

VIGGO SADOLIN

PROGRAMMERING AF REGNEPROCESSER  
I  
SKOMAL

Matematisk Institut  
DLH  
1983

## Forord

Programmeringssproget SKOMAL er specielt udviklet til beskrivelse af regneprocesser i skolens matematikundervisning.

Brugerfladen i SKOMAL afviger i høj grad fra de gængse generelle programmeringssprog. Programmeringsmiljøet er det samme som i MYRESNAK.

Der er i SKOMAL taget hensyn til, at eleven ikke daglig skriver programmer. Notationen er tilpasset skolens matematikundervisning. Det er i SKOMAL muligt at udføre et program ordre for ordre, mens man iagttager, hvorledes lagerbilledet dynamisk skifter udseende. Eleverne får herved et værktøj til sporing af fejl, og at lave fejl bliver derved et konstruktivt led i problemløsningsprocessen. Formålet har været at udvikle et let-tilgængeligt sprog, hvor eleverne ubesværet får adgang til at opleve og eksperimentere med regneprocesser, algoritmer, variable og funktioner og derved tilegne sig større indsigt i disse vigtige begreber og deres anvendelse.

For at kunne benytte denne bog er det naturligvis nødvendigt at have adgang til en mikrodatamat, men det er også nødvendigt at have en SKOMAL-fortolker, som er udviklet til den pågældende mikrodatamat. Der foreligger ved udgivelsen af denne bog SKOMAL-fortolkere til mikrodatamaterne: APPLE IIe og PICCOLINE. Aktuelle oplysninger om SKOMAL-fortolkere til skolemarkedets mikrodatamater kan fås ved skriftlig henvendelse til forlaget.

## Indhold

### Indledning

1. Ordre
2. Programmer
3. Modtager- og afleveringsvariable
4. Betingede ordre
5. Program kalder program

### Opgaver

### Oversigt

### Stikordsregister

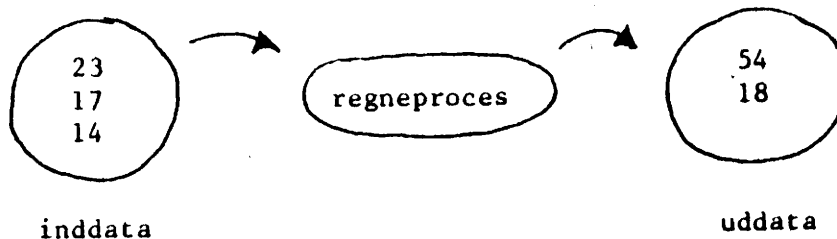
## Indledning.

Hvis vi ønsker at beregne summen samt gennemsnittet af tallene 23, 17 og 14, kan udregningen for eksempel forløbe således:

$$\begin{array}{r} 23 \\ 17 \\ + 14 \\ \hline 54 \end{array} \qquad \begin{array}{r} 54:3 = 18 \\ \underline{3} \\ 24 \\ \underline{24} \\ 0 \end{array}$$

$$\text{sum} = 54 \qquad \text{gennemsnit} = 18$$

Vi har her udført en *regneproces*. Vort udgangspunkt var tallene 23, 17 og 14, som vi vil kalde for *inddata* for regneprocessen. Resultatet af processen er tallene 54 og 18, som vi vil kalde for *uddata* for processen.



For at være i stand til at udføre regneprocessen må vi råde over nogle beskrivelser af en eller anden slags. Mange er i stand til at udføre regneprocessen ganske automatisk, måske endda mens de samtidig laver noget helt tredje. De har de nødvendige opskrifter lagret i de "grå celler" og kan hente dem frem og benytte dem, når de har brug for det. Det er dog ikke alle, der går rundt med indbyggede beskrivelser for netop denne regneproces. Det er således ikke alle, der uden videre kan udføre regneprocessen, som er beskrevet ved:

- (1) Find sum og gennemsnit af tallene 23, 17 og 14.

Vi kan imidlertid lave en beskrivelse af regneprocessen, som formodentlig næsten alle kan følge:

- (2)
1. Find summen af tallene 23, 17 og 14 ved at lægge de tre tal sammen.
  2. Find gennemsnittet af tallene ved at dividere summen af tallene med 3.

En beskrivelse, der angiver, hvorledes en proces skal udføres, kaldes for en *algoritme*. Den sidste algoritme er blevet mere detaljeret, og vi siger, at den beskriver processen på et *lavere niveau*. Man behøver ikke så me-

gen indbygget viden for at kunne benytte den, men den forudsætter alligevel, at man kan udføre visse elementære operationer "på egen hånd". Processen kunne udmærket beskrives på et endnu lavere niveau.

