
  end
  end
end;

procedure rysuj_wokol(x,y,kolor:shortint);
var kierunek,pole:shortint;

begin
  for kierunek:=1 to 8 do
    begin
      buf_p[31,3]:=0;
      pole:=1;

```

```

        nowe_pole_rek(kierunek,x,y,pole,kolor)
    end
end;
var ruszaj:boolean;

procedure badaj_wokol(kto_rusza,x,y:shortint;var punkty:integer);
var kierunek,nr_pola,x1,y1:shortint;
    koniec:boolean;

begin
    ruszaj:=false;
    for kierunek:=1 to 8 do
        begin
            koniec:=false;
            nr_pola:=0;
            x1:=0;y1:=0;
            repeat
                nr_pola:=nr_pola+1;
                nowe_pole(kierunek,x,y,nr_pola,x1,y1);
                if (x1<1) or (x1>8) or (y1<1) or (y1>8)
                    then nr_pola:=0
                    else sprawdz_pole(nr_pola,koniec,x1,y1,kto_rusza)
                until (nr_pola=0) or (koniec=true);
                if nr_pola>1 then begin punkty:=punkty+nr_pola;ruszaj:=true end;
            end;
            if ruszaj=true then punkty:=punkty+punktacja[x,y]
        end;

procedure wydr4(b_p:buf);
var i:byte;
begin
    for i:=1 to b_p[0,3]-1 do
        if i<20 then
            begin gotoxy(3,4+i);write('x ',b_p[i,1],' y ',b_p[i,2]) end;
    end;

procedure badanie(kto_rusza:shortint);
var x,y:byte;
    punkty:integer;
begin
    buf_p[0,3]:=1;
    for x:=1 to 8 do
        for y:=1 to 8 do
            if tablica[x,y]=2 then
                begin
                    punkty:=0;
                    badaj_wokol(kto_rusza,x,y,punkty);
                    if ruszaj=true then
                        begin
                            buf_p[buf_p[0,3],3]:=punkty;
                            buf_p[buf_p[0,3],1]:=x;
                            buf_p[buf_p[0,3],2]:=y;
                            buf_p[0,3]:=buf_p[0,3]+1
                        end
                    end
                end;

procedure rysuj_pole1(x,y:byte;kolor:shortint);
var i:byte;
    kol:shortint;
begin

```

```

rysuj_pole(x,y,kolor);
beep(1);kol:=kolor;
for i:=1 to 30 do
  begin
    rysuj_pole(x,y,kol);
    if kol=0 then kol:=kolor else kol :=0
  end;
rysuj_pole(x,y,kolor)
end;

procedure wybierz_ruch;
begin
  buf_p[32,3]:=-1000;
  for i:=1 to buf_p[0,3]-1 do
    if buf_p[32,3]<buf_p[i,3] then buf_p[32]:=buf_p[i];
  rysuj_pole1(buf_p[32,1],buf_p[32,2],kto_rusza);
  delay(700);
  rysuj_wokol(buf_p[32,1],buf_p[32,2],kto_rusza)
end;

procedure wydr3(x,y:byte);
  var znak:char;
  begin
    gotoxy(3,8+poz);write(' ');
    gotoxy(3,8+poz);write('x',x,' y',y,' w',bufor[3000,3]);
    if wydr=true then znak:=readkey;
  end;

procedure wybierz_ruch1;
var i,a,b:integer;
  znak:char;
begin
  {if kto_rusza=1 then bufor[3000,3]:=-1000 else bufor[3000,3]:=1000;}
  bufor[3000,3]:=-1000;
  for i:=1 to bufor[0,3]-1 do
    if bufor[3000,3]<bufor[i,3] then bufor[3000]:=bufor[i];
    rysuj_pole1(bufor[3000,1],bufor[3000,2],kto_rusza);
    delay(700);
    rysuj_wokol(bufor[3000,1],bufor[3000,2],kto_rusza)
end;

procedure wybierz_ruch2;
var x,y,j:byte;
  suma,i:integer;
begin
  i:=1;suma:=0;
  bufor[3000,3]:=-1000;
  repeat
    x:=bufor[i,1];y:=bufor[i,2];j:=0;
    while (bufor[i,1]=x) and (bufor[i,2]=y) do
      begin suma:=suma+bufor[i,3]; i:=i+1 end;
    if bufor[3000,3] < suma then
      begin
        bufor[3000,3]:=suma;
        bufor[3000,1]:=x;
        bufor[3000,2]:=y;
        if okno_g=1 then wydr3(x,y)

```

