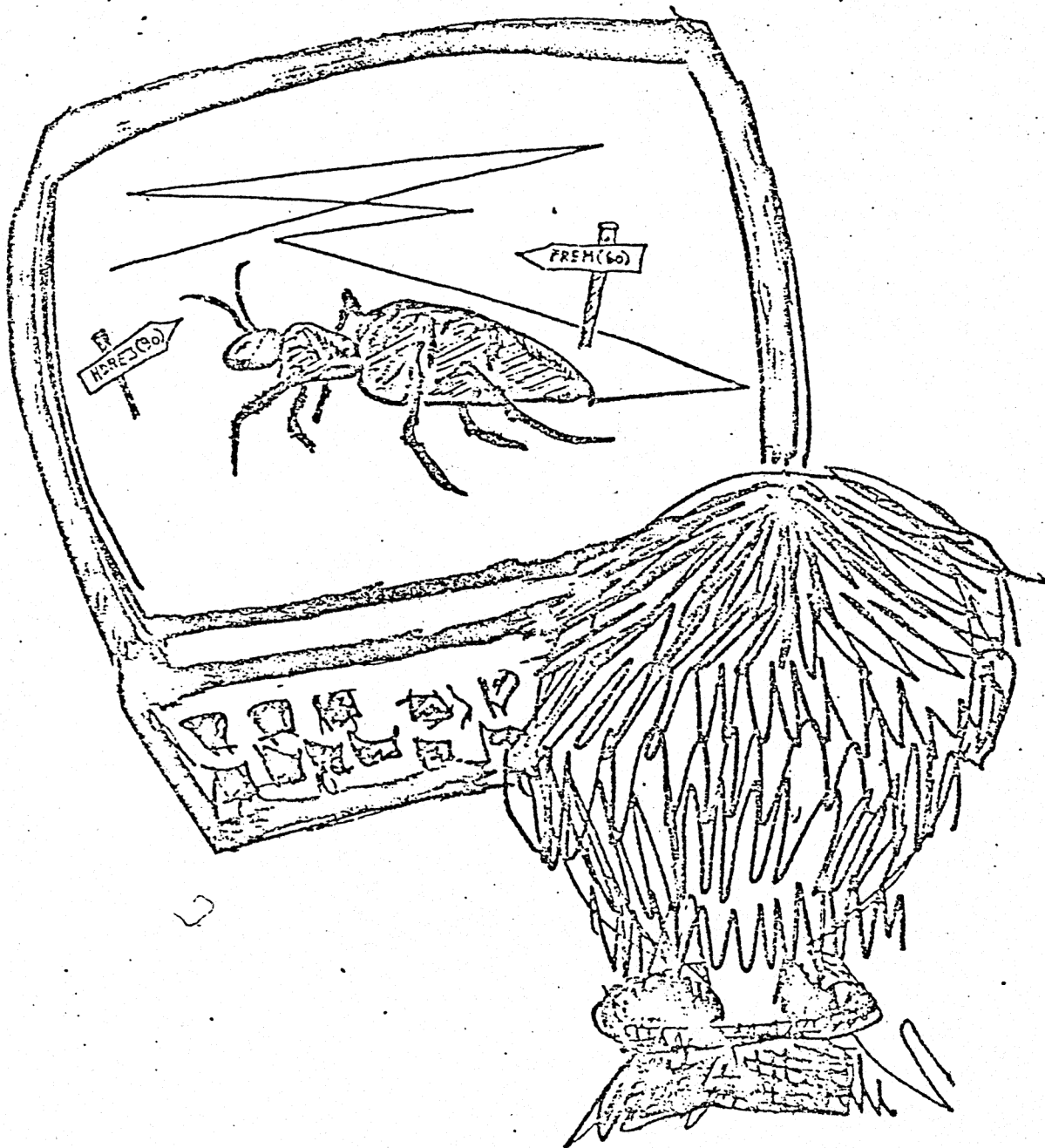


VIGGO SADOLIN

MYREGEOMETRI



## Forord

Dette undervisningsmateriale sigter primært mod anvendelse i skolens matematikundervisning på 5.-10. klassetrin.

§6 samt de sidste af opgaverne bagerst i hæftet sigter dog mod gymnasiet og seminarier og er for vanskelige til folkeskolen.

Hæftet er iøvrigt af interesse for enhver, som ønsker at stifte bekendtskab med emnerne programmering og mikrodatamater.

## INDHOLD:

## INDLEDNING

1. Myreordrer.
2. Myreprogrammer.
3. Myreprogram kalder myreprogram.
4. Variable.
5. Myreprogram kalder sig selv.
6. Tilbagevikling.

Opgaver.

Indledning.

En myre befinder sig på en strandbred med meget fint sand. Når myren kravler sætter den et tydeligt og smukt spor i sandet. Hvis myren flyver, så rører den slet ikke sandet og sætter derfor intet spor.

Myren er måske lidt speciel. Den ligner til forveksling en trekant, og så kan den, hvis man beder den om det, gøre sig usynlig.

Du kan ikke se hele strandbredden, men blot et rektangulært stykke af den. Myren kan godt bevæge sig uden for det felt, du kan se; men den følger dine ordrer, selv når du ikke kan se den. Til at begynde med står myren med alle ben på sandet, dens næse peger nordpå, den befinder sig lidt til venstre for midten, og du kan se den. Myren er klar til at modtage dine ordrer, og den er i stand til at huske selv ret indviklede instruktioner, blot du forklarer den det på myrens sprog, som hedder MYRESNAK.

1. Myreordrer.

Til at begynde med kan myren udføre nogle simple ordrer, som vi vil kalde for myreordrer. Du kan bede myren om at gå frem, bakke, dreje til højre eller til venstre.

Du giver myren en ordre ved at skrive ordren og trykke på RETURN-tasten. Myren udfører da straks ordren.

Øvelse 1.1

Skriv FLEM(50) og tryk RETURN - se, hvad der sker.

Skriv HDREJ(90) - - - -

FREM(100)

VDREJ(45)

BAK(150)

HDREJ(45)

- fortsæt som du selv synes.

Øvelse 1.2

Slet skærmen ved at skrive RENS og tegn derpå, hvad du har lyst til.

---

I øjeblikket befinder myren sig nede på sandet. Den kravler og laver spor i sandet. Hvis du skriver FLYV, så rører myren ikke mere sandet, og så laver den heller ikke noget spor. Myren bliver ved med at flyve, lige indtil du skriver KRAVL. Myren er da nede på sandet igen.

