


```

    { dies ist fr eine flssige Animation unerl,,álich }
    mov ax,$1010
    mov bx,255
    int 10h
end;
*)
{----- Init -----}

PROCEDURE Init ;
var i : 1..12;
    j : word ;
begin

    { Eingabeschirm }

    textcolor(white);
    textbackground(blue);
    clrscr;
    gotoxy(10,1); write('D i e   T r m e   v o n   H a n o i   ');
    gotoxy(10,2); write('----- ');
    gotoxy(10,4); write('Anz. Scheiben ó 12 : '); readln( Anzahl );
    gotoxy(50,5); write('(1 bis 4 mit Kran)') ;
    gotoxy(10,5); write('Tempo (min=1 max=9) : '); readln( T );

    textcolor(lightblue);
    gotoxy(1,23); for j := 1 to 80 do write('²');
    textcolor(white);

    { St,,ebe initialisieren }

    for i:= 1 to Anzahl do Stab[1].Scheibe[i].Form := S[Anzahl+1-i];
    for i:= 1 to 3 do Stab[i].const_x := (i-1)*25 + 2 ;
    for i:= 1 to 3 do Stab[i].const_y := 23 ;
    Stab[1].Zaehler:=Anzahl; Stab[2].Zaehler:=0; Stab[3].Zaehler:=0;

    { Scheiben vom 1.Stab zeichnen }

    for i := 1 to Anzahl do begin
        gotoxy( 2,23 - i);
        Stab[1].Scheibe[i].zeichnen
    end;

    { Schiene, Laufkatze und Greifer zeichnen }

    Katze.Schiene_zeichnen(Stab[1]);
    gotoxy(80,25); { Cursor an den Rand }

    { weitere Initialisierungen }

    Bewegungen := 0 ;
    Wdhlg := 'j';
    T_ := T ;
    delay(1000)
end;

{----- Methoden von TScheibe -----}

procedure TScheibe.loeschen;
begin
    write(S[0])
end;

procedure TScheibe.zeichnen;
begin
    write(Form)
end;

{----- Methoden von TStab -----}

procedure TStab.Scheibe_aufnehmen;
begin
    inc(Zaehler);

```

```

    Scheibe[Zaehler] := bewegteScheibe ;
    bewegteScheibe.form := S[0] ;
    gotoxy(const_x, const_y - Zaehler) ;
    Scheibe[Zaehler].zeichnen ;
    delay(Tempo[T])
end;

```

```

procedure TStab.Scheibe_abgeben ;
begin
    bewegteScheibe := Scheibe[Zaehler] ;
    gotoxy(const_x, const_y - Zaehler) ;
    Scheibe[Zaehler].loeschen ;
    dec(Zaehler) ;
end;

```

```

{----- Methoden von TKatze -----}

```

```

procedure TKatze.Schiene_zeichnen(a : TStab);
var i : integer;
begin
    gotoxy(1,6); for i := 1 to 80 do write('Í');

    gotoxy(a.const_x + 7, 6); write('íËíííííííííí');
    gotoxy(a.const_x + 7, 7); write(' Èíííííííííí ');
    gotoxy(a.const_x + 7, 8); write('      3 3      ');
    gotoxy(a.const_x + 7, 9); write(' ..... ');

    Position := a.const_x;
    inAktion := true;
end;

```

```

procedure TKatze.Greifer_senken(a : TStab);
var i : integer;
begin

    if inAktion then

        for i:= 9 to ( 22 - a.Zaehler ) do
            begin
                gotoxy(a.const_x,i-1);write('      3 3      ');
                gotoxy(a.const_x,i );write(' ..... ');
                delay(Tempo[T]);
            end
        end;

```

```

procedure TKatze.Greifer_heben (a : TStab);
var i : integer;
begin

    if inAktion then

        for i:= ( 21 - a.Zaehler ) downto 9 do
            begin
                gotoxy(a.const_x,i );write(' ..... ');
                gotoxy(a.const_x,i+1);write(' ..... ');
                delay(Tempo[T]);
            end
        end;

```

```

procedure TKatze.Scheibe_heben (a : TStab);
var i : integer;
begin

    if inAktion then

        for i:= ( 19 - a.Zaehler ) downto 9 do
            begin
                gotoxy(a.const_x,i );write(' ..... ');
                gotoxy(a.const_x,i+1); bewegteScheibe.zeichnen ;
                gotoxy(a.const_x,i+2);write(' ..... ');
                delay(Tempo[T]);
            end
        end;