```

        end;
        suma:=0;
until i>=bufor[0,3];
rysuj_pole1(bufor[3000,1],bufor[3000,2],kto_rusza);
delay(500);
rysuj_wokol(bufor[3000,1],bufor[3000,2],kto_rusza)
end;

```

```

procedure wybierz_ruch3;
var x,y,j:byte;
    suma,i:integer;

begin
    i:=1;
    bufor[3000,3]:=-1000;
    suma:=0;
    repeat
        x:=bufor[i,1];y:=bufor[i,2];j:=0;
        while (bufor[i,1]=x) and (bufor[i,2]=y) do
            begin suma:=suma+bufor[i,3]; i:=i+1;j:=j+1 end;
        if bufor[3000,3] < (suma div j) then
            begin
                bufor[3000,3]:=suma div j;
                bufor[3000,1]:=x;
                bufor[3000,2]:=y;
                {if okno_g=1 then wydr3(x,y);}
            end;
        suma:=0;
    until i>=bufor[0,3];
    rysuj_pole1(bufor[3000,1],bufor[3000,2],kto_rusza);
    delay(500);
    rysuj_wokol(bufor[3000,1],bufor[3000,2],kto_rusza)
end;

```

```

procedure uaktualnij_tablice(x,y:byte);
var x1,y1,kier:shortint;
begin
    for kier:=1 to 8 do
        begin
            x1:=0;y1:=0;
            nowe_pole(kier,x,y,1,x1,y1);
            if tablica[x1,y1]=0 then tablica[x1,y1]:=2
        end
    end
end;

```

```
end;
```

```

procedure rekurencja(poziom,kolor,poprz_wynik,x,y:integer;tablica:tab);
var buf_p:buf;
    nowa_tablica:tab;
    i,wynik:integer;

```

```

    procedure nowe_pole(kierunek,x,y,nr_pola:byte;var x1,y1:shortint);
    begin
        case kierunek of
            1:begin x1:=x+nr_pola;y1:=y+nr_pola end;
            2:begin x1:=x+nr_pola;y1:=y          end;
            3:begin x1:=x+nr_pola;y1:=y-nr_pola end;
            4:begin x1:=x;          y1:=y-nr_pola end;

```



```

5:begin x1:=x-nr_pola;y1:=y-nr_pola end;
6:begin x1:=x-nr_pola;y1:=y end;
7:begin x1:=x-nr_pola;y1:=y+nr_pola end;
8:begin x1:=x; y1:=y+nr_pola end;
end
end;

procedure nowa_t(x,y,kolor:shortint;var nowa_tablica:tab);
var x1,y1,kier,pole:shortint;
procedure nowe_pole_rek1(kierunek,x,y,nr_pola,kolor:shortint);
var x1,y1:shortint;
begin
nowe_pole(kierunek,x,y,nr_pola,x1,y1);
case tablica[x1,y1] of
2,0:buf_p[31,3]:=0;
-1,1:if kolor=tablica[x1,y1] then buf_p[31,3]:=1
else begin nr_pola:=nr_pola+1;
nowe_pole_rek1(kierunek,x,y,nr_pola,kolor);
if buf_p[31,3]=1 then nowa_tablica[x1,y1]:=kolor
end
end
end;
end;

begin
nowa_tablica:=tablica;
nowa_tablica[x,y]:=kolor;
for kier:=1 to 8 do
begin
buf_p[31,3]:=0;
pole:=1;
nowe_pole_rek1(kier,x,y,pole,kolor)
end;
for kier:=1 to 8 do
begin
x1:=0;y1:=0;
nowe_pole(kier,x,y,1,x1,y1);
if nowa_tablica[x1,y1]=0 then nowa_tablica[x1,y1]:=2
end
end;
end;