### Øvelse 1.3

Rens skærmen og skriv:

```
FREM(40)
HDREJ(120)
FLYV
FREM(80)
KRAVL
HDREJ(120)
FREM(80)
```

### Øvelse 1.4

Skriv RENS og lav en tegning.

---

Når myren laver en tegning, sætter den et hvidt spor på en sort baggrund. Hvis myren laver en streg, som du gerne vil fjerne uden at rense hele skærmen, så kan du gøre det ved at bede myren om at lave samme streg med farven sort.

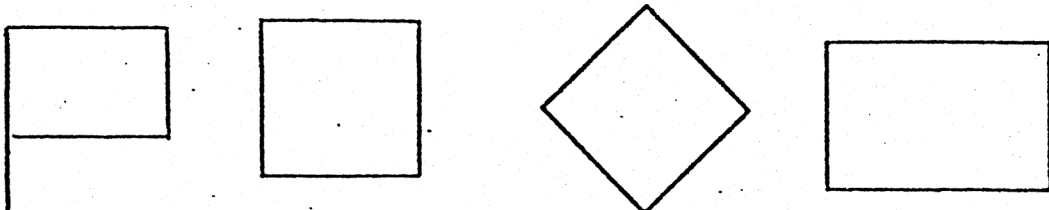
### Øvelse 1.5

Rens skærmen og skriv:

```
FREM(45)
HDREJ(90)
FREM(80)
SORT
BAK(80)
HDREJ(45)
FREM(40)
HVID
FREM(45)
```

### Øvelse 1.6

Prøv om du kan få myren til at lave følgende figurer:



---

Du kan få myren til at forsvinde ved at skrive ERYM. Den kommer først igen, når du skriver MYRE.

### Øvelse 1.7

Lav en tegning, uden at myren er synlig.

## 2. Myreprogrammer.

Myren tegner en trekant med sidelængde 70, hvis du beder den udføre ordrene:

```
FREM(70)
HDREJ(120)
FREM(70)
HDREJ(120)
FREM(70)
HDREJ(120)
```

Den sidste ordre er egentlig ikke nødvendig, men den bevirker, at myren peger i samme retning, som da den begyndte at tegne trekanten.

### Øvelse 2.1

Hvis du ikke allerede har prøvet det, så få myren til at tegne trekanten.

---

Det er muligt at få myren til at huske opskriften på, hvorledes denne trekant tegnes. Du skal blot skrive HUSK, før du giver den opskriften. Opskriften vil vi kalde for et myreprogram.

Myreprogrammet kan se sådan ud:

```
TREKANT
  FREM(70)
  HDREJ(120)
  FREM(70)
  HDREJ(120)
  FREM(70)
  HDREJ(120)
SLUT
```

### Øvelse 2.2

Skriv HUSK og indtast myreprogrammet TREKANT. Skriv TEGNING, og du kommer tilbage til tegnesiden. Myren har nu lært, hvorledes denne trekant tegnes. Du skal blot skrive TREKANT, så laver den trekanten. Prøv det.

---

Nu kan det jo godt være, at du fik skrevet noget forkert, da du indtastede myreprogrammet. Myren tegnede måske slet ikke en trekant, eller myren sagde BIB (en høj lyd), da du skrev TREKANT. Det siger den nemlig, når den møder en ordre, den ikke genkender. Hvis du ønsker at lave om på dit indtastede program, så skal du skrive RET. Du får da mulighed for at udskifte, slette eller indsætte linier i dit program.

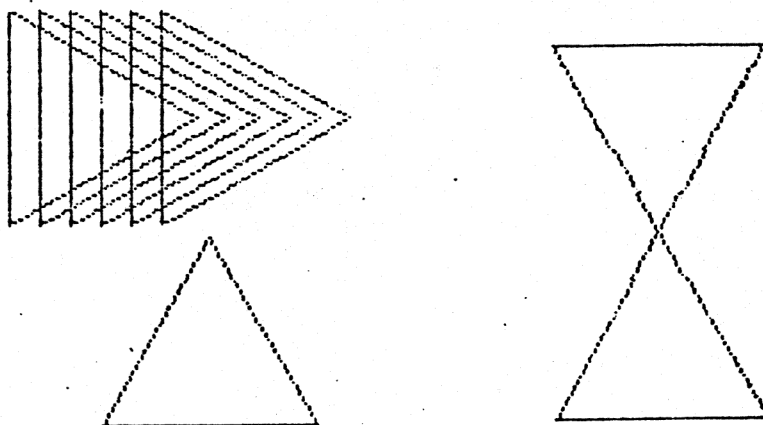
### Øvelse 2.3

Hvis dit myreprogram TREKANT ikke virkede tilfredsstillende, så ret det ved at skrive RET.

Hvis dit program virkede, som du ønskede, så prøv alligevel at undersøge, hvorledes RET fungerer. Du kan for eksempel ændre programmet, så det tegner en trekant af en anden størrelse.

### Øvelse 2.4

Benyt TREKANT til at tegne figurer, som vist nedenfor.



### Øvelse 2.5

Lav et myreprogram, FIRKANT, som fortæller myren, hvorledes den skal kravle i firkant (kvadrat eller rektangel).

Benyt programmet til at tegne figurer lige som i øvelse 2.4.

---

Hvis du skriver KATALOG, får du at vide, hvor meget du har lært myren - det vil sige hvilke myreprogrammer, du har indtastet.

### Øvelse 2.6

Lav selv nogle flere myreprogrammer, hvor du selv finder på navne for programmerne. Afprøv hvert af dine programmer.

Når du er færdig, kan du skrive KATALOG.

---

Der er kun plads til et begrænset antal programmer i datamaskinen. Hvis du ønsker at skabe plads, kan du slette programmer ved at skrive SLET. Vigtige programmer smider man jo ikke væk, og der er mulighed for at gemme sådanne programmer ved at skrive GEM. GEM bevirker at dit program gemmes på din private diskette. Et program, som ligger på disketten, kan først benyttes af myren, når du har skrevet HENT. Alle de programmer, der ligger i datamaskinen (altså ikke dem på disketten) slettes, når du stopper MYRESNAK ved at skrive STOP eller slukker for datamaskinen.

#### Øvelse 2.7

Prøv at anvende ordrene SLET, GEM og HENT. Ordren KATALOG giver dig kun navnene på programmerne i datamaskinen (ikke dem på disketten).

Når du laver et myreprogram, og du ønsker at gentage et sæt myreordrer et bestemt antal gange, kan det udtrykkes kort ved at benytte ordene GENTAG og HERTIL.

