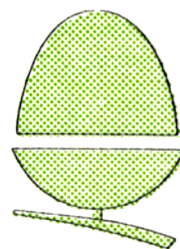


# Egebladet



Dansk blad for BBC/Acorn computere

---

NR.8

JULI/AUGUST

1989

---



# Indhold

## Artikler

- BBC-redaktøren** ..... 4  
Thomas Olsson introduceres ...
- Referat af generalforsamlingen** .... 7  
Morten kamp giver et referat at GF'89 ...
- DisARMing** ..... 13  
Følg med fra starten i  
H.O.Lundes ARM-tæmning ...
- C-kursus** ..... 17  
J.H.Ovesen og K.Latt åbner  
for C's godtepose ...
- GIF-GAF** ..... 21  
K.Latt om grafikformater ...
- DisARMing** ..... 13  
Følg med fra starten i  
H.O.Lundes ARM-tæmning ...
- Maskinkodekursus** ..... 24  
Torben Mogensen runder  
sit 6502 kursus af med 7. del ...

## Faste emner

- Redaktionelt** ..... 3
- Nyheder** ..... 10
- Læserbreve** ..... 11
- Hint og tips** ..... 20
- Annoncer m.m** ..... 30

## Listninger

- Fraktaler** ..... 23  
Henrik Pedersen om et "do it your-  
self"-fraktalprogram ...
- Mazegenerator** ..... 28  
Thomas Olsson implementerer  
en maskinkode Maze-generator ...

## Forskelligt

- BBS-siden** ..... 32

# Egebladet

Egebladet nr.8 Juli/August 1989

"Egebladet" udgives af foreningen "QUERCUS"

Redaktørene	(ARC):	K. Latt
	(ARC):	J. H. Ovesen
	(BBC):	T. Olsson
Grafisk tilrettelægning:		K. Latt
I redaktionen iøvrigt:		M. J. Søndergaard
		B. M. Andersen
Forside:		K. Latt

"Egebladet" udkommer såvidt muligt hver anden måned.  
Redaktionen sluttet den 20. Juli 1989  
Deadline for næste nummer 25. August 1989

Henvendelser vedrørende klubben :  
"QUERCUS" v/ M.J.Søndergaard  
Årestrupvej 76  
7470 Karup J  
Tlf. 97 10 21 95

Henvendelser, indlæg og øvrige ting omkring "Egebladet":  
"Egebladet" v/ K.Latt  
Nørregade 68 B # 1.h.  
5000 Odense C  
Tlf. 65 91 13 86

# Quercus

Formand: Olav M. Christensen  
Sundevej 30, Hvalpsund  
9640 Farsø  
Tlf. 98 63 81 50

Næstformand: Thomas Christensen  
Rosenvængets Allé 1 3. th.  
2100 København Ø  
Tlf. 31 42 28 96

Kasserer: Martin J. Søndergaard  
Årestrupvej 76  
7470 Karup J  
Tlf. 97 10 21 95

Quercus BBS: Tlf. 31 48 81 48  
SysOp: Næstformanden

Indbetalinger til klubben/henvendelser angående medlemskab:

Giro: 4 36 71 54  
QUERCUS  
Dansk Acorn og BBC brugergruppe  
Årestrupvej 76  
7470 Karup J  
Tlf. 97 10 21 95

# Redaktionelt

## Formanden beretter, den gamle redacteur takker af ...

Kære læsere af Egebladet.

Jeg vil gerne først og fremmest undskylde, at bladet denne gang har været lidt længe undervejs. Dette skyldes blandt andet, at bladet har skiftet redaktion, samtidig med at der har været en del omrokninger i bestyrelsen. Se venligst **"Referat af generalforsamlingen"** for en nærmere forklaring.

Bladet skulle gerne have et noget ændret udseende denne gang, dels - som tidligere nævnt - fordi bladet har skiftet redaktion og dels fordi vi nu vil prøve at lave bladet på et "rigtigt" **DTP-program**, nemlig **Acorns Desktop Publisher** (venligst udlånt af **Rex Data** - det er også mig).

Grunden til, at jeg har trukket mig tilbage som redaktør for bladet er ikke, at jeg ikke gider det mere; og jeg vil stadig levere indlæg af forskellig art til bladet. Den egentlige årsag er, at jeg skal starte på en uddannelse, som varer de næste 1 1/2 år. Indtil videre fortsætter jeg dog som formand (igen jævnfør referatet), så jeg har ikke helt opgivet ævred. Jeg har bare ikke tid til også at bestride jobbet som redaktør.

Jeg håber, I vil tage godt imod vores nye redaktører - de gør deres bedste. Med venlig hilsen og tak for den tid, der er gået.

**Olav Christlansen**, formand for Quercus.

Den nye redaktion takker, på hele klubbens vegne, **Olav** for hans store arbejde med bladet og ønsker ham held og lykke med sin uddannelse.

## De nye redaktørers tanke-dryp

...