En helt anden algoritme for samme regneprocess kunne være:

$$(3) \quad \begin{aligned} \text{SUM} &\leftarrow +(+ (23, 17), 14) \\ \text{GENNEMSNI} &\leftarrow \div (\text{SUM}, 3) \end{aligned}$$

Denne algoritme beskriver processen på stort set samme niveau som (2), men det betyder ikke nødvendigvis, at lige så mange vil være i stand til at anvende den. Ganske mange vil nok synes, at (3) er helt uforståelig.

En beskrivelse af regneprocessen kunne også have en udformning, der er rettet mod et regneteknisk hjælpemiddel for eksempel en lommeregner. Nedenfor ses en *taste-algoritme* for regneprocessen.

$$23 \boxed{+} 17 \boxed{+} 14 \boxed{=} \quad (\text{summen vises i lystavlen})$$

$$\boxed{\div} 3 \boxed{=} \quad (\text{gennemsnittet vises i lystavlen})$$

Heller ikke alle vil være i stand til at benytte denne algoritme. Den er ganske vist detaljeret, men kræver en form for speciel viden om, hvorledes man læser taste-algoritmer.

Når man taler til en person, må man gøre sig klart hvilket niveau, man kan benytte for at blive forstået, og man må vælge et sprog eller dialekt, som den pågældende forstår. Det er vist et ganske almindeligt og anerkendt princip, og det gælder naturligvis også, når vi laver beskrivelser af regneprocesser, som andre skal udføre.

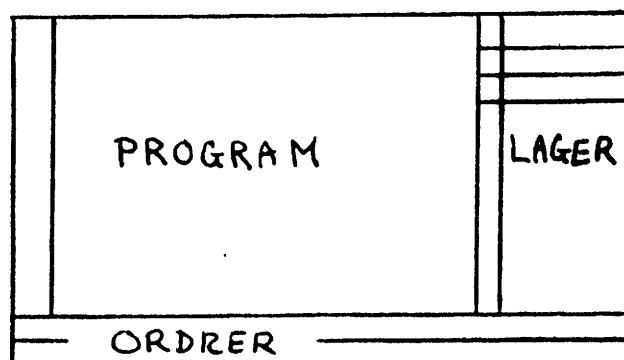
Vi står over for helt samme problemer, når vi skal meddele os til en datamaskine. Vi må udforme algoritmen så detaljeret, at datamaskinen kan udføre den, og vi må benytte et sprog, som datamaskinen forstår. Et sprog, som datamaskinen forstår, kaldes for et *programmeringssprog*, og en algoritme, som er udformet i et programmeringssprog, kaldes et *program*. Programmeringssproget fastlægger rammerne for, hvor detaljeret algoritmen skal udformes, for at datamaskinen kan udføre den. Vi vil i dette materiale benytte programmeringssproget SKOMAL.

## 1. Ordre.

For at få indsigt i det niveau SKOMAL-sproget givet mulighed for at benytte, er der i SKOMAL indbygget en datamaskinmodel bestående af:

- (a) Et sted hvor programmet befinder sig.
- (b) Et sted, *lageret*, hvor alle mellemregninger kan gemmes og senere hentes og bruges.

Modellen ser således ud:



Før vi begynder at opbygge programmer, vil vi se lidt på, hvorledes lageret fungerer. Lageret består af en række hylder, som kan navngives ved hjælp af bogstaver. På hylderne kan vi anbringe tal. Vi kan hente tal på hylder, foretage udregninger og anbringe resultatet af udregningerne på en hylde i lageret. Alle sådanne lageroperationer foregår i SKOMAL ved hjælp af *ordrer*.

Hvis vi ønsker at oprette en hylde med navnet A og lægge et tal ind på denne hylde, kan vi benytte ordren IND(A). Den bevirker, at der nederst på *lagersiden* skrives A=, hvorpå vi kan indtaste det tal, som skal anbringes på hylde A. Hvis vi ønsker at oprette flere hylder med tal på, kan det gøres i én ordre. Ordren IND(A,B,C) virker på samme måde som de tre ordrer IND(A), IND(B) og IND(C).

### Øvelse 1.1 Start SKOMAL.

Skriv LAGER (efterfulgt af tryk på RETURN-tasten), og du befinder dig på lagersiden.

Skriv IND(A) og anbring tallet 17 på hylde A.

Skriv IND(X,Y) og anbring 314.7 på hylde X og -2.438 på hylde Y.

Ved hjælp af en *tildelingsordre* kan vi hente tal på hylde, bruge tallene i en udregning og til sidst gemme resultatet på en hylde. Ordren  $A:=B+C$  aflæser indholdet af hylde B og C, lægger disse to tal sammen og gemmer resultatet på hylde A.