Programmet TREKANT gentager ordrene FREM(70) og HDREJ(120) tre gange. Det kan udtrykkes sådan:

```
TREKANT
  GENTAG(3)
    FREM(70)
    HDREJ(120)
  HERTIL
SLUT
```

En sådan gentagelse kaldes en sløjfe.

#### Øvelse 2.8

Prøv programmet TREKANT ovenfor. Gem TREKANT på disketten.

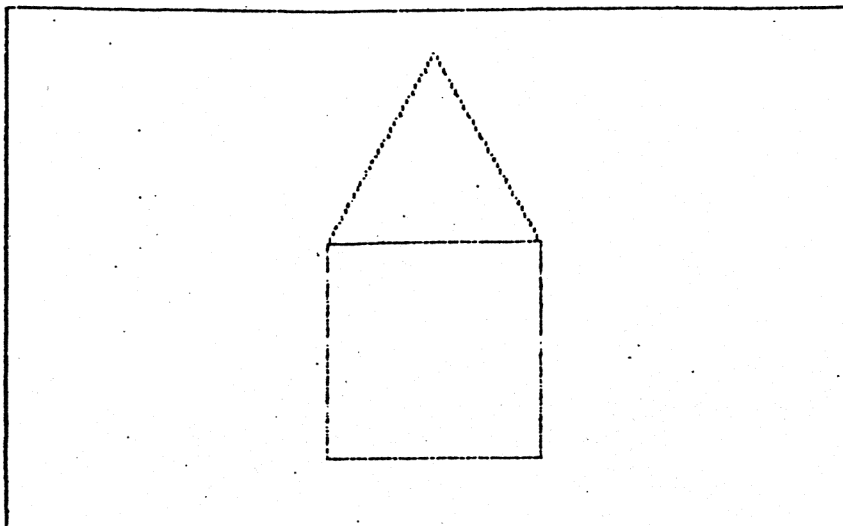
#### Øvelse 2.9

Prøv at lave et program, KVADRAT, der tegner et kvadrat ved at benytte en sløjfe. Gem programmet.

#### Øvelse 2.10

Prøv at få myren til at tegne et hus som vist nedenfor.





### 3. Myreprogram kalder myreprogram.

Når vi har lært myren at kravle i en trekant ved hjælp af myreprogrammet TREKANT, så kan vi benytte ordren TREKANT, når vi ønsker at lære myren en ny figur. De ting, vi har lært myren, kan vi altså bruge i andre myreprogrammer.

#### Øvelse 3.1

Hent TREKANT fra disketten. List TREKANT ved at skrive LIST.

Indtast derpå følgende myreprogram

```
DREJ
  GENTAG(6)
    TREKANT
  HDREJ(10)
  HERTIL
  SLUT
```

Prøv programmet.

Udskift ordren HDREJ(10) med en ordre, der anbringer de 6 trekanter, så de danner en sekskant.

Lav ændringer i programmet, så det laver en masse trekanter drejet hele vejen rundt.

#### Øvelse 3.2

Udskift TREKANT med FIRKANT i øvelse 3.1 og lav nogle pæne figurer.

#### Øvelse 3.3

Lav et myreprogram, der hedder HUS, og som tegner huset fra øvelse 2.10.

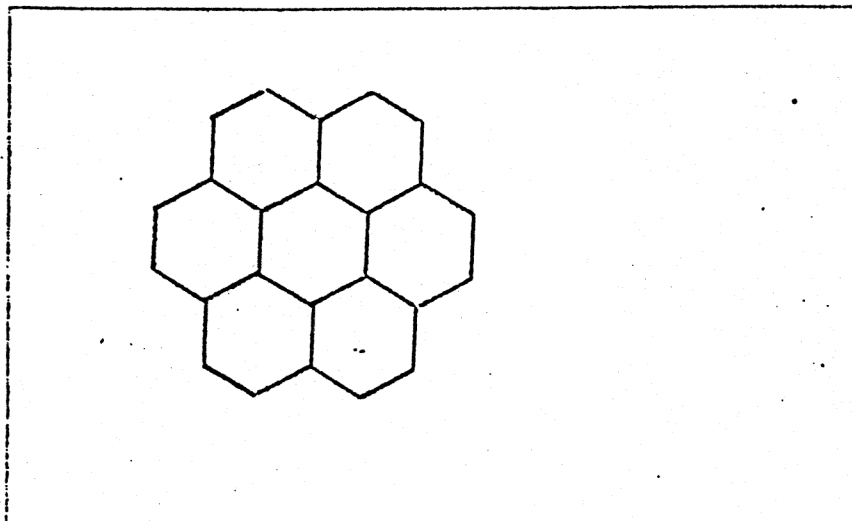
#### Øvelse 3.4

Lav et program, der tegner to ens huse ved siden af hinanden. Her kan du for eksempel benytte ordren START, som anbringer myren i start-situationen, efter at det første hus er tegnet.

Når et myreprogram benytter et andet myreprogram, så siger vi, at det første program *kalder* det andet program. Programmet DREJ i øvelse 3.1 kalder for eksempel programmet TREKANT.

#### Øvelse 3.5

Lær myren at tegne nedenstående figur.



#### 4. Variable.

Du har benyttet myreprogrammet TREKANT til at tegne trekanter af samme størrelse rundt omkring på skærmen, men hvis du ville tegne en trekant med en anden størrelse, blev du nødt til at rette i programmet. Det kan gøres smartere. Du kan nemlig bruge en variabel, når du vil fortælle myren, hvordan den skal tegne en trekant. Programmet kan da se sådan ud:

```
TREKANT(A)
  GENTAG(3)
    FREM(A)
      HDREJ(120)
    HERTIL
  SLUT
```

Først når du bruger programmet, fortæller du, hvor stor trekanten skal være. Hvis du for eksempel skriver TREKANT(50), så tegner myren en trekant med sidelængden 50.

#### Øvelse 4.1

Prøv det nye program TREKANT og gem det bagefter på din diskette. Hvis du i forvejen har et program, der hedder TREKANT, på din diskette, så bliver det gamle program blot smidt ud.

#### Øvelse 4.2

Lav et program, der kan tegne kvadrater i forskellig størrelse.

