

```

x,clear,dambræt<
a,n<
begin comment damspillende program;
  integer stacktop,ran,stp,oldtop,stinc,bufant,byte,maxniv,niveau,state,
  link,point,mi,t,i,ii,q,k,ant,alt,j,db,højde,bogst,goodnes,looki;
  integer array b[1:32,1:6];
  boolean newmove,quin,color,m,v,dm,di,slut,valg,look;
  boolean array buf,draw,a[0:10],stack[0:1000],hb[0:10,1:40];

```

```

  comment
b[1:32,1:6]

```

```

h g f e d c b a

```

```

32 31 30 29 1
 28 27 26 25 2
24 23 22 21 3
 20 19 18 17 4
16 15 14 13 5
 12 11 10 9 6
 8 7 6 5 7
 4 3 2 1 8

```

```

h h h h
 h h h h h
h h h h
 . . . .
 . . . .
 s s s s s
s s s s
 s s s s

```

```

;

```

```

procedure tostack;
begin integer i;
  buf[bufant]:=newmove;
  for i:=1 step 1 until bufant do
  stack[stp+i-1]:=buf[i];
  stack[stp+bufant]:=deadpos;
  stack[stp+bufant+1]:=buf[0];

```

```

    stack[stp+bufant+2]:=boolean(bufant+3);
    stinc:=stinc+bufant+3;
    stp:=stp+bufant+3;
    alt:=alt+1
end tostack;

procedure COLOR(color);
value color; boolean color;
writetext(if color then <<hvid> else <<sort>);

integer procedure look forward(color);
value color; boolean color;
begin integer i,ii,k,t,gieri;
    boolean m,v;
    if false then begin writecr; writetext(<<look forward >); COLOR(color); end;
    bufant:=state:=1; stp:=stacktop+2;
    link:=if color then 3 else 5;
    alt:=stinc:=0;niveau:=niveau+1;

    for i:=1 step 1 until 32 do
    begin looki:=i;
        gieri:=gier(look);
        if false then begin writecr; writetext(<<i, gieri: >); write(<<ddd>,i,gieri); end;
        if gieri=2 then
            begin comment felt optaget af egen farve;
                quin:=booleanb[i,1]shift2;
                newmove:=buf[1]:=400v(booleani)v(quinshift8);
                maxniv:=bufant:=1; buf[0]:=false; byte:=10;
                move(i,false,-1,1);
                b[i,1]:=integer(quinshift-2);
            end gennemgang af trækmuligheder;
        endfor;

    if state=1 then
    begin comment spillet tabt paa dette niveau;
        newmove:=false;
        buf[0]:=boolean(-100);
        tostack
    end;
end;

```

```
stack[stacktop]:=booleanniveau;  
stack[stacktop+1]:=booleanstinc;  
oldtop:=stacktop;  
stacktop:=stacktop+stinc+4;  
stack[stacktop-2]:=false;  
stack[stacktop-1]:=boolean(oldtop-1);  
stack[oldtop-2]:=boolean(stacktop-2);  
lookforward:=stacktop-2-(integerstack[stacktop-3])  
end lookforward ;
```

```
corecode look,looki,b,color;  
3,46  
2,44  
2,48  
2,46  
arna2  
hhreLZ  
xr,mlnre2  
xr,arc42  
ara3,il  
pi(c17)  
hvnre1NOA  
arn5 12D  
ara4IOA  
hhnreLOC  
hhnreNOC  
e:arnc42,arc42  
e1:arc42,hrs1  
e2:6  
e;
```

```
procedure inmove(i,j);  
value i,j; integer i,j;  
comment brik flyttes fra felt nr.j til felt nr.i;  
begin boolean bil;  
bil:=booleanb[i,1];  
if byte>30 then
```

```

begin byte:=0; buf[bufant]:=newmove;
  bufant:=bufant+1; newmove:=false
end;
newmove:=newmove∨ (( boolean i ∨( bil shift 10)) shift byte );
byte:=byte+10;
b[i,1]:=b[j,1]; b[j,1]:=0;
if bil then
point:=(if bil shift 2 then 10 else 4) else point:=0;
point:=point+ b[i,2] - b[j,2]+(if(i>28∨i<5)^( -,buf[1]shift-8)then 10 else0);
buf[0]:= boolean ( integer buf[0]+point)
end procedure inmove;

procedure reestab(i);
value i;integer i;
comment brik flyttes i felter tilbage;
begin boolean c,old; integer i1,i2,j;
  for j:=1 step 1 until i do
begin c:= (newmove shift (10-byte))^(30010m);
  i1:= integer(c^3307m);
  old:= (c^(3003m70)) shift -10;

  b[i1,1]:= integer old;
  newmove:=newmove ^(( 30m100) shift ( byte - 10));
  byte :=byte - 10;
  if byte= 0 then
begin byte := 40;
  bufant := bufant - 1;
  newmove := boolean buf [ bufant]
end;
  i2 := integer(( newmove shift ( 10 - byte)) ^3307m);
  if old then
point := ( if old shift 2 then 10 else 4) else point := 0;
point := point + b[ i1,2] - b[ i2,2]+(if(i1>28∨i1<5)^( -,buf[1]shift -8) then10else0);
buf[0] := boolean ( integer buf[0] - point )
end for loop
end procedure reestab;

procedure move(place,take,bad,niveau);
valueplace,take,bad,niveau;
booleantake;integerplace,bad,niveau;

