

- 1 Geben Sie den Befehl an, mit dem man in Delphi ein Programm „geordnet“ beendet.
`application.terminate;`

- 2 Es soll die Summe aller geraden Zahlen von 0 bis 2000 berechnet werden. Realisieren Sie das durch jeweils ein Programm unter Verwendung einer Es reicht, wenn Sie die wesentlichen Programmzeilen aufführen.

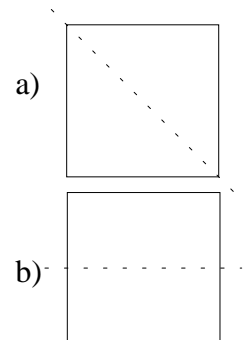
a) for-Schleife	b) repeat-until-Schleife	c) while-Schleife
<pre>var Summe,i:integer; ... Summe:=0; for i:=0 to 1000 do Summe:=Summe+2*i; ...</pre>	<pre>var Summe,i:integer; ... Summe:=0; i:=0; repeat Summe:=Summe+i; i:=i+2; until i>2000; ...</pre>	<pre>var Summe,i:integer; ... Summe:=0; i:=0; while i<=2000 do begin Summe:=Summe+i; i:=i+2; end; ...</pre>

- 3 `var Feld:array[1..10,1..10] of integer;`
 Die Inhalte eines derart definierten Feldes sollen jeweils wie im Unterricht behandelt an einer Geraden gespiegelt werden. Die Lage der Geraden ist den beiden nebenstehenden Skizzen zu entnehmen.

- a) an der Winkelhalbierenden von oben links nach unten rechts
 b) an einer waagrechten Linie in Bildmitte

Schreiben Sie die entsprechenden Programmzeilen für beide Fälle.

a)	<code>var x,y,zw:integer;</code>	b)	<code>var x,y,zw:integer;</code>
	<code>...</code>		<code>....</code>
	<code>for y:=1 to 10 do</code>		<code>for x:=1 to 10 do</code>
	<code>for x:=1 to y do</code>		<code>for y:=1 to 5 do</code>
	<code>begin</code>		<code>begin</code>
	<code> zw:=Feld[x,y];</code>		<code> zw:=Feld[x,y];</code>
	<code> Feld[x,y]:=Feld[y,x];</code>		<code> Feld[x,y]:=Feld[x,11-y];</code>
	<code> Feld[y,x]:=zw;</code>		<code> Feld[x,11-y]:=zw;</code>
	<code>end;</code>		<code>end;</code>
	<code>...</code>		<code>...</code>

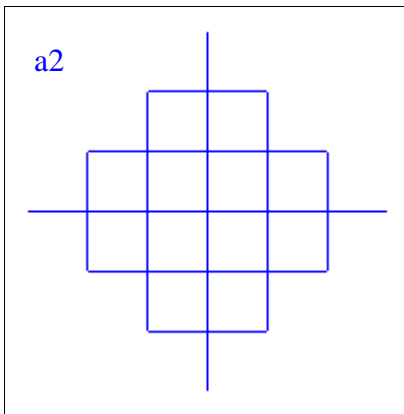
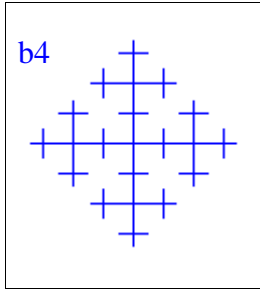
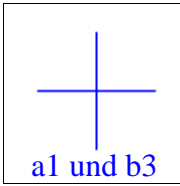


- 4 Zeichnen Sie auf oder beschreiben Sie genau, was durch das nebenstehende Programm auf dem Bildschirm ausgegeben wird.

Unterscheiden Sie dabei folgende 4 Fälle:

- a) Der Faktor in der Prozedur Zeichne hat den Wert 1,0.
 Im Edit1-Feld steht im ersten Fall eine 1 und im zweiten Fall eine 3.
 b) Der Faktor in der Prozedur Zeichne hat den Wert 0,5.