Der må godt være flere variable i et program. Det næste program har to variable, og det kan tegne mange forskellige figurer.

```
POLY(A,V)
  GENTAG(400)
    FREM(A)
    HDREJ(V)
  HERTIL
SLUT
```

Før du prøver dette program, skal du lige vide, at når myren tegner en figur, som du ikke gider vente på, at den skal gøre færdig, så kan du bare trykke på mellemrumstasten, da afbrydes programmet.

#### Øvelse 4.3

Prøv programmet POLY. Det dur til mere, end du måske tror.

#### Øvelse 4.4

Lav et myreprogram, der kan tegne rektangler i forskellig størrelse og med forskelligt udseende.

#### Øvelse 4.5

Programmet i denne øvelse kan benyttes til at tegne rhomber med, men der skal være et bestemt sammenhæng mellem den benyttede V og U værdi, for at det bliver en rhombe.

Prøv at benytte programmet og find dette sammenhæng.

```
RHOMBE(A,V,U)
  GENTAG(2)
    FREM(A)
    HDREJ(V)
  FREM(A)
  HDREJ(U)
  HERTIL
SLUT
```

#### Øvelse 4.6

Benyt dine erfaringer fra øvelse 4.5 til at foretage ændringer i programmet RHOMBE, således at det kun benytter de to variable A og V.

Du kan for eksempel udskifte ordren HDREJ(U) med HDREJ(beregningsudtryk), hvor beregningsudtrykket sikrer, at det altid bliver en RHOMBE. Den første linie skal du da også ændre til RHOMBE(A,V).

Afprøv den ændrede version. Gem programmet.

Ved et programkald leveres der værdier til de variable, der indgår i det program, som kaldes.

Når vi skriver

```
TREKANT(105)
```

så leverer dette kald værdien 105 til den variabel, som indgår i beskrivelsen af programmet TREKANT, altså A.

Lad NTREKANT være beskrevet ved

```
NTREKANT(B,N)
  GENTAG(N)
    TREKANT(B)
  VDREJ(360/N)
  HERTIL
  SLUT
```

Når vi skriver

```
NTREKANT(60,10)
```

så leverer vi tallet 60 til den variable B og tallet 10 til den variable N. I programmet NTREKANT kaldes programmet TREKANT, og værdien 60 overføres til den variable A i TREKANT (det sker endda 10 gange).

#### Øvelse 4.7

Prøv programmet NTREKANT.

Det er ikke altid et program tegner, som planlagt. Man kan blive overrasket, hvis man benytter samme variabel i to programmer, hvor det ene kalder det andet med et beregningsudtryk.

#### Øvelse 4.8

Afprøv først FIGUR og KVADRAT

```
FIGUR(B)          KVADRAT(A)
  KVADRAT(B/2)    GENTAG(4)
  HDREJ(90)       FREM(A)
  KVADRAT(B/2)    HDREJ(90)
  SLUT            HERTIL
                  SLUT
```

og dernæst FIG og KVADRAT

```
FIG(A)
  KVADRAT(A/2)
  HDREJ(90)
  KVADRAT(A/2)
  SLUT
```

Forklaring følger!

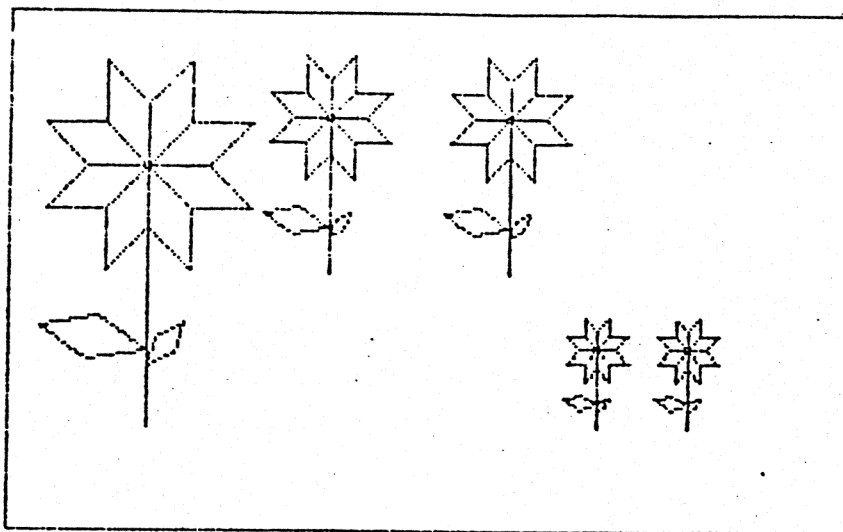
MYRESNAK findes i to versioner. I den, som du benytter i øjeblikket, er alle programmer fælles om de variable. Der findes med andre ord kun et sæt variable. Den kaldes den *globale version*. I den anden version har hvert program sit private sæt af variable, som kun kendes af dette ene program. Når to programmer her benytter variabelnavnet A, da har de to A'er intet med hinanden at gøre, det er to forskellige A'er. Denne version kaldes den *lokale version*. Den skal vi først benytte i §6.

#### Øvelse 4.9

Gør rede for forløbet af øvelse 4.7 og prøv eventuelt at benytte FIGUR og FIG med MYRESNAK (lokal version).

#### Øvelse 4.10

Lav et program, BLOMSTERHAVE, som tegner en blomsterhave som vist nedenfor. Den skal naturligvis ikke være magen til, for det er dig, der skal bestemme, hvor blomsterne skal stå, hvordan de skal se ud, hvor store de skal være og hvor mange der skal være.



### 5. Myreprogram kalder sig selv.

Du har ofte i dine beskrivelser benyttet, at et myreprogram gerne må kalde et andet myreprogram. Et myreprogram må endda gerne kalde sig selv, og det åbner mulighed for at beskrive nogle nye figurer på en let og overskuelig måde.

Hvis vi ønsker, at myren skal blive ved med at kravle 40 frem og dreje 120 til højre, har vi tidligere beskrevet denne figur ved:

```
GAMMEL
  GENTAG(1000)
    FREM(40)
    HDREJ(120)
  HERTIL
  SLUT
```

Ved at lade beskrivelsen kalde sig selv, kan den samme figur beskrives ved:

```
NY
  FREM(40)
  HDREJ(120)
  NY
  SLUT
```

Den eneste forskel er, at GAMMEL beder myren om at gøre noget 1000 gange, og NY siger, at myren skal blive ved i det uendelige.

#### Øvelse 5.1

Prøv programmet NY.