```

```

begin integer j,sq,sqo,gieri;
  if false then begin writecr; writetext(⟨<move⟩); writetext(if quin then ⟨< dam⟩ else ⟨< alm⟩); end;
  if -,quin then
  begin comment almindelig brik;
    for j:=link,link+1 do
    begin sq:=looki:=b[place,j];
      gieri:=gier(look);
      if false then begin writecr; writetext(⟨< looki, gieri: ⟩); write(⟨ddd⟩,looki,gieri); end;
      case gieri of
      begin

        comment frit felt;
        if -,take ^ state<2 then
        begin state:=2;
          inmove(sq,place); tostack;
          reestab(1)
        end case1;

        comment felt opt. af egen farve;;

        comment felt opt. af modstander;
        if state<3 then
        begin inmove(sq,place);
          sqo:=sq; sq:=looki:=b[sq,j];
          if gier(look)=1 then
          begin
            if state<2 then
            begin stinc:=alt:=0; stp:=stacktop+2; state:=3end;
              inmove(sq,sqo);
              move(sq,true,bad,niveau+1);
              if maxniv<niveau then maxniv:=niveau;
              if maxniv=niveau then tostack else maxniv:=maxniv-1;
              reestab(2)
            end else reestab(1)
          end case 3

        end case
      end
    end almindelig brik else
    begin comment dam;

```

```

for j:=3 step 1 until 6 do
  if bad+j then
  begin sq:=place; sq:=looki:=b[place,j];
    try:
    case gier(look) of
    begin
      begin comment frit felt;
        if -,take ^ state<2 then
        begin state:=2;
          inmove(sq,sq); tostack; reestab(1);
          sq:=sq;
          for looki:=b[sq,j] while gier(look)=1 do
            begin sq:=looki; inmove(sq,sq); tostack; reestab(1); sq:=sq end;
          sq:=looki
        end else
        begin sq:=sq;
          for looki:=b[sq,j] while gier(look)=1 do sq:=sq:=looki;
            sq:=looki
          end;
        goto try
      end case1;

      comment optaget af egen brik;;

      begin comment optaget af modstander;
        inmove(sq,sq);
        sq:=sq; sq:=looki:=b[sq,j];
        if gier(look)=1 then
        begin
          if state<3 then
          begin stinc:=alt:=0; stp:=stacktop+2; state:=4end;
            inmove(sq,sq);
            move(sq,true, (j-4) mod 4+3, niveau+1);
            if maxniv<niveau then maxniv:=niveau;
            if maxniv=niveau then tostack else maxniv:=maxniv-1;
            reestab(2)
          end else reestab(1)
        end
      end case 3
    end
  end
end