Im Edit1-Feld steht im dritten Fall eine 1 und im vierten Fall eine 3.



```

procedure Linie(x1,y1,x2,y2:extended);
begin
  Form1.Imagel.Canvas.MoveTo(round(x1),round(y1));
  Form1.Imagel.Canvas.LineTo(round(x2),round(y2));
end;

procedure Zeichne(n:integer;x,y,L:extended);
const Faktor=1.0; // bei Aufgabe b: Faktor:=0.5
begin
  if n>0
  then begin
    Linie(x,y,x+L,y);Zeichne(n-1,x+L,y,L*Faktor);
    Linie(x,y,x,y+L);Zeichne(n-1,x,y+L,L*Faktor);
    Linie(x,y,x-L,y);Zeichne(n-1,x-L,y,L*Faktor);
    Linie(x,y,x,y-L);Zeichne(n-1,x,y-L,L*Faktor);
  end;
end;

procedure TForm1.BZeichneClick(Sender: TObject);
begin
  LoescheBild;
  Zeichne(StrToInt(Form1.Edit1.text),250,250,30);
end;

```

- 5 Die Potenz y^x mit frei wählbarem y und x , wobei y aus dem Bereich der reellen Zahlen und x aus dem Bereich der natürlichen Zahlen zu wählen ist soll mit Hilfe der **rekursiven** Funktion `function pot(x:integer; y:extended):extended; berechnet werden.`

Schreiben Sie die entsprechenden Programmzeilen dieser Prozedur.

```

function pot(x:integer;y:extended):extended;
begin
  if x>1
  then result:=y*pot(x-1,y)
  else result:=y;
end;

```

- 6 Es soll simuliert werden, wie sich zwei anfangs getrennte Gase mittels Diffusion durchmischen. Dazu wird ein StringGrid (20 Spalten, 10 Zeilen) wie nebenstehend mit zwei Symbolen (für die zwei Gassorten) gefüllt. Ihre Aufgabe ist es nun, die Programmzeilen in der unten stehenden Prozedur zu ergänzen, so dass folgendes geschieht: Zunächst wird per Zufall eine Zelle des StringGrid ausgewählt (x- und y-Koordinate erwürfeln).

```

procedure TForm1.FormCreate(Sender: TObject);
var x,y:integer;
begin
  for y:=0 to 9 do
  for x:=0 to 9 do begin
    Form1.Stringgrid1.Cells[x,y]:='O';
    Form1.Stringgrid1.Cells[x+10,y]:='X';
  end;
end;

procedure TForm1.BWeiterClick(Sender:
TObject);
var ....
begin
  ....
end;

```

Dann wird per Zufall die Zelle über, unter, rechts oder links von diesem Element gewählt. Die Inhalte der beiden gewählten Zellen werden ausgetauscht.

```
var x,y,xx,yy:integer;
    zw:string;
begin
  repeat
    x:=random(20);
    y:=random(10);
    repeat xx:=random(3)-1;yy:=random(3)-1;until (xx*yy=0)and(xx<>yy);
  until (x+xx>-1)and(x+xx<20)and(y+yy>-1)and(y+yy<10);
  zw:=Form1.StringGrid1.Cells[x,y];
  Form1.StringGrid1.Cells[x,y]:=Form1.StringGrid1.Cells[x+xx,y+yy];
  Form1.StringGrid1.Cells[x+xx,y+yy]:=zw;
end;
```

7 Für die 20 Schüler, die alle sowohl einen Mathematik- als auch einen Informatik-Kurs belegt haben, wurde nebenstehendes Feld angelegt. Schreiben Sie die Programmzeilen, durch die folgende Aufgaben erledigt werden:

```
type tSchueler=
  record
    Name:string;
    Mathe, Inf: integer;
  end;
var Schueler:array[1..20] of tSchueler;
```

a) Diejenigen der 20 Schüler, deren Durchschnittsnote aus den Ergebnissen in den Fächern Mathematik und Informatik höher als 11 Punkte ist, sollen in einem Memo-Feld (Memo1) ausgegeben werden (pro Schüler eine Zeile, der Name genügt).

```
var i:integer;
...
for i:=1 to 20 do
  if (Schueler[i].Mathe+Schueler[i].Inf)/2>11
  then Form1.Memo1.Lines.Add(Schueler[i].Name);
...

```

b) Alle 20 Schüler sollen ihren Ergebnissen (15-Punkte-System) im Fach Mathematik nach geordnet in einem Memo-Feld (Memo2) ausgegeben werden, oben die besten, unten die schlechtesten.

```
var i,k:integer;
...
for i:=1 to 20 do
  for k:=15 downto 0 do
    if Schueler[i].Mathe=k
    then Form1.Memo2.Lines.Add(Schueler[i].Name+' '+IntToStr(Schueler[i].Math));
...

```