procedure sprawdz_pole(var nr_pola:shortint;var koniec:boolean;
x,y,kolor:shortint);
begin
if (tablica[x,y]=0) or (tablica[x,y]=2) then nr_pola:=0;
if tablica[x,y]=kolor then koniec:=true
end;

var zapisz:boolean;

procedure badaj_wokol(kto_rusza,x,y:shortint;var punkty:integer);
var kierunek,nr_pola,x1,y1:shortint;
koniec:boolean;

begin
for kierunek:=1 to 8 do
begin
koniec:=false;
nr_pola:=0;
x1:=0;y1:=0;

```

```

repeat
  nr_pola:=nr_pola+1;
  nowe_pole(kierunek,x,y,nr_pola,x1,y1);
  if (x1<1) or (x1>8) or (y1<1) or (y1>8)
    then nr_pola:=0
    else sprawdz_pole(nr_pola,koniec,x1,y1,kto_rusza);
until (nr_pola=0) or (koniec=true);
if nr_pola>1 then
  begin
    zapisz:=true;
    punkty:=punkty+(nr_pola-1)*kto_rusza;
    case kierunek of
      6,2:if (y=1) or (y=8)
          then punkty:=punkty+bb;
      8,4:if (x=1) or (x=8)
          then punkty:=punkty+bb
    end
  end
end;
if zapisz=true then punkty:=punkty+punktacja[x,y]
end;

procedure badanie(kto_rusza:shortint);
var x,y,punkty:integer;

begin
  buf_p[0,3]:=1;
  for x:=1 to 8 do
    for y:=1 to 8 do
      if tablica[x,y]=2 then
        begin
          zapisz:=false;
          punkty:=0;
          badaj_wokol(kto_rusza,x,y,punkty);
          if zapisz=true then
            begin
              buf_p[buf_p[0,3],3]:=punkty;
              buf_p[buf_p[0,3],1]:=x;
              buf_p[buf_p[0,3],2]:=y;
              buf_p[0,3]:=buf_p[0,3]+1
            end
          end
        end
      end;

procedure wydr1;
var znak:char;
begin
  gotoxy(3,5+poziom);write(' ');
  gotoxy(3,5+poziom);write('x',buf_p[i,1], ' y',buf_p[i,2], ' p',buf_p[i,3], '
w',wynik);
  if wydr=true then znak:=readkey;
end;

procedure wydr2;
var znak:char;
begin
  gotoxy(3,7+poz);write(' ');
  gotoxy(3,7+poz);write(bufor[0,3], ' x',x, ' y',y, ' ',wynik);
  if wydr=true then znak:=readkey;
end;

```

```

procedure wybierzbuf_p;
var i:byte;
begin
  for i:=1 to buf_p[0,3]-1 do
    if buf_p[i,3]>buf_p[1,3]
      then buf_p[1]:=buf_p[i];
  buf_p[0,3]:=2
end;

procedure wybierzbuf_pl;
var i:byte;
begin
  buf_p[32,3]:=1;
  for i:=1 to buf_p[0,3]-1 do
    if buf_p[i,3]>buf_p[1,3]
      then begin buf_p[1]:=buf_p[i];buf_p[32,3]:=2 end
    else
      if buf_p[i,3]=buf_p[1,3]
        then begin
          buf_p[buf_p[32,3]]:=buf_p[i];
          buf_p[32,3]:=buf_p[32,3]+1
        end;
  buf_p[0,3]:=buf_p[32,3]
end;

begin
  buf_p[0,3]:=1;
  badanie(kolor);
  if poziom>1 then wybierzbuf_pl;
  for i:=1 to buf_p[0,3]-1 do
    begin
      wynik:=poprz_wynik+kolor*buf_p[i,3];
      if okno_g=1 then wydr1;
      if poziom=1 then begin x:=buf_p[i,1];y:=buf_p[i,2] end;
      if poziom<poz then
        begin
          nowa_t(buf_p[i,1],buf_p[i,2],kolor,nowa_tablica);
          rekurencja(poziom+1,kolor*(-1),wynik,x,y,nowa_tablica)
        end;
      if (poziom=poz) and (bufor[0,3]<3000) then
        begin
          bufor[bufor[0,3],3]:=wynik*kto_rusza;
          bufor[bufor[0,3],1]:=x;
          bufor[bufor[0,3],2]:=y;
          bufor[0,3]:=bufor[0,3]+1;
          mysl:=mysl+5;sound(mysl);
          if okno_g=1 then wydr2
        end
    end
end;