```

```

        end case
    end
end dam
end procedure move;

boolean procedure comp;
comment comp sammenligner et træk i stakken med hidtil bedste (hb)
og har værdien true, hvis der er flere træk tilbage og false ellers.
desuden reetablerer comp dambrættet;

begin integer i,ii,j,k,c,niveau,p1,p2,ant;
    boolean boo,bool;
    if false then begin writecr; writetext(⟨<comp⟩); end;
    bool:=true;
    et mere:
    i:=integerstack[1];
    ant:=integerstack[i-1]-1;
    k:=i-ant-4;
    for j:=3 step 1 until ant do
        a[j]:=stack[k+j];
        niveau:=p1:=0; boo:=true;
        for i:=i while i≠0 do
            begin comment pointsammentælling ved linkning gennem stakken;
                p1:=p1+(if boo then integer stack[i-2] else -integerstack[i-2]);
                ii:=i; i:=integerstack[i];
                boo:=-,boo;
                niveau:=niveau+1
            end pointsammentællingfor;
            a[1]:=booleanniveau;
            a[2]:=booleanp1;
            if integerhb[1,niveau]=0 then goto gem else
            begin p2:=integerhb[2,niveau];
                if (boo∧p1<p2)∨(-,boo∧p2<p1)∨(p1=p2∧random) then
                    gem:
                    begin
                        for j:=1 step 1 until ant do
                            hb[j,niveau]:=a[j];hb[j,niveau]:=false
                        end
                    end;
            end;
end;

```

```

vrkd:
ant:=integerstack[ii-1];
j:=ii-ant;
k:=ii-4;
if bool then move back(stack,j,k);
c:=oldtop-1;
bool:=c+3=j;
if bool^oldtop=3 then
begin comp:=true; goto exit end;
if -, bool then
begin stack[c-1]:=booleanj;
stack[j]:=false;
stack[j+1]:=booleanc;
stack[c+2]:=boolean(integer(stack[c+2])-ant);
stacktop:=stacktop-ant;
if stack[stacktop-5] then goto et mere
end flere træk tilbage paa dette niveau else
begin stack[c-1]:=false;
stacktop:=oldtop;
oldtop:=integerstack[c]+1;
c:=integerstack[oldtop];
niveau:=c+1;
if integerhb[1,c]≠c then goto gem1 else
begin boo:=(cmod2)=1;
p1:=integerhb[2,c]; p2:=integerhb[2,niveau];
if (boo^p1<p2)∨(-,boo^p2<p1)∨(p1=p2^random) then
gem1:
begin i:=1;
hb[1,c]:=booleanc;
for i:=i+1 while i<3 ∨ integerhb[i,niveau]≠0 do
hb[i,c]:=hb[i,niveau]; hb[i,c]:=false
end; hb[1,niveau]:=false
end;
ii:=stacktop-2;
goto vrkd
end ikke flere træk paa dette niveau;
comp:=false;
ant:=integerstack[stacktop-3]-3;
ii:= stacktop-5-ant;
k:=ii+ant-1;