Øvelse 1.2 Skriv NUL, og lageret nulstilles.

Anbring 23 på hylde H og 13.8 på hylde F.

Skriv  $G:=H+F$  og se, hvad der sker.

Skriv  $I:=G+1$

Skriv  $A:=G+1$

Skriv  $B:=-A$

Skriv  $C:=123.84$

Skriv  $D:=-A+15$

I SKOMAL skrives de fire regnetegn (+, -, ·, :) sådan (+, -, \*, /). I en tildelingsordre må der højst benyttes ét regnetegn i udtrykket til højre for *tildelingstegnet* ':='.

Når vi gemmer et tal på en hylde, hvor der i forvejen ligger et tal, da skubbes det gamle tal ud og er tabt, og kun det nye tal ligger der.

Øvelse 1.3 Nulstil lageret.

Læg tallene 1,5, -10 på hylde A, B og C.

Skriv  $A:=A+1$

Skriv  $A:=A+1$

Skriv  $A:=A+1$

Skriv  $A:=2*A$

Skriv  $A:=A/2$

Skriv  $A:=B+C$

Skriv  $B:=B-C$

Skriv  $A:=0$

Øvelse 1.4 Eksperimentér selv med lageret.

Øvelse 1.5 Anbring 3 på hylde X og 5 på hylde Y.

Ombyt indholdet af hylde X og Y uden at benytte tal, men kun navne på hylde.

5 - (-10)

Ved at skrive NORMAL kommer vi tilbage til den side, vi begyndte på ved starten af SKOMAL. På denne side, *normalsiden*, kan vi ikke se, hvad der ligger på hylderne i lageret. De ordrer, vi hidtil har benyttet fungerer fortsat, men hvis vi på et tidspunkt ønsker at vide, hvad der ligger på hylde B, må vi bede om at få tallet udskrevet. Det sker ved ordren UD(B). Indholdet af hylderne H,G og F udskrives ved ordren UD(H,G,F).

Øvelse 1.6 Eksperimentér på normalsiden med IND-ordren, Ud-ordren og tildelingsordren.

Når vi er på lagersiden og udfører en UD-ordre, kan man ikke se noget, men ordren er blevet udført med udskrift på normalsiden. Sideskift mellem normalsiden og lagersiden bevirker, at lageret nulstilles. På normalsiden udskrives tal med 2 decimaler ved opstart af SKOMAL, men dette kan ændres til et antal decimaler mellem 1 og 6. Ved at skrive DEC(4) skiftes til udskrift med 4 decimaler.

Øvelse 1.7 Afprøv DEC.

## 2. Programmer.

I det foregående afsnit har vi ved hjælp af ordrer fået datamaskinen til at foretage udregninger og flytte rundt på tal. Når vi afgav en ordre, så udførte datamaskinen straks ordren, men når ordren var udført, så kunne den ikke benyttes igen. Datamaskinen husker ikke ordrene, således som vi hidtil har benyttet dem.

Et sæt af ordrer kan imidlertid samles til en enhed og navngives, og da vil datamaskinen huske denne enhed. Enheden vil vi kalde for et program.

Et program består altså af et sæt af ordrer, som fastlægger en regneprocess.

Et program, der beregner summen af tre tal, kan se sådan ud

```
SUM
  IND(A,B,C)
  S:=A+B
  S:=S+C
  UD(S)
SLUT
```

Programmet består her af fire ordrer, programnavnet SUM samt det afsluttende ord SLUT.

Øvelse 2.1 Skriv HUSK og indtast programmet.

Hvis der er indtastningsfejl, så skriv RET og ret fejlene.

Skift til lagersiden.

Skriv SUM.

Du ser nu programmet SUM og en pil ud for den første ordre.

Ved et enkelt tryk på mellemrumstasten får du udført den ordre, der står ud for pilen, og pilen vil, når ordren er udført, rykke frem til næste ordre.

Udfør programmet indtil pilen står ud for SLUT.

Endnu et tryk på mellemrumstasten bevirker, at vi forlader programmet og dets lagerbillede og returnerer til det lagerbillede vi havde, da vi skrev SUM.

Øvelse 2.2 Skift til normalsiden og udfør programmet SUM.

Øvelse 2.3 Udbyg programmet SUM, således at også gennemsnittet beregnes og udskrives. Skift programnavn til BEREKN.

Afprøv programmet BEREKN både på lagersiden og normalsiden.

Øvelse 2.4 Lav et program, der indlæser radius R og beregner og udskriver cirkelens omkreds og areal.

Øvelse 2.5 Lav et program, der indlæser x og beregner og udskriver  $f(x)$ , hvor  $f(x) = x^2 - 1$ .