procedure wskaznik(var x,y:byte);
var znak:char;
  ch,att,xx,yy:byte;
const kon:boolean=false;
begin
  if okno_g=1 then
    begin
      zapamietaj_ekran;
      framewindow(1,4,13,24,12,12,'');
      textattr:=13;

```

```

        wydr4(buf_p);
    end;
kon:=false;
textcolor(15);
gotoxy(kursor_x,kursor_y);
getchar(wherex,wherey,ch,att);
putchar(wherex,wherey,219,11);
repeat
    while keypressed do znak:=readkey;
    znak:=readkey;
    if znak=#0 then
        begin
            znak:=readkey;
            xx:=wherex;yy:=wherey;
            case znak of
{G}    #72:gotoxy(wherex,wherey-1) ;
{D}    #80:gotoxy(wherex,wherey+1) ;
{P}    #77:gotoxy(wherex+1,wherey) ;
{L}    #75:gotoxy(wherex-1,wherey)
            end;
            putchar(xx,yy,ch,att);
            getchar(wherex,wherey,ch,att);
            putchar(wherex,wherey,219,11);
        end
    else
        case znak of
            #27,#13,#32:begin
                case wherex of
                    20..24:x:=1;
                    25..29:x:=2;
                    30..34:x:=3;
                    35..39:x:=4;
                    40..44:x:=5;
                    45..49:x:=6;
                    50..54:x:=7;
                    55..59:x:=8;
                    69..80:x:=9;
                    else x:=0
                end;
                case wherey of
                    5..6 :y:=1;
                    7..8 :y:=2;
                    9..10:y:=3;
                    11..12:y:=4;
                    13..14:y:=5;
                    15..16:y:=6;
                    17..18:y:=7;
                    19..20:y:=8;
                    20..25:y:=9;
                    else y:=0
                end
            end
        end;
        if (x in [1..9]) and (y in [1..9]) then kon:=true else kon:=false;
    until kon=true;
    if okno_g=1 then odtworz_ekran;
    textcolor(15);
    kursor_x:=wherex;kursor_y:=wherey;
end;

procedure zmiany_punktacji(x,y:byte);

```

```

var i,j:byte;
begin
  if biale+czarne<20 then poz:=3 else poz:=poziom_g;
  if biale+czarne=58 then
    for i:=1 to 8 do for j:=1 to 8 do punktacja[i,j]:=1;
  if punktacja[x,y]=ro then
    begin
      case x of
        1:case y of
          1:begin
              punktacja[2,1]:=br;
              punktacja[2,3]:=br;
              punktacja[1,2]:=br
            end;
          8:begin
              punktacja[1,7]:=br;
              punktacja[2,7]:=br;
              punktacja[2,8]:=br
            end
        end;
      8:case y of
        1:begin
              punktacja[7,1]:=br;
              punktacja[7,2]:=br;
              punktacja[8,2]:=br
            end;
          8:begin
              punktacja[8,7]:=br;
              punktacja[7,7]:=br;
              punktacja[7,8]:=br
            end
        end
      end
    end
  end
end;

```

```

procedure ruch_komputera(kto_rusza:shortint);
var por:byte;
begin
  gotoxy(30,24);
  textattr:=10;
  case kto_rusza of
    1:write('***** B I A L E *****');
    -1:write('***** C Z A R N E *****')
  end;
  if okno_g=1 then
    begin
      zapamietaj_ekran;
      framewindow(1,5,20,9+poz,12,12,'');
      textattr:=13;
    end;
  bufor[0,3]:=1;
  mysl:=30;
  sound(mysl);
  rekurencja(1,kto_rusza,0,0,0,tablica);
  nosound;
  if okno_g=1 then odtworz_ekran;
  if bufor[0,3]>1
  then
    begin

```