```

```
    move ahead(stack,ii,k);  
    exit:  
end comp;
```

```
booleanprocedure random;  
code ran;  
2,44  
pma1,mlnre  
gma1,sca1  
arna1,ck1  
grp-1,hvr2  
e:9901  
e;
```

```
integer procedure readkoord;  
codei;  
2,44  
panrelt58ITA  
sy58,  
hsre  
hvre7LTA  
is-57  
bss,hvre3  
sr48D  
ck10,grre4  
ck-1IOA  
ck1,src42  
hvre3LT  
hsre  
hvre7LTA  
is-10  
bss,hvre3  
ck9IOB  
ck1,pmre4  
grre4,arnre5  
srre4X  
mlre5X  
arc42NOB  
ck-1,grp-1  
hvre3LOC
```

```
hvre3NOC
arnre1,ca60
e3:src42,grp-1
hsre
e7:ck10,grc54
hvre8
e1:qq
e5:8
e4:qq
e:lyns3,ca13
src42ITA
nc60,ca58
grre1,hvre
hvreLZ
hrs1
e8:e;
```

```
procedure typeboard;
begin integer bi,i,j;
    boolean s1,s2,s3,boo;
    boo:=false;
    s1:=boolean <<xxxxx>>;
    s2:=boolean <<>>;
    s3:=boolean <<,,,,,>>;
    writecr;
    for i:=32 step -4 until 2 do
    begin
        writetext(ifboothenstrings1elsestrings2);
        for j:=0 step 1 until 3 do
        begin
            bi:=integer(booleanb[i-j,1]shift3)+1;
            writetext(casebiof(strings3,<< gal >>,<< gal >>,<< gal >>,
            <<,,,,,>>,<<,,s,,>>,<<,,,,,>>,<<,,h,,>>));
            ifj<3thenwritetext(strings1)
        end;
        if boo thenwritecr;
        writetext(strings1);
        if-,boothenwritecr;
        for j:=0 step 1 until 3 do
        begin
```

```

        bi:=integer(booleanb[i-j,1]shift3)+1;
        writetext(casebiof(strings3, << gal >, << gal >, << gal >,
        <<,,,s,,,>, <<,,,sss,,,>, <<,,,h,,,>, <<,,,hhh,,,>));
        if j<3 then writetext(strings1);
    end; if -, booth then writetext(strings1);
    writeinteger(<<,dd>, 9-(i:4)); writecr;
    if booth then writetext(strings1);
    for j:=0 step 1 until 3 do
    begin bi:=integer(booleanb[i-j,1]shift3)+1;
        writetext(casebiof(strings3, << gal >, << gal >, << gal >,
        <<,,,,,>, <<,,,s,,,>, <<,,,,,>, <<,,,h,,,>));
        if j<3 then writetext(strings1)
    end; if -, booth then writetext(strings1);
    writecr; boo:=-, boo
end; writecr;
writetext(<<,,,>);
for j:=8 step 1 until 1 do
begin writechar(48+j); writetext(<<,,,,>) end;
writecr;
end typeboard;

```

```

procedure printboard;
begin integer bi, i, j, sel;
    boolean s1, s2, s3, boo;
    sel:=select(8);
    boo:=false; s3:=410610300;
    s1:=410610644644644644644644;
    s2:=410610610610610610610;
    writecr; writecr; writechar(30);
    writechar(9); writetext(<<HVID>);
    writecr; writecr;
    for i:=32 step -4 until 2 do
    begin writetext(if booth then strings1 else strings2);
        for j:=0 step 1 until 3 do
        begin bi:=integer(booleanb[i-j,1]shift3)+1;
            writetext(casebiof(strings3, << gal >, << gal >, << gal >,
            <<,,,,,>, <<,,,s,,,>, <<,,,,,>, <<,,,h,,,>));
            if j<3 then writetext(strings1)
        end;
        if booth then writecr;
    end;
end;

```

```

writetext(strings1);if-,boothenthenwritecr;
for j:=0 step 1 until 3 do
begin bi:=integer(booleanb[i-j,1]shift3)+1;
  writetext(casebiof(strings3, << gal >, << gal >, << gal >,
    <<,,s,,>, <<,,sss,,>, <<,,h,,>, <<,,hhh,,>));
  if j<3 then writetext(strings1);
end; if-,boothenthen writetext(strings1);
writeinteger(<<,dd>, 9-(i:4)); writecr;
ifboothenthen writetext(strings1);
for j:=0 step 1 until 3 do
begin bi:=integer(booleanb[i-j,1]shift3)+1;
  writetext(casebiof(strings3, << gal >, << gal >, << gal >,
    <<,,,,>, <<,,s,,>, <<,,,,>, <<,,h,,>));
  if j<3 then writetext(strings1);
end; if-,boothenthen writetext(strings1);
writecr; boo:=-,boo
end; writecr;
writetext(<<,,>);
for j:=8 step 1 until 1 do
begin writechar(48+j); writetext(<<,,,,>) end;
writecr; writecr; writechar(30);
writechar(9); writetext(<<sort>); writechar(66);
select(sel)
end printboard;