---

Ved at indføre variable får vi et program NYPOLY, der svarer til POLY fra øvelse 4.3.

```
NYPOLY(A,V)
  FREM(A)
  HDREJ(V)
  NYPOLY(A,V)
  SLUT
```

#### Øvelse 5.2

Tegn nogle af figurerne fra øvelse 4.3 med programmet NYPOLY. Tegn for eksempel nogle regulære polygoner og stjerner.

---

Ved et programkald leveres der, som vi tidligere har set, værdier til de variable, som indgår i det program, der kaldes. Når et myreprogram kalder sig selv, bevirker det, at vi har mulighed for at ændre på værdien af en variabel.

Myreprogrammet SPIRAL kalder sig selv med et udtryk "H+2" og ændrer derved på værdien af den variable H hver gang, der udføres et kald.

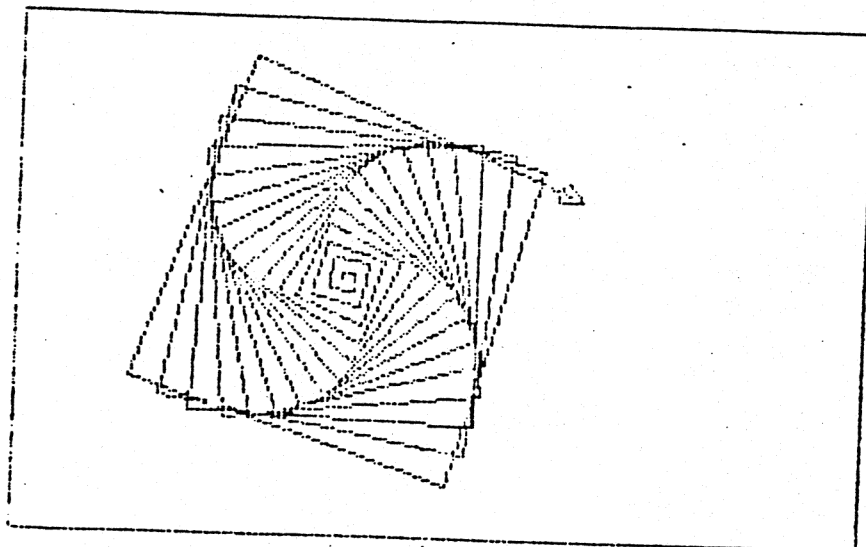
```
SPIRAL(H)
  FREM(H)
  HDREJ(90)
  SPIRAL(H+2)
SLUT
```

### Øvelse 5.3

Prøv programmet SPIRAL.

### Øvelse 5.4

Foretag en ændring i myreprogrammet SPIRAL, således at det kan tegne en spiral, der minder om den nedenfor.

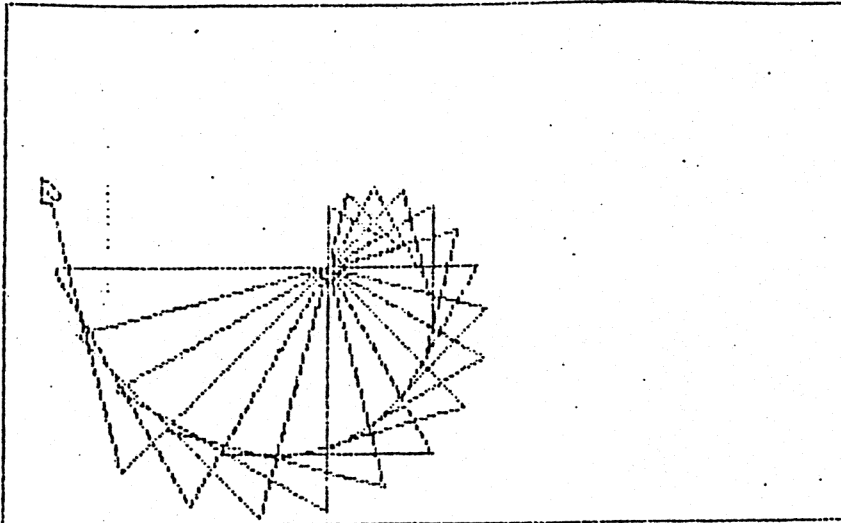


### Øvelse 5.5

Indfør en variabel V for drejningsvinklen i programmet SPIRAL og eksperimentér med forskellige spiraler. Den variable V skal indgå både i linie 1, linie 3 og linie 4.

### Øvelse 5.6

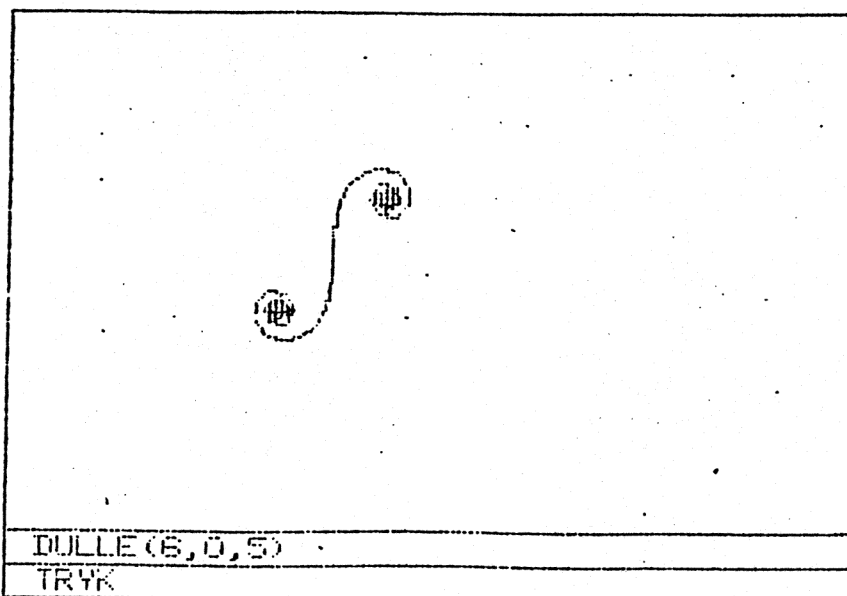
Lav et program, der drejer en trekant rundt, idet trekanten bliver større og større. Tryk på mellemrumstasten, når du synes figuren er pæn, for ellers bliver myren jo ved at kravle i det uendelige. Prøv om du kan lave en figur, der minder om den nedenfor.