```

        wybierz_ruch3;
        uaktualnij_tablice(bufor[3000,1],bufor[3000,2]);
        zmiany_punktacji(bufor[3000,1],bufor[3000,2]);
        koniec[1]:=false
    end
else
    begin
        buf_p[0,3]:=0;
        badanie(kto_rusza);
        if buf_p[0,3]>1
            then
                begin
                    wybierz_ruch;
                    uaktualnij_tablice(buf_p[32,1],buf_p[32,2]);
                    zmiany_punktacji(buf_p[32,1],buf_p[32,2]);
                    koniec[1]:=false
                end
            else
                begin
                    if koniec[1]=true then koniec[2]:=true;
                    koniec[1]:=true;
                    gotoxy(20,24);
                    write('BRAK RUCHU  RUSZA  PRZECIWNIK          ');
                    beep(10);
                    gotoxy(20,24);
                    write('          ')
                end
            end;
        wylicz_pola
    end;
end;

```

```

procedure ruch_gracza(kto_rusza:shortint);
var x,y:byte;
    punkty:integer;
    p:shortint;
    zn:char;
label powtorz,zle;
begin
    gotoxy(30,24);
    textattr:=10;
    case kto_rusza of
        1:write('***** B I A L E *****');
        -1:write('***** C Z A R N E *****')
    end;
    buf_p[0,3]:=0;
    badanie(kto_rusza);
    if buf_p[0,3]>1
        then
            begin
powtorz:x:=0;y:=0;
                wskaznik(x,y);
                if (x=9) and (y=9) then
                    begin
                        koniec[1]:=true;koniec[2]:=true;
                        exit
                    end;
                p:=tablica[x,y];
                rysuj_pole(x,y,kto_rusza);delay(50);
                if p<>2 then goto zle;
                punkty:=0;
            end
        end;
end;

```

```

badaj_wokol(kto_rusza,x,y,punkty);
if ruszaj=true
then
begin
rysuj_pole1(x,y,kto_rusza);
rysuj_wokol(x,y,kto_rusza);
uaktualnij_tablice(x,y);
zmiany_punktacji(x,y);
koniec[2]:=false
end
else
begin
buf_p[0,3]:=0;
badanie(kto_rusza);
if buf_p[0,3]>1
then
zle: begin
gotoxy(20,24);
write('ZLE POSTAWILES PIONA POPRAW SIE !!!');
beep(10);
rysuj_pole(x,y,p);
gotoxy(20,24);
write(' ');
goto powtorz
end
else
begin
if koniec[2]=true then koniec[1]:=true;
koniec[2]:=true;
gotoxy(20,24);
write('BRAK RUCHU RUSZA PRZECIWNIK ');
beep(10);
gotoxy(20,24);
write(' ')
end
end
end
else
begin
if koniec[2]=true then koniec[1]:=true;
koniec[2]:=true;
gotoxy(20,24);
write('BRAK RUCHU RUSZA PRZECIWNIK ');
beep(10);
gotoxy(20,24);
write(' ')
end;
wylicz_pola;
delay(1000);
end;

```

```

procedure podmenu;
begin
case long of
11:r_g:=rg1;
12:r_g:=rg2;
13:r_g:=rg3;
14:r_g:=rg4;

```

```

21:k_g:=kg1;
22:k_g:=kg2;
31:z_g:=zg1;
32:z_g:=zg2;
41:p_g:=pg1;
42:p_g:=pg2;
43:p_g:=pg3;
44:p_g:=pg4;
45:p_g:=pg5;
51:o_g:=og1;
52:o_g:=og2;
end;
end;

begin
r_g:=rg2;k_g:=kg1;z_g:=zg1;p_g:=pg3;o_g:=og2;
menu_g:=1;rodzaj_g:=2;kolor_g:=1;zaczyna_g:=1;poziom_g:=3;okno_g:=2;
initmenu(m);
REPEAT
punktacja:=punkt;
textattr:=12;
clrscr;
wac:=wac1;
{for ii :=1 to length(wac) do linia[ii+25].znak:=wac[ii];}