```

```

procedure move back(a, i1, i2);
value i1, i2; integer i1, i2;
boolean array a;
begin boolean move; integer i, v, t;
  for i:=i1 step 1 until i2 do
  begin move:=a[i];
    for i:=i while integermove≠0 do
    begin t:=integer(move^3307m);
      v:=integer((moveshift-10)^3m370);
      move:=(moveshift-10)^10030m; b[t, 1]:=v
    end
  endfor
end move back;

```

```

procedure move ahead(a, i1, i2);

```

```

value i1,i2;integer i1,i2;
boolean array a;
begin boolean move;integer i,v,t;
  v:=integer((a[i1]^3003m70)shift-10);
  for i:=i1 step 1 until i2 do
    begin
      move:=a[i];
      t:=integer(move^3307m);
      move:=move^30m100shift-10;
      for i:=i while integermove≠0 do
        begin b[t,1]:=0;
          t:=integer(move^3307m);
          move:=move^30m100shift-10
        endfor;b[t,1]:=0
      end;
      if t>28vt<5 then v:=integer(booleanv^2011330);
      if t<0vt>32then begin printstack;goto exitend;
      b[t,1]:=v
    end move ahead;

boolean procedure deadpos;
deadpos:=( (state=2^niveau>goodnes) vstate=1) vniveau>25;

```

```

boolean procedure printstack;
begininteger i,ii,iii,l,tabul,sel,k,j,t,højde,bogst;
  boolean move;
  sel:=select(8); i:=ii:=1;writetext(⟨<stackudskrift:⟩);
  writecr;writecr;
  for i:=i while i≠0 do
    begin tabul:=2;
      k:=ifintegerstack[i]=0then1else3;
      for j:=0step1untilkdo
        beginwriteinteger(⟨ -ddd⟩,integerstack[i+j]);writecrend;
          i:=integerstack[i]; iii:=i;
          if i=0 then goto exit;
          for i:=i while i≠ii+4 do
            begin k:=i-4;tabul:=tabul+3;
              writechar(30);writechar(tabul);
              i:=i-integerstack[i-1];
            end
          end
        end
      end
    end
  end
end

```

```

    for j:=i step 1 until k do
    begin move:=stack[j];
      for i:=i while integer move≠0 do
      begin t:=integer(move^3307m);
        højde:=(t-1):4+1;
        bogst:=48+(2×t-1)mod8+(højde+1)mod2;
        writechar(bogst);writechar(9-højde);writechar(27);
        move:=move^30m100shift-10
      end
    end;
    for l:=1 step 1 until 3 do
    begin
      writeinteger(⟨ -ddd⟩,ifl=1then(ifstack[k+1]then1else0)elseinteger0);
      end;writecr
    end;writecr;writecr;writecr;i:=ii:=iii
end;
exit:
writecr;writecr;writecr;
writetext(⟨<udskrift af sammenligningsstack:⟩);writecr;
writecr;writetext(⟨< niveau point træk⟩);writecr;
for j:=1step1until39do
if integerhb[1,j]=jthen
begin write(⟨ -ddd⟩,integerhb[1,j],integerhb[2,j]);
  writetext(⟨<,,,,⟩);
  move:=hb[3,j];
  for i:=4,i+1while integer move≠0 do
  beginfor i:=i while integer move≠0 do
  begin t:=integer(move^3307m);
    højde:=(t-1):4+1;
    bogst:=48+(2×t-1)mod8+(højde+1)mod2;
    writechar(bogst);writechar(9-højde);writechar(27);
    move:=move^30m100shift-10
  end;move:=hb[i,j]
  end;writecr
end;writechar(66);
select(sel)
end printstack;

11:if where(⟨<dabræt⟩,db)≠0 then

```

```

begin
  select (16);
  for i:=1 step 1 until 32 do
  for j:=1 step 1 until 6 do b[i,j]:=0;
  for i:=0 step 1 until 10 do
  for j:=1 step 1 until 40 do hb[i,j]:=false;
  hb[10,40]:=4010000001;
  if reserve(⟨<dambræt⟩,1)≠0 then
  begin
    writetext(⟨<fejl i reserve⟩);
    goto fejl;
  end;
  if where(⟨<dambræt⟩,db)≠0 then
  begin
    writetext(⟨<fejl i reserve⟩);
    goto fejl;
  end;
end;