Da dette er vores første blad, er vi specielt bekymrede over resultatet af vore anstrengelser ... Er det godt nok? Layout'et, er det for gnidret? Artiklerne, falder du i søvn? Stoffets sammensætning, er den for teknisk eller måske for populistisk? I alle tilfælde er det **NU** du skal gøre redaktionen opmærksom på svaghederne.

I bladets midte er der denne gang et **"Reader-survey"**, som gerne skulle besvares af så mange så muligt, d.v.s også **DIG!!!**

**Ekstra gulerod** .. Der trækkes lod om en inflationsramt **500-kroneseddel** blandt de indsendte "survey's".

Redaktionen vil gerne gøre det klart at : **"Vi tåler al kritik"**, så hold dig nu endelig ikke tilbage. Klubbens fortsatte eksistens afhænger simpelthen af dit respons.

Efter disse smagløse og ubegrundede udfald, vil vi gerne lige fortælle hvem vi er.

**Archimedes** redaktionen består af **Jens Henrik Ovesen** og **Klm Latt**. Begge læser vi **Datateknik (Stud. Scient. Tech)** ved **Odense Universitet**.

**J.H.O.** er 23 år og har i kraft af sit årelange virke indenfor **BEEB-spil-crackning** et indgående kendskab til **BBC'en** og har også check på **Archimedes'en**.

**KLatt** er 26 år og var landets første ejer af en **Archimedes**, derfor har han en stor erfaring i brugen, programmeringen m.v. af **Archle**.

I **BBC**-redaktionen sidder : **Thomas Olsson**, han er 22 år, er filosofisk anlagt og lader sig ikke introducere på en kvart spalte ... (Se næste artikel).

# BBC-redaktøren

## Thomas Olsson om livet, danskerne, ensomhed, moral og meget mere...

*Som nævnt i referatet fra generalforsamlingen, har vi på Egebladet fået en ny redaktør for B'er og Master-stof. Valgt til dette blev 6580, Thomas Olsson, som vi har bedt om at skrive lidt om sin baggrund for netop dette hverv.*

På generalforsamlingen blev det luftet, om det mon ikke var en god ide, at splitte hvervet som redaktør, så vi fik en Archle- og en B/Master-redaktør. Forslaget faldt i god jord, og vi valgte, som opstillet, Jens H. Ovesen og Kim Latt som Archle-redaktører.

Derefter skulle nogen jo opstille til det andet hverv. Der var en stilhed som i graven, indtil jeg hørte mig selv sige: "Okay, jeg stiller op.". Hvorfor nu det? Hvilke forudsætninger har jeg dog som BBC-redaktør? Og hvad vil jeg i det hele taget med et sådant ansvar?

Jo, det kan faktisk udtrykkes meget kort, men da det ikke ligefrem vælter ind med stof til bladet (se løvrigt nedenfor), vil jeg ikke gøre det. Næh, lad os tage det hele fra begyndelsen ...

I mine unge dage (er nu 22 år) har jeg drevet et par computerforeninger, begge med tilhørende blad, faktisk meget lig Quercus, hvis man ser bort fra det landsdækkende aspekt. Foreningerne var i bund og grund forskellige, da den ene udelukkende henvendte sig til ORIC-ejere (Er der nogen der kan huske den?) og den anden til alle computer-ejere. Begge klubblade kæmpede dog med samme økonomiske og samme stof-problemer, som herværende

Egeblad. Jeg var, som formand for klubberne, også redaktør for bladene, og har derfor beskeden erfaring med det, jeg kalder snævre fagblade.

Nå, men det viste sig snart, at ORIC var en døende maskine, og ideen med en klub for alle, var der stort set kun interesse for fra de dengang uhyggeligt mange 64-freaks, der gjorde sig som crackere af diverse spil.

Nogenlunde samtidig købte jeg min første Master, og interessen for 'resten af pøbelen' fortog sig kraftigt.

Og her sad jeg så i 2 stive år med en følelse af, at jeg var den eneste i Danmark, der havde en ordentlig computer. Alle lo ad den, indtil de så den køre, naturligvis. Men det springende punkt var, at der ikke var nogle fede spil til den. Det skal dertil siges, at jeg på det tidspunkt kendte hele 2 andre, der begge havde B'ere, men det kunne ikke rigtig svare sig med programudveksling, da 75% af B'er programmerne ikke kunne køre, og resten stort set var hjermedøde spil.

Men, næsten intet varer evigt, og en dag mødte jeg Thomas SysOp gennem Den blå Avls. Vi gjorde nogen forretninger, og udvekslede erfaringer, men der skulle gå et helt år, inden han, som en sidebemærkning, nævnte Quercus. Meget forbløffet over, at jeg aldrig have hørt om den landsomspændende klub, fik jeg Olav's nummer, og blev omsider medlem. Lige tids nok til den netop afholdte generalforsamling, som jeg så deltog i.

Resten burde være bekendt fra referatet, og nu er jeg så BBC-redaktør. Hvad vil jeg så med det?