Du får en oversigt over dine programmer ved at skrive KATALOG, og hvis du ønsker at se et program, skal du blot skrive LIST. Der er kun plads til et begrænset antal programmer i datamaskinen. Hvis du ønsker at skabe plads, kan du slette programmer ved at skrive GLEM. 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, når du har skrevet HENT. Alle de programmer, der ligger i datamaskinen (altså ikke dem på disketten) slettes, når du stopper SKOMAL ved at skrive STOP eller slukker for datamaskinen.

Programmet fra øvelse 2.5 kan for eksempel se sådan ud:

```
FKT
  IND(X)
  Y:=X*X
  Y:=Y-1
  UD(Y)
SLUT
```

Hvis vi ønsker at beregne flere funktionsværdier, for eksempel ti, så er det ret besværligt at skulle skrive FKT ti gange, og det kan da også gøres smartere.

En måde at gøre det på er at indramme det afsnit i programmet, som ønskes gentaget, med ordene GENTAG og HERTIL. Programmet ser da sådan ud:

```
MANGEFKT
  GENTAG(10)
    IND(X)
    Y:=X*X
    Y:=Y-1
    UD(Y)
  HERTIL
SLUT
```

Tallet i parentesen efter GENTAG angiver, hvor mange gange ordrene mellem GENTAG og HERTIL skal gentages.

Øvelse 2.6 Afprøv programmet MANGEFKT både på lagersiden og normalsiden.

Øvelse 2.7 Lav et program, der udskriver en tabel over funktionen  $g(x) = x(x-3)$  for  $x = 1, 2, 3, \dots, 12$ .  
Afprøv programmet både på lagersiden og normalsiden.

Øvelse 2.8 Lav et program, der beregner summen af de første 15 naturlige tal.  
Afprøv programmet både på lagersiden og normalsiden.

En programudførelse kan afbrydes ved at holde S-tasten nede, indtil datamaten siger 'BIB', og derpå trykke på mellemrumstasten.

### 3. Modtager- og afleveringsvariable.

Vi har hidtil benyttet ordren IND til at levere inddata til et program. Det er blot én måde at gøre det på. Vi kan også anvende den skrivemåde, vi plejer at bruge, når vi arbejder med funktioner i matematik.

I stedet for programmet FKT kunne vi udforme et program sådan:

```
FUNK(X)
  Y:=X*X
  Y:=Y-1
  UD(Y)
SLUT
```

Programmet FUNK benytter modtagervariablen X. Inddata modtages gennem den variable X. Når vi skriver FUNK(3), bevirker det, at vi får udregnet og udskrevet funktionsværdien for  $x=3$ .

Øvelse 3.1 Afprøv programmet FUNK både på lagersiden og normalsiden.

Øvelse 3.2 Lav et program, der beregner og udskriver summerne:

```
1
1+3
1+3+5
1+3+5+7
:
:
1+3+5+7+...+(2n+1)
```

Når vi i et program har brug for flere tal som inddata, kan det i parentes efter programnavnet angives ved en *variabelliste* (variable adskilt af komma). Et program, der beregner produktet af to naturlige tal a og b ved hjælp af addition, kan da se således ud:

```
PRODUKT(A,B)
  P:=0
  GENTAG(B)
    P:=P+A
  HERTIL
  UD(P)
SLUT
```

Øvelse 3.3 Afprøv programmet PRODUKT med forskellige værdier for inddata.

Øvelse 3.4 Lav et program POTENS, der udregner  $x^y$ , hvor x er et vilkårligt reelt tal, og y er et naturligt tal.

Øvelse 3.5 Hvis du har slettet programmet FUNK, så indtast det igen.

Skift til lagersiden.

Anbring tallet 7 på hylde H.

Skriv FUNK(2) og udfør programmet (også SLUT).

Fortæller lagerbilledet, hvor der på hylde H ligger tallet 7, dig noget om uddata fra programmet FUNK?

Som du erfarede i øvelse 3.5 findes der to lagerbilleder. Lagerbilledet, som vises, når vi beder om at få et program udført, kalder vi niveau 0 billedet. Lagerbilledet, som vises, når programmet udføres, kalder vi niveau 1 billedet. Lageret på niveau 1 nulstilles automatisk, når programudførelsen er afsluttet. Hidtil har vi benyttet UD-ordren til at levere uddata fra et program. UD-ordren bevirker, at uddata skrives på normalsiden, og det er da ligegyldigt om programmet er sendt til udførelse fra lagersiden eller normalsiden. Det er muligt at få uddata fra et program gemt i lageret på niveau 0. Det gøres ved i programmet at angive en *afleveringsvariabel* i parentes efter programnavnet. I programmet FUNKTION er Y angivet som afleveringsvariabel, således at den beregnede funktionsværdi overføres til lagerbilledet på niveau 0.

```
FUNKTION(X;Y)
  Y:=X*X
  Y:=Y-1
SLUT
```

Øvelse 3.6 Nulstil lageret på niveau 0. Skriv på lagersiden ordren

FUNKTION(2;E) og udfør programmet.