```

```
end;
```

```
procedure TKatze.Scheibe_senken(a : TStab);  
var i : integer;  
begin
```

```
    if inAktion then
```

```
        for i:= 9 to ( 21 - a.Zaehler ) do  
            begin
```

```
                gotoxy(a.const_x,i-1);write('          3 3          ');  
                gotoxy(a.const_x,i  );write('          .....          ');  
                gotoxy(a.const_x,i+1);          bewegteScheibe.zeichnen ;  
                delay(Tempo[T]);  
            end
```

```
        end;
```

```
procedure TKatze.verschieben(a : TStab);  
var richtung, schritte,  
    i : integer;  
begin
```

```
    if inAktion then begin
```

```
        richtung := a.const_x - Position ;  
        schritte := abs(a.const_x - Position);
```

```
        for i := 0 to schritte do  
            begin
```

```
                gotoxy(position + 7, 6); write('íÉíííííííÉí');  
                gotoxy(position + 7, 7); write(' ÈííØØíí¼ ');  
                gotoxy(position + 7, 8); write('          3 3          ');  
                gotoxy(position + 7, 9); write('          .....          ');  
                gotoxy(position      ,10); bewegteScheibe.zeichnen;  
                delay(Tempo[T]); {wait ;}
```

```
                if richtung > 0 then inc(position)      { vorw,,rts }  
                    else dec(position);                { rckw,,rts }
```

```
            end
```

```
        end  
    end;
```

```
{----- Prozeduren fr Cursor -----}
```

```
(*
```

```
PROCEDURE Cursor_aus ; assembler ;  
asm                                { BIOS-Interrupt 10h }  
    mov ah,1                        { Cursorgr"ae setzen }  
    mov ch,1                        { Startzeile Cursor }  
    mov cl,0                        { Endzeile Cursor => }  
    int 10h                         { nicht sichtbar }  
end;
```

```
PROCEDURE Cursor_an ; assembler ;  
asm                                { Cursorgr"ae setzen }  
    mov ah,1                        { Startzeile Cursor }  
    mov ch,13h                     { Endzeile Cursor }  
    mov cl,14h                     { Standard fr VGA }  
    int 10h                         { Standard fr VGA }  
end;
```

```
*)
```

```
{----- Interruptprozedur Tempo_ -----}
```

```
(*
```

```
{ $F+ }  
PROCEDURE newexit;  
begin  
    exitproc := oldexit;  
    SetIntVec($08,oldint)  
end;
```

```
PROCEDURE Tempo_; interrupt;  
var Taste : char ;
```

```

        Fehler: integer;
begin
  asm
    pushf                    { Interrupt simulieren }
    call [oldint]           { alten Intr. aufrufen }
  end;
  if keypressed then begin
    Taste := readkey;
    if Taste in ['1'..'9'] then begin
      Val(Taste,T_,Fehler);
      gotoxy(36,5);write('(',T_,')')
    end
  end
end;

```

```

PROCEDURE Exitprozedur;
begin
  oldexit := exitproc;
  ExitProc := @NewExit;
  GetIntVec($08,oldint);
end;

```

*)

```

{----- rekursive Prozedur Hanoi -----}

```

```

PROCEDURE Hanoi(k, a, b: Integer);
begin
  if k >= 1 then begin
    Hanoi(k - 1, a, 6 - a - b);      { 1. Rekursion }

    Katze.verschieben (Stab[a]);
    Katze.Greifer_senken(Stab[a]);
    Stab[a].Scheibe_abgeben;        { von a }
    Katze.Scheibe_heben (Stab[a]);
    Katze.verschieben (Stab[b]);
    Katze.Scheibe_senken(Stab[b]);
    Stab[b].Scheibe_aufnehmen;      { nach b }
    Katze.Greifer_heben (Stab[b]);
    T := T_;
    gotoxy(32,5);write(T,' ');
    if T<=4 then Katze.inAktion := true
      else Katze.inAktion := false;
    Bewegungen := Bewegungen + 1;  { z„hlen }

    Hanoi(k - 1, 6 - a - b, b);     { 2. Rekursion }
  end
end;

```

```

{----- Hauptprogramm -----}

```

BEGIN

```

  repeat
    Init;
    Hanoi(Anzahl,1,2);
    gotoxy(10,25); write('Anzahl Bewegungen : ',Bewegungen);
    gotoxy(55,25); write('Noch einmal (J/N) : '); Wdhlg := readkey;
  until (wdhlg = 'N') OR (wdhlg = 'n')

```

END.