repeat
long:=menuchoice(m,ke);
podmenu;
until (long=6) or (long=7);
erasemenu(m,true);
if long=7 then exit;
textattr:=0+11;
clrscr;
for i:=1 to 8 do
for j:=1 to 8 do rysuj_pole(i,j,0);
for i:=0 to 9 do
begin
rysuj_pole(0,i,4);
rysuj_pole(9,i,4);
rysuj_pole(i,0,4);
rysuj_pole(i,9,4)
end;
for i:=0 to 9 do
begin
rysuj_pole(10,i,3);
rysuj_pole(i,10,5)
end;
putchar(65,23,32,16+0);
for i:=1 to 8 do
begin textcolor(15);textbackground(1);
gotoxy(17+i*5,4);write(i);
gotoxy(17+i*5,22);write(i);
gotoxy(18,4+i*2);write(i);
gotoxy(62,4+i*2);write(i)
end;

textattr:=$4f;
window(69,21,80,24);
clrscr;

```



```

window(1,1,80,25);
framewindow(69,21,80,24,$4f,$4f,'');
gotoxy(72,22);write('KONIEC');
gotoxy(73,23);write('GRY');

rysuj_pole(4,4,1);uaktualnij_tablice(4,4);
rysuj_pole(5,5,1);uaktualnij_tablice(5,5);
rysuj_pole(4,5,-1);uaktualnij_tablice(4,5);
rysuj_pole(5,4,-1);uaktualnij_tablice(5,4);
wylicz_pola;
poz:=poziom_g;
if zaczyna_g=2 then zacz_g:=-1 else zacz_g:=1;
repeat
  zacz_g:=zacz_g*(-1);
  if zacz_g=1 then kto_rusza:=-1 else kto_rusza:=1;
  case rodzaj_g of
    1:if kolor_g=1 then
      if kto_rusza=1 then ruch_komputera(kto_rusza)
        else ruch_komputera(kto_rusza)
      else
        if kto_rusza=1 then ruch_komputera(kto_rusza)
          else ruch_komputera(kto_rusza);
    2:if kolor_g=1 then
      if kto_rusza=1 then ruch_komputera(kto_rusza)
        else ruch_gracza(kto_rusza)
      else
        if kto_rusza=1 then ruch_gracza(kto_rusza)
          else ruch_komputera(kto_rusza);
    3:if kolor_g=1 then
      if kto_rusza=1 then ruch_gracza(kto_rusza)
        else ruch_komputera(kto_rusza)
      else
        if kto_rusza=1 then ruch_komputera(kto_rusza)
          else ruch_gracza(kto_rusza);
    4:if kolor_g=1 then
      if kto_rusza=1 then ruch_gracza(kto_rusza)
        else ruch_gracza(kto_rusza)
      else
        if kto_rusza=1 then ruch_gracza(kto_rusza)
          else ruch_gracza(kto_rusza)
  end;
until (koniec[1]=true) and (koniec[2]=true);
while keypressed do znak:=readkey;
textattr:=13;
gotoxy(20,24);write('KONIEC GRY - NACISNIJ SPACJE');
repeat
  while keypressed do znak:=readkey;
  znak:=readkey;
until znak=#32;
UNTIL menu_g=7
end.

{*****}

unit get_put;

interface
uses tpcrt,dos;
procedure getchar(x,y:byte;var cha,att:byte);
procedure putchar(x,y,cha,att:byte);

```

```

implementation
procedure getchar(x,y:byte;var cha,att:byte);
  var
    x1,y1:byte;
    reg:registers;
  begin
    x1:=wherex;
    y1:=wherey;
    gotoxy(x,y);
    with reg do
      begin
        AH:=$08;
        BH:=0
      end;
    intr($10,reg);
    with reg do
      begin
        cha:=AL;
        att:=AH
      end;
    gotoxy(x1,y1);
  end;

procedure putchar(x,y,cha,att:byte);
  var
    x1,y1:byte;
    reg:registers;
  begin
    x1:=wherex;
    y1:=wherey;
    gotoxy(x,y);
    with reg do
      begin
        AH:=$09;
        AL:=cha;
        BH:=0;
        BL:=att;
        CX:=1
      end;
    intr($10,reg);
    gotoxy(x1,y1);
  end;
end.

```