```

igen:

```

i:=read integer;
if i>0 then
begin
  b[i,1]:=integer ((boolean read integer)shift -3);
  for j:=2 step 1 until 6 do b[i,j]:=read integer;
  goto igen
end;
put (hb,db,1);
put (b,db,1);
end læs b;
15: if get (hb,db,1)<0 then goto 15;
hb[10,40]:=boolean(integerhb[10,40]-1);
ran:=integerhb[10,40];
16: if put (hb,db,1)<0 then goto 16;
12: if get (b,db,1)< 0 then goto 12;
select (16); writechar(66);
writetext(⟨< inputkonventioner for gier - damspil:

```

```

<farve>      ::=<sort>|<hvid>

```

spilleren taster hvilken farve brikker han  
 ønsker at spille med. hvid begynder altid.

```
<styrke> ::= <heltal>,
```

det tal der tages angiver det minimale antal træk, som gier ser frem. tal  $\leq 1$  opfattes som 1, og tal  $\geq 25$  opfattes som 25. med styrke lig 2 tager det ca. 15 sek. for gier at trække, og med styrke lig 3 tager det ca. 2 min. normalt vil styrke 3 give passende modstand.

```
<din tur:> ::= <startpos> <mellempos> <slutpos><
<startpos> ::= <pos>,
<mellempos> ::= |<pos>|<pos>,<mellempos>
<slutpos> ::= <pos>
<pos> ::= <bogstav><højde>
<bogstav> ::= a|b|c|d|e|f|g|h
<højde> ::= 1|2|3|4|5|6|7|8
```

notationen er sædvanlig skakbrætnotation. hvis man prøver at taste en position ind som ikke er korrekt, svares der med annull, og hele trækket er annulleret. desuden kan annullering ske efter sædvanlig help 3 konvention.

```
det kan anbefales at udstyre sig med et dambræt naar man vil
spille med gier, men har man ikke det, vil kaon bevirke udskrift
af et dambræt paa linieskriveren, hver gang gier har trukket.});
writechar(66);printboard;
select(17);writecr;fori:=1step1until40dohb[1,i]:=false;
writetext(⟨<GIER - DAMSPIL⟩);writecr;
```

```
14:
writecr;
writetext(⟨<farvevalg: ⟩);j:=lyn;
if j=18 then
begin writetext(⟨<ort⟩); valg:=false end else if j=56 then
begin writetext(⟨<vid⟩); valg:=true end else goto 14;
writecr;writetext(⟨<styrke: ⟩);goodnes:=readinteger;
if-, valg then goto os;
start:
```

```

color:=valg;
stacktop:=3;
stack[1] := 39 0 11;
oldtop:= stacktop;
stack[2] := false;
niveau := 0;
lookforward(color);
if state=1 then
begin writetext(⟨<jeg vinder dette spil⟩); goto exitend;
om igen:
i:=1; dm:=false; q:=0;
writecr; writetext(⟨<din tur  ⟩);
comment indlæsning af modstanders træk;
char:=0;
for k:=k while char ≠17 do
begin k:=readkoord;
  if char<0vk>32vk≤0 then
  begin writecr;
    for i:= 29,49,37,37,20,35,62 do
    writechar(i); goto om igen
  end;
  if q=40 then
  begin q:=0;draw[i]:=dm;
    dm:= false; i:=i+1
  end;
  dm:=dmv(((booleank)∨(booleanb[k,1]shift10))shiftq);
  q:=q+10;
end;
draw[i]:=dm;
draw[i+1]:= false; comment slut indlæsning af træk;
comment kontrol af modstanders træk;
mi:=i; ii:=stacktop-3;
nyttræk:
ant:=integerstack[ii]-3;
ii:=ii-ant-3;
if mi≠ant then
next:
begin if ii=4 then
  begin writecr;
    writetext(⟨<ej lovligt træk - prøv igen⟩);