### Øvelse 5.7

Programmet DULLE laver nogle ejendommelige figurer. Hvis du skriver DULLE(6,0,5), får du figuren nedenfor. Eksperimentér med programmet, idet du i begyndelsen holder dig til tilfældet  $V=0$ , og bagefter studerer tilfældet  $V=10$ .

```
DULLE(A,V,T)
  FREM(A)
  HDREJ(V)
  DULLE(A,V+T,T)
  SLUT
```





## 6. Tilbagevikling (MYRESNAK lokal version).

Når et myreprogram kalder sig selv, siger vi, at beskrivelsen er rekursiv. Vi har hidtil kun arbejdet med uendelige rekursive beskrivelser, hvor den eneste måde at stoppe myren på var at trykke på mellemrumstasten. I dette afsnit vil vi se på, hvorledes det er muligt at lave endelige rekursive beskrivelser. I det følgende er alle variable lokale, det vil sige, at enhver variabel er privat og kun kendes af det program, som den tilhører. Hvis to programmer begge anvender variabelnavnet A, da har de to A'er intet med hinanden at gøre.

Når en rekursiv beskrivelse kalder sig selv, da bevirker hvert kald, at der oprettes et nyt sæt variable. Et rekursivt kald er derfor at betragte som ethvert andet programkald. Man forlader det kaldende program og går ind i det kaldte, som her blot er identisk med det kaldende.

Lad os se på følgende rekursive myreprogram.

```
HAK(A,T)
  HVIS(A=0)SLUT
  FREM(A)
  HDREJ(90)
  HAK(A-T,T)
  FREM(A)
  HDREJ(90)
  SLUT
```

Hver gang der udføres et kald, vil vi sige, at vi går et niveau "ud". Beskrivelsens virkemåde kan da illustreres ved niveau-billeder, som vist nedenfor.

Niveau-billeder.

niveau=0	HAK(40,10)	fremvikling	tilbagevikling
niveau=1  $\begin{array}{r l} A & 40 \\ \hline T & 10 \end{array}$	HAK(A,T) HVIS(A=0) SLUT FREM(A) HDREJ(90) HAK(A-T,T) FREM(A) HDREJ(90) SLUT		
niveau=2  $\begin{array}{r l} A & 30 \\ \hline T & 10 \end{array}$	HAK(A,T) HVIS(A=0) SLUT FREM(A) HDREJ(90) HAK(A-T,T) FREM(A) HDREJ(90) SLUT		
niveau=3  $\begin{array}{r l} A & 20 \\ \hline T & 10 \end{array}$	HAK(A,T) HVIS(A=0) SLUT FREM(A) HDREJ(90) HAK(A-T,T) FREM(A) HDREJ(90) SLUT		
niveau=4  $\begin{array}{r l} A & 10 \\ \hline T & 10 \end{array}$	HAK(A,T) HVIS(A=0) SLUT FREM(A) HDREJ(90) HAK(A-T,T) FREM(A) HDREJ(90) SLUT		
niveau=5  $\begin{array}{r l} A & 0 \\ \hline T & 10 \end{array}$	HAK(A,T) HVIS(A=0) SLUT FREM(A) HDREJ(90) HAK(A-T,T) FREM(A) HDREJ(90) SLUT		

Øvelse 6.1

Prøv programmet HAK og se, om myren tegner som angivet.

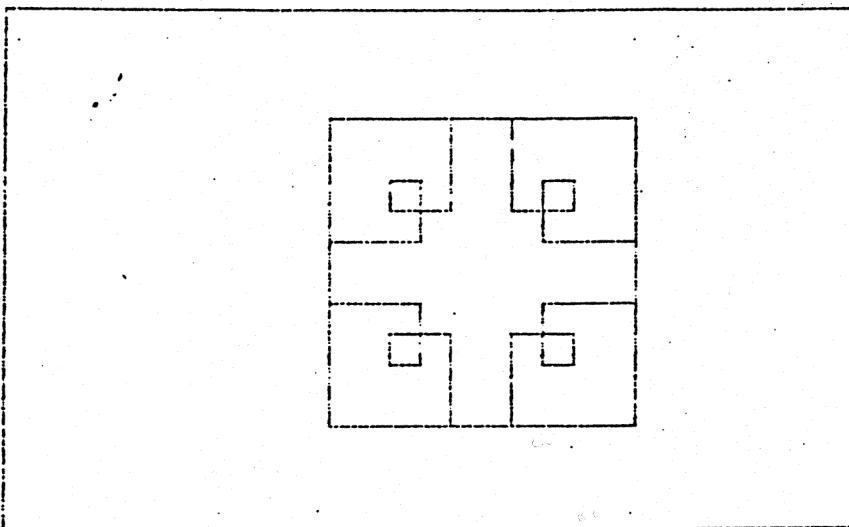
Øvelse 6.2

Tag en blyant og lav den tegning, som HAK ville tegne, hvis du skrev HAK(50,10). Benyt niveau-billeder.

Lad derpå myren tegne figuren.

Øvelse 6.3

Lav en beskrivelse for tegningen nedenfor.



---

En betinget ordre, som den vi brugte i myreprogrammet HAK, kan vi også have glæde af, når vi blot ønsker at stoppe rekursionen. Der behøver ikke at blive tegnet noget under tilbageviklingen.

Øvelse 6.4

Lav en endelig rekursiv beskrivelse af figuren fra øvelse 5.6. Indfør en variabel, som holder styr på niveauerne, og benyt denne variabel til at stoppe tegningen, når niveau 12 er nået.

En rekursiv beskrivelse som HAK går en række niveauer ud og vender derpå pænt hjem. HAK vikler kun frem og tilbage én gang. I en rekursiv beskrivelse kan der godt være adskillige frem- og tilbageviklinger. Det er der for eksempel, hvis det rekursive kald er anbragt inde i en sløjfe, som i myreprogrammet FIRKANTER.

```

FIRKANTER(A,N)
  HVIS(N=0) SLUT
  GENTAG(4)
    FIRKANTER(A/3,N-1)
  FREM(A)
  HDREJ(90)
  HERTIL
  SLUT