Fortsæt uden at nulstille lageret på niveau 0 med

```
FUNKTION(3;Y)
  A:=4
  FUNKTION(A;B)
  FUNKTION(A;A)
  N:=2
  M:=3
  FUNKTION(N+M;L)
  Eksperimentér selv videre.
```

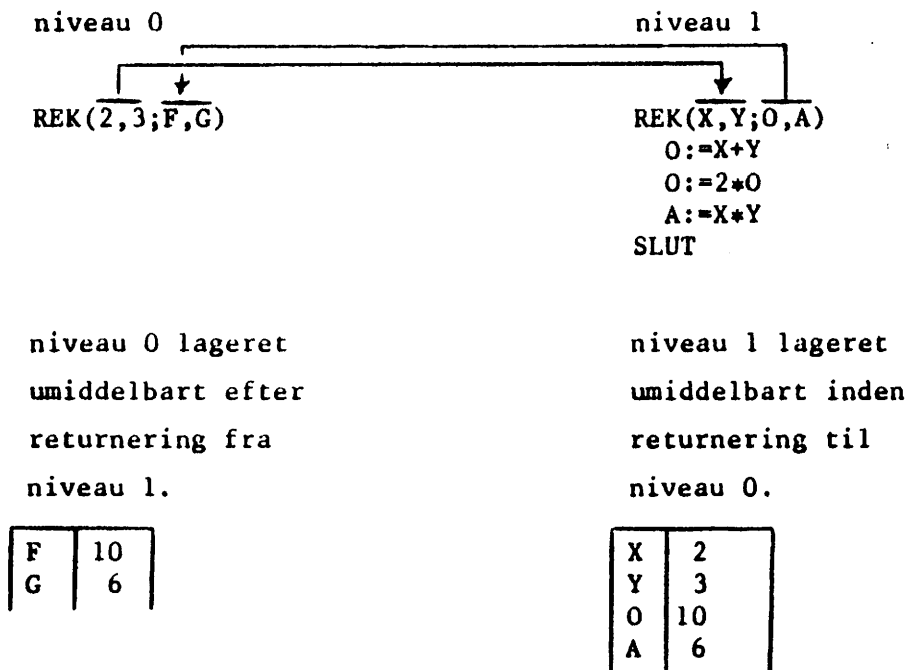
Hvis der er brug for flere afleveringsvariable angives de adskilt af komma. Et program, der for et rektangel bestemmer omkreds og areal ved hjælp af sidelængderne, kan se sådan ud:

```
REK(X,Y,O,A)
  O:=X+Y
  O:=2*O
  A:=X*Y
SLUT
```

Øvelse 3.7 Afprøv programmet REK på lagersiden og på normalsiden.

Øvelse 3.8 Løs øvelse 2.4 ved at benytte to afleveringsvariable.

Betegnelserne modtagervariable og afleveringsvariable benyttes om programmets variable og understreger, hvad de variable i programmet bruges til i forbindelse med inddata og uddata i programmet. Når vi skal have udført et program og skriver et *programkald*, da vil tallene til venstre for semikolon i programkaldet afleveres til programmets modtagervariable, og de variable til højre for semikolon i programkaldet vil modtage værdier fra programmets afleveringsvariable.



Øvelse 3.9 Lav et program CYLINDER, der ud fra højden samt radius i grundcirklen beregner overfladen og rumfanget af en cylinder. Programmet skal både udskrive uddata på normalsiden og levere uddata til lageret på niveau 0. Afprøv programmet på både lagersiden og normalsiden.

#### 4. Betingede ordrer.

Udførelse af enhver ordre kan gøres betinget af et udsagn.

Den *betingede ordre*

`HVIS(N=0)A:=B+1`

vil bevirke, at ordren `A:=B+1` kun bliver udført, hvis udsagnet `N=0` er sand.

Udsagn kan opbygges ved hjælp af tegnene:

= lig med  
 <> forskellig fra  
 < mindre end  
 > større end  
 <= mindre end eller lig med  
 >= større end eller lig med

Der må gerne indgå regneudtryk i opbygningen af et udsagn, for eksempel  $A+B \leq H-1$ .

Øvelse 4.1 Lav et program, der beregner funktionsværdien for funktionen  $h$ , hvor  $h(x) = |x|$ .