```

```

    goto omigen
end;
goto nyt træk
end;
di:=draw[1];
for i:=1 step 1 until ant do
if integer di+integerstack[i+ii] then gotonext
else di:=draw[i+1];

comment modstanderens træk trækkes;
move ahead(draw,1,mi);

comment vi begynder at se os om efter et modtræk;
os:stacktop:=oldtop:=3;
stack[1]:=39011;stack[2]:=false;niveau:=0;
color:=-,valg;
ii:=look forward(color);
if state=1 then
begin writetext(⟨<du vinder dette spil⟩); goto exit end;
if alt=1 then
begin for i:=ii step 1 until stacktop-6 do
hb[i-ii+3,1]:=stack[i];
hb[i-ii+3,1]:=false
end else
begin
move ahead(stack,ii,stacktop-6);
slut:=false;
for i:=i while -,slut do
begin for i:=i while -,dead pos do
begin color:=(stack[oldtop]shift-1)=valg;
ii:=look forward(color);
if state>1 then move ahead(stack,ii,stacktop-6);
endfor;ifkbonthenprintstack;
slut:=comp; niveau:=integerstack[oldtop];
state:=ifstack[stacktop-5]then1else3
end
end;
comment paa dette sted i programmet skulle vi
have fundet det bedste træk,som derfor lægges ind nu;
writecr;writetext(⟨<jeg trækker ⟩);

```

```

hb[1,1]:=false;
i:=2; m:=hb[3,1];
v:=(mshift-10)^3m370;
t:=integer(m^3307m);
højde:=((integer)t-1):4+1;
bogst:=48+((2*(integer)t)-1)mod8+(højde+1)mod2;
writechar(bogst);writechar(9-højde);
m:=(mshift-10)^10030m;
fori:=i+1 while integerm≠0 do
begin for i:=i while integerm≠0 do
begin b[t,1]:=0;
t:=integer(m^3307m);
højde:=((integer)t-1):4+1;
bogst:=48+((2*(integer)t)-1)mod8+(højde+1)mod2;
writechar(27);writechar(bogst);
writechar(9-højde);
m:=(mshift-10)^10030m;
end;
m:=hb[i+1,1]
end;writechar(17);
if t<5vt>28 then v:=vv2011330;
if t<1vt>32 then begin printstack; goto exit end;
b[t,1]:=integerv;
ifkaonthenprintboard;
typeboard;
goto start;

exit:
writecr; writetext(⟨<skal vi prøve igen⟩);j:=lyn;
ifj=33thenbeginwritechar(49);writecr;gotol2endelseifj=37then
beginwritechar(53);writechar(33);writecrendelse goto exit;

```

```

writetext(⟨<tak for spillet⟩);writecr;

```

```

fejl:

```

```

endt<

```

```

Ff

```

```

1, 4, 1, 0, 0, 5, 0,
2, 4, 2, 0, 0, 6, 5,
3, 4, 2, 0, 0, 7, 6,
4, 4, 2, 0, 0, 8, 7,

```

5, 4, 4, 2, 1,10, 9,  
6, 4, 4, 3, 2,11,10,  
7, 4, 4, 4, 3,12,11,  
8, 4, 2, 4, 0,12, 0,  
9, 4, 2, 5, 0,13, 0,  
10, 4, 4, 6, 5,14,13,  
11, 4, 4, 7, 6,15,14,  
12, 4, 4, 8, 7,16,15,  
13, 0, 4,10, 9,18,17,  
14, 0, 4,11,10,19,18,  
15, 0, 4,12,11,20,19,  
16, 0, 2,12, 0,20, 0,  
17, 0, 2,13, 0,21, 0,  
18, 0, 4,14,13,22,21,  
19, 0, 4,15,14,23,22,  
20, 0, 4,16,15,24,23,  
21, 6, 4,18,17,26,25,  
22, 6, 4,19,18,27,26,  
23, 6, 4,20,19,28,27,  
24, 6, 2,20, 0,28, 0,  
25, 6, 2,21, 0,29, 0,  
26, 6, 4,22,21,30,29,  
27, 6, 4,23,22,31,30,  
28, 6, 4,24,23,32,31,  
29, 6, 2,26,25, 0, 0,  
30, 6, 2,27,26, 0, 0,  
31, 6, 2,28,27, 0, 0,  
32, 6, 1,28, 0, 0, 0,  
-1,