```

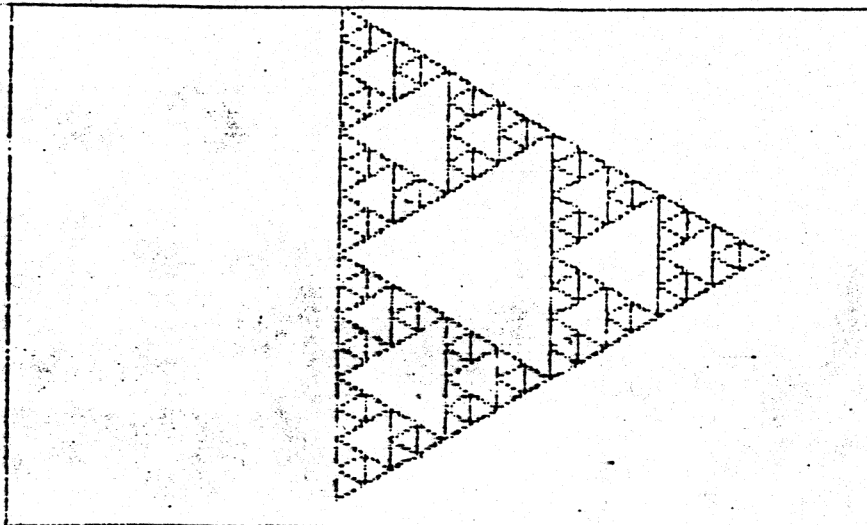
#### Øvelse 6.5

Prøv programmet FIRKANTER med 1,2,3 og 4 som værdier for N. Bliv ved med at gentage tegningen for hver N-værdi, indtil dine tanker er foran myrens beyægelser.

Tegn niveau-billeder svarende til værdierne N=2 og A=45. Her får du brug for en sløjfe-tæller på hvert niveau.

#### Øvelse 6.6

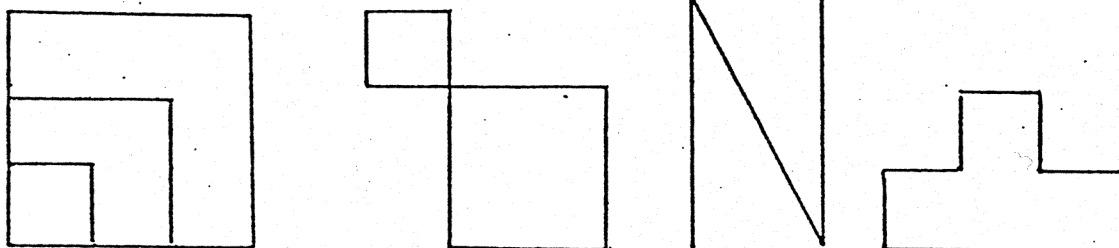
Lav et program TREKANTER, der tegner figuren nedenfor.



## OPGAVER

Opgave 1 (\$1)

Lad myren tegne følgende figurer:

Opgave 2 (\$5)

Indtast følgende myreprogram:

```
CIRKEL(A,V)
  FREM(A)
  HDREJ(V)
  CIRKEL(A,V)
  SLUT
```

Skriv: CIRKEL(4,4) - tryk på mellemrumstasten, når cirklen er sluttet.

```
START
HDREJ(90)
ERYM
```

Find nu cirkelns diameter målt i myreskridt ved at gå frem, indtil du møder cirkelperiferien. Hvis du kommer til at overskride cirkelperiferien og ønsker at bakke, så husk at bakke med farven sort.

Skriv derpå:

```
SORT
BAK(diameteren)
VDREJ(90)
HVID
MYRE
```

Prøv nu at tegne en cirkel med halvt så stor diameter.

Tegn også en cirkel med en diameter, som er  $\frac{3}{4}$  af den oprindelige.

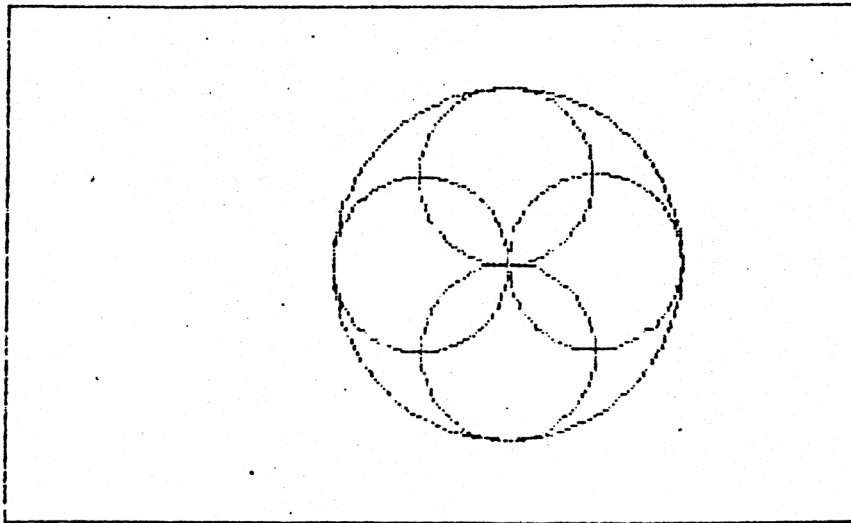
Opgave 3 (\$5, opgave 2)

Lav et myreprogram, der har en enkelt variabel  $R$ , og som tegner en cirkel med radius  $R$ .

---

Opgave 4 (\$5, opgave 2)

Lav en beskrivelse af figuren nedenfor.

Opgave 5 (\$5)

Eksperimentér med programmet

```
POLYGON(A,V)
FREM(A)
HDREJ(V)
FREM(A)
HDREJ(2*V)
POLYGON(A,V)
SLUT
```

---

Opgave 6 (\$5)

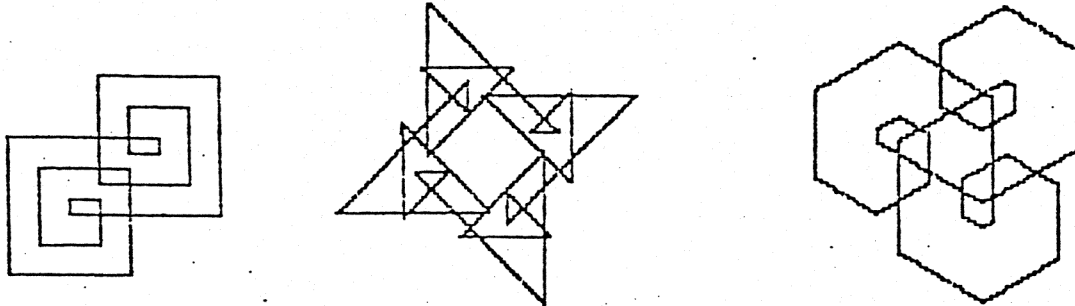
I øvelse 5.7 tegnede du figurer med programmet DULLE. Find for tilfældet  $V=0$  et eksplicit udtryk for antallet af "duller" på figuren, som funktion af vinkeltilvæksten  $T$ .