Øvelse 4.2 Lav et program, der beregner funktionsværdien for funktionen  $s$ , hvor  $s$  er defineret ved

$$s(x) = \begin{cases} -1 & \text{for } x < 0 \\ 0 & \text{for } x = 0 \\ 1 & \text{for } x > 0 \end{cases}$$

Øvelse 4.3 Lav et program, der som inddata har to reelle tal, og som uddata har det største af disse to tal.

Det er tilladt at benytte en betinget afslutning som programordre. Ordren

HVIS( $K > 0$ )SLUT

bevirker, at programudførelsen afsluttes, hvis udsagnet  $K > 0$  er sand.

Lige som det er muligt, at afslutte en programudførelse, er det også muligt at afslutte en gentagelse. Ordren

HVIS( $E < 0.01$ )UDHOP

bevirker, at den gentagelse, hvori ordren er anbragt, afsluttes, og ordren umiddelbart efter gentagelsen er den næste, der udføres. En gentagelse, hvori der er anbragt et UDHOP, behøver ikke at have en antals-styring efter ordet GENTAG. Der må gerne være både antals-styring og UDHOP, og der må endda gerne være flere UDHOP.

Følgende program bestemmer, hvor mange led, man skal tage med i summen  $1+2+3+4+\dots$  for at summen bliver mindst 1000.

```

ANTAL
  S:=0
  N:=1
  GENTAG
    S:=S+N
    HVIS(S>=1000)UDHOP
    N:=N+1
  HERTIL
  UD(N,S)
SLUT

```

Øvelse 4.4 Afprøv programmet ANTAL både på lagersiden og normalsiden.

Øvelse 4.5 Lav et program, der finder hvor mange 2-taller, man skal gange sammen for at få et produkt, der er større end 10000.

Øvelse 4.6 Lav et program, der udskriver FIBONACCI-tallene, der er mindre end 1000.

$$F_0=0, F_1=1$$

$$F_i=F_{i-1}+F_{i-2} \quad \text{for } i \geq 2.$$

5. Program kalder program.

Et program må gerne benytte et andet program. I et program, må der altså gerne være anbragt programkald. Nedenfor kalder programmet TABEL2 et andet program, nemlig FUNKTION.

```

TABEL2
  IND(A,H,N)
  X:=A
  GENTAG(N+1)
    FUNKTION(X;F)
    UD(X;F)
    X:=X+H
  HERTIL
SLUT

FUNKTION(X;Y)
  Y:=X*X
  Y:=Y-1
SLUT

```

Øvelse 5.1 Afprøv de to programmer på lagersiden og normalsiden.

Der findes ikke blot et lager på niveau 0 og niveau 1. Ethvert programkald bevirker, at man går ud på et nyt niveau med eget lager, og der bliver man,

indtil en SLUT-ordre sender én tilbage til det niveau, hvor programkaldet fandt sted. Ethvert niveau har sit eget private lager, så det er ikke bare niveau 0 og niveau 1, som har et eget lager. SKOMAL er blot lavet sådan, at når et program sendes til udførelse fra lagersiden, da ser vi kun lagerbillederne på niveau 0 og niveau 1.

Øvelse 5.2 Benyt programmet TABEL2 til at tabellægge funktionen  $g$ , hvor

$$g(x) = x^2 - 3x + 2$$

Benyt  $A = -1$ ,  $H = 0.3$ ,  $N = 13$ .

Øvelse 5.3 Benyt TABEL2 til at tabellægge funktionen  $h$  fra øvelse 4.1. Benyt  $A = -2$ ,  $H = 0.5$ ,  $N = 7$ .

Øvelse 5.4 Lav et program, der for to naturlige tal udfører heltalsdivision. Inddata skal være to naturlige tal. Uddata skal returneres i afleveringsvariable og bestå af kvotient og rest. Programmets første linie kan se sådan ud

DIV(N,D;Q,R)

Øvelse 5.5 Lav et program EUKLID, der udfører Euklids algoritme. Benyt programmet DIV fra øvelse 5.4.

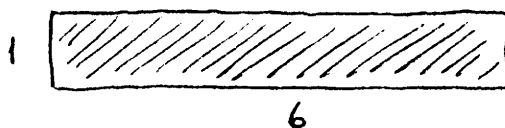
Øvelse 5.6 Lav et program, der for et givet naturligt tal, større end 1, afgør om det er et primtal. Programmet skal udskrive tallet, hvis det er et primtal. Benyt små tal som inddata.

Øvelse 5.7 Lav et program, der løser to ligninger med to ubekendte.