---

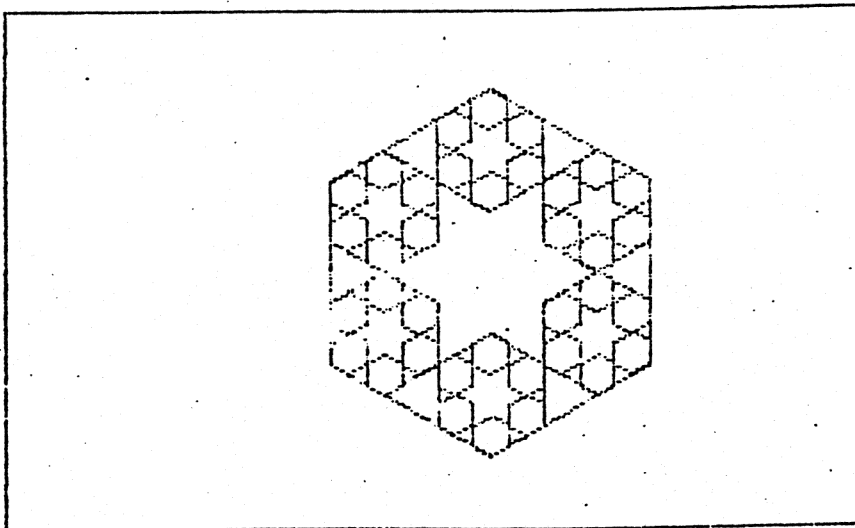
Opgave 7 (\$6)

Nedenstående tre figurer kan alle beskrives ved et program A, som kalder et rekursivt program B. De to programmer kan tegne en hel familie af beslægtede figurer.

Lav beskrivelsen og tegn figurerne nedenfor, samt mindst to nye.

Opgave 8 (\$6)

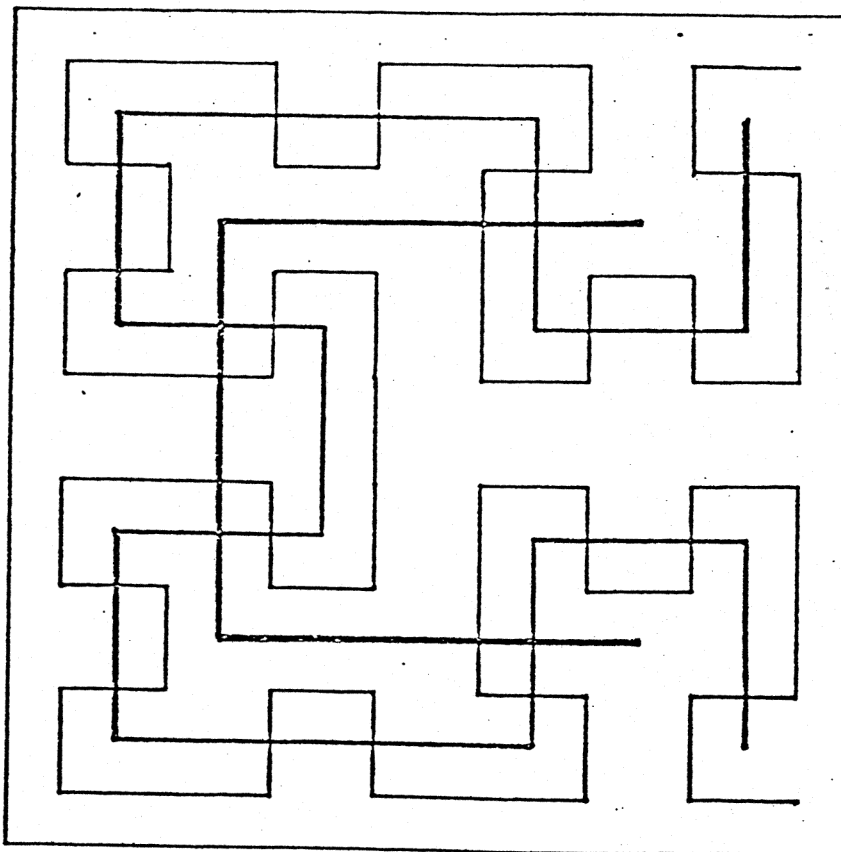
Lav et rekursivt myreprogram, der kan tegne nedenstående figur.



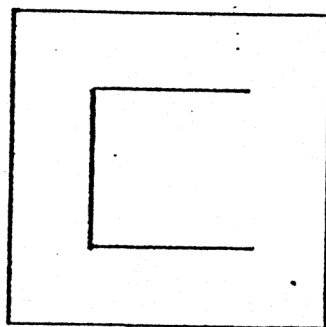
Opgave 9 (§6)

Et klassisk emne i matematikken er at konstruere en konvergent følge af kurver, hvor grænsekurven udfylder hele enhedskvadratet. Den mest kendte er vel nok Peanos kurve. Følgen af kurver, som vi vil studere her, skyldes den tyske matematiker Hilbert.

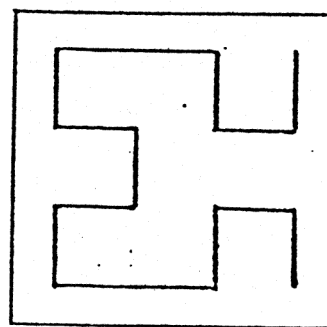
Nedenfor er tegnet de første Hilbert-kurver. Lav et program, som kan tegne en vilkårlig kurve i denne følge.



Hilbert<sub>1</sub>, Hilbert<sub>2</sub> og Hilbert<sub>3</sub>



Hilbert<sub>1</sub>

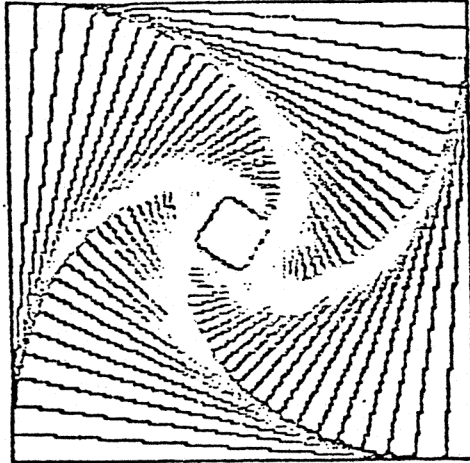


Hilbert<sub>2</sub>



Opgave 10 (§6)

Lav et myreprogram, der kan tegne nedenstående figur.



Lav et program, der tegner en tilsvarende figur, men blot benytter sekskanter i stedet for kvadrater.

---