## Opgaver

Opgave 1 Til beregning af kvadratroden af et tal kan benyttes følgende metode:

Lad os beregne  $\sqrt{6}$ . Tallet  $\sqrt{6}$  er sidelængden i kvadratet med arealet 6. Vi ser først på det rektangel, som har sidelængderne 1 og 6. Dette rektangel har også arealet 6, men det er ikke et kvadrat. Vi vil nu forsøge at lave rektangler, der mere og mere ligner et kvadrat. Alle de rektangler, vi vil se på, skal have arealet 6. Vi starter som sagt med det rektangel, som har sidelængderne 1 og 6.



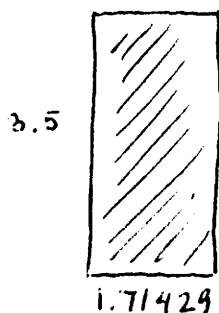
Hvis vi sammenligner det med kvadratet med samme areal, da er den lille side for lille og den store side for stor. Når vi nu skal lave et nyt rektangel, vil vi derfor sætte den ene sidelængde til gennemsnittet af den lille og store side. Vi får da, at den ene side i det nye rektangel bliver

$$\frac{1+6}{2} = 3.5$$

Den anden side må, idet arealet skal være 6, være

$$\frac{6}{3.5} = 1.71429$$

Det nye rektangel ser sådan ud:



På denne måde fortsættes nu, idet den første side i det næste rektangel sættes til gennemsnittet af de to gamle sidelængder, og den anden side fås ved at dividere den første sidelængde op i 6. Tegn de to næste rektangler.

—KVROD(xiv)

Lav nu et program, der beregner kvadratroden af et tal. Den første linie i programmet kan se sådan ud:

KVADRATROD(X;Y)

I programmet kan du lade regneprocessen stoppe, når forskellen på sidelængderne i rektanglet er blevet mindre end 0.001.

Hvis du vil se sidelængderne af alle de forskellige rektangler, så kan du indbygge en udskrift af sidelængderne i dit program.

Afprøv dit program med inddataværdier mellem 1 og 100 og sammenlign resultaterne med en tabel eller med resultaterne på din lommeregner.

Opgave 2 Lav et program, som løser andengradsligningen

$$Ax^2 + Bx + C = 0$$

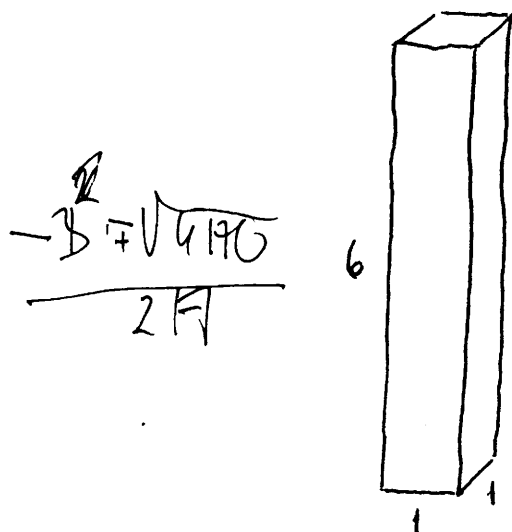
Programmet skal udskrive løsninger, når disse findes.

Opgave 3 Til beregning af kubikrødder kan benyttes en metode, der helt svarer til den, vi benyttede i opgave 1 til beregning af kvadratrødder.

Lad os beregne  $\sqrt[3]{6}$ .

Tallet  $\sqrt[3]{6}$  er sidelængden i terningen med rumfanget 6. Vi vil nu udelukkende betragte klodser (kasser), hvor grundfladen er et kvadrat, og hvor rumfanget er 6.

Vi starter med sidelængderne 1, 1 og 6.



$$\frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

$$B^2 - 4AC > 0$$

$$B^2 > 4AC$$

$$D = AC$$

$$47 = D \cdot 4$$

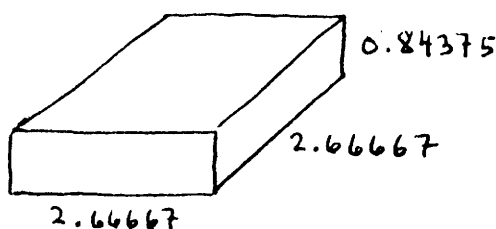
Den næste klods fås nu sådan:

- (1) Sidelængden i grundkvadratet fås ved at tage gennemsnittet af de tre sidelængder:

$$\frac{1+1+6}{3} = 2.66667$$

- (2) Højden af klodsen fås ved at sørge for, at rumfanget bliver 6:

$$\frac{6}{2.66667 \cdot 2.66667} = 0.84375$$



Beregn og tegn de næste to klodser.

Lav nu et program, der beregner kubikroden af et tal. Den første programlinie kan se sådan ud:

KUBIKROD(X;Y)

Afprøv dit program og sammenlign med tabel eller lommeregner.

Opgave 4 Peter er 148 cm høj og regner med at vokse 3% i løbet af det næste år.

Hans højde efter dette år kan da fås ved at sige:

$$148 \cdot 1.03 = 152.44$$

Tallet 1.03 kaldes for vækst-faktoren.

Hvis han fortsat vokser 3% om året, hvor høj er han da året efter ?  
(brug din lommeregner)

Lav et program, der beregner og udskriver en tabel over Peters højde i de næste ti år.

Mål din egen højde og benyt programmet til at udskrive en 10-årig højdetabel, idet du prøver med forskellige vækstfaktorer.

Hvor mange mennesker bor der i Indien om 100 år, hvis vækstfaktoren er uændret som i dag?

Hvad bliver til mest?

- (a) 5000 kr., som i 10 år halvårligt får tilskrevet 8% i rente.
- (b) 7000 kr., som i 10 år halvårligt får tilskrevet 6% i rente.

Hvis du indsætter 3000 kr. i en bank, som tilskriver rente med 9% halvårligt, hvornår er de 3000 kr. da vokset til 20000 kr?

Opgave 5 Lav et program, der udskriver en moms-tabel, som vist nedenfor.

| pris    | pris incl.moms |
|---------|----------------|
| 100.00  | 122.00         |
| 200.00  | :              |
| 300.00  | :              |
| :       | :              |
| :       | :              |
| 2000.00 |                |

Opgave 6 Hvis man bliver ved med at lægge sammen, hvor store summer kan man da få ?

$$(1) \quad 1 + \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots$$

$$(2) \quad 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \dots$$

Opgave 7 Lav et program, der opløser et naturligt tal N, som er større end 1, i primfaktorer.

Opgave 8 Den 1.januar 1984 startede Lone en opsparing, der består i, at hun halvårligt indsætter et fast beløb på 200 kroner. Hun får tilskrevet 6% halvårligt i rente, nemlig 1.januar og 1.juli.

Hvor meget har Lone sparet op den 1.januar 1992, hvor hun indsætter de sidste 200 kr.?

Det faste opsparingsbeløb kalder vi Y og den halvårlige rentefod R. I eksemplet her er Y=200 og R=0.06.

Det, der sker den 1.januar og 1.juli, kan da udtrykkes ved, at der først tilskrives renter til den aktuelle kapital K, og derpå indsættes beløbet Y. Denne beregning kan udtrykkes i de to tildelingsordrer:

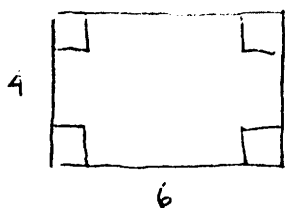
$K := K * F$

$K := K + Y$

hvor  $F = 1 + R$ .

Lav et program, der beskriver regneprocessen, og benyt programmet til at finde Lones kapital den 1. januar 1992.

Opgave 9 En metalplade er 4 meter bred og 6 meter lang. Ved at skære kvadrater bort i hjørnerne, folde siderne op og svejse dem sammen kan man lave et kar.



Rumfanget af karret er afhængig af, hvor store kvadrater vi har skåret bort i hjørnerne af pladen.

Hvor stort kan rumfanget af karret blive?

Lav et program, du kan benytte til at gætte dig frem til det største rumfang. Du kan kalde sidelængden i det lille kvadrat for  $x$  og udtrykke rumfanget ved hjælp af  $x$ .

$$K := K * R$$

$$Y = 2 - x$$

$$L = 6 - Y$$

$$B = 4 - Y$$

$$R = R + x$$

$$UD(x)$$

$$UD(R)$$

Slut -

# Oversigt

| ordre      | kort version | funktion                                                     |
|------------|--------------|--------------------------------------------------------------|
| HUSK       | KT           | indtastning af program                                       |
| LIST       |              | udskrift af program                                          |
| KATALOG    |              | udskrift af programnavne                                     |
| RET        |              | retning i program                                            |
| GLEM       | LG           | sletning af program                                          |
| LAGER      |              | skifter til lagerbillede og nulstiller variable på niveau 0. |
| NORMAL     | NM           | skifter til tekstsider og nulstiller variable på niveau 0.   |
| NUL        |              | nulstiller variable på niveau 0.                             |
| DEC(<tal>) |              | udskrift med <tal> decimaler                                 |
| GEM        |              | lagring af program på disketten                              |
| HENT       |              | hentning af program på disketten                             |
| SKRIV      |              | udskrift af program på skriveren                             |
| STOP       |              | stopper fortolkeren                                          |