Jo, nok har jeg siddet med Master i 3 år og har fået en hel del erfaringer med denne maskine, men det betyder jo ikke, at jeg i de næste mange år kan lave hundredevis

af artikler om BBC. Nej, ikke uden stof udefra.

'Jamen', siger den ikke alt for kvikke læser, 'Er det ikke nemt at få? Skriver man ikke bare i bladet, at man gerne vil have stof tilsendt?'

Nej, det er desværre ikke nok.

"Hvorfor ikke", spørger den samme læser.

Lad mig komme med et eksempel.

Til daglig er jeg tekniker på en radio, hvor vi tit og ofte laver en lytter-quiz, eller på anden måde vil have lytterne til at ringe ind.

Det er meget nemt; den nystartede disc-jockey siger bare lidt om, at der er quiz, hvad præmierne er, telefonnummeret osv. Og så sker der ellers **ingenting**.

Overhovedet ingenting.

Så tænker han: "*Hvorfor i alverden er der ikke nogen, der ringer ind? Jeg kunne da godt tænke mig at vinde en kam eller en rejse eller lignende*"

Og der er stadig ingen, der ringer.

DJ'en sidder og sveder, og hver gang musikken er slut, skal han finde på en ny undskyldning for ikke at have en lytter igennem endnu.

Til sidst bliver det for meget for ham, og han ringer til en af de mere erfarne DJ's på stationen. "*Det ordner jeg! Lige et øjeblik.*" Hvorpå næsten straks rører på.

Den grønne DJ sidder forvirret i nogle sekunder, hvorpå telefonen ringer.

Det er selvfølgelig den erfarne, der siger, at han gerne vil igennem, og gætte.

"*Det kan man da ikke, det er jo snyd!*", men der er ingen kære mor. Så endelig har han en "lytter" igennem, der selvfølgelig svarer forkert. Kort efter ringer telefonen igen, og minsandten! Denne gang er det en vaskægte lytter, der gerne vil være med i quizz'en. Og sådan lavede han radio lykkeligt, til programmet var slut.

**Hvorfor var alt det bøvl nødvendigt?**

Se, danskerne er et specielt folkefærd. Der er ikke ret mange, der gør noget, som andre ikke har gjort før.

Hører man at de vil have lyttere igennem i radioen, venter man til der har været en lytter igennem. Ret ulogisk, men også ret sikkert. Tænk nu, hvis man blev gjort til grin eller noget andet slemt. Desværre gælder der ikke kun radio.

Forestil dig, at det var ganske normalt i vores klub, at havde man lavet en ny rutine, så var det første man gjorde, at skrive et par ord om den, derefter at sende hele molevitten til redaktøren. "*Jamen, jeg er ikke så god til at udtrykke mig skriftligt.*" Nå! Jeg skulle ellers mene, at hvis der er **NOGEN**, der er kritiske med hensyn til skriftlig nøjagtighed, er det da en computer.

Ergo: Har du skrevet et program, kan du også skrive bare 2 linier om det. Send det så til mig, så skal jeg nok fylde noget mere på.

Når du har læst hertil, er du måske blandt de 80%, der tænker: "*Ja, det er da meget smart. Det skal nok få nogen til at indsende stof.*" I så fald tænker du forkert! Nu skal du høre, hvad du skal tænke: "*Hvis alle gør det, bliver det et virkeligt brugbart blad.*" Og derefter: "*Jamen, jeg vil da gerne have et brugbart blad, og hvis alle skal være med, må jeg hellere finde den smarte rutine frem, jeg lavede i går.*" Gå ind på denne tanke, og vi får et blad, der er langt lædigere end de fleste engelske, alene på grund af stoffets aktualitet og brugbarhed.

Dette var en stor bue udenom, og dog alligevel igennem, problemets kerne. Kerne var vort eget ønske om et godt blad, multipliceret med vor følelse af, at vore erfaringer ikke er noget værd. Det turde nu være fastslået, at vore erfaringer er det eneste, der er noget værd. Det er jo ikke

nogen hemmelighed, at der ikke sker så meget på **BBC**-markedet som for 7 år siden.

Jeg skal gerne indrømme, at der sker betydeligt mere end på **ORIC**-markedet. Det vil altså også sige, at vi **ENDNU** har fat i den lange ende.

Når først **BBC**-ejerne er spredt for alle vinde, er der intet, der kan restaurere situationen. Hvis det står til mig, sker dette ikke foreløbig. Midlet? Ja, vi har en klub, som stort set kun eksisterer for udgivelsen af et blad, samt en komsammen en gang om året.

Al sikker kommunikation foregår gennem bladet, og jeg ser det meget vigtigt, at du beholder interessen for bladet. **MEN**, det er ikke mig, der ved hjælp af utallige artikler om det, jeg nu synes er interessant, skal holde din interesse ved lige. Det er **DIG**, der skal vække "de andre"-s opmærksomhed. Og tro mig: Det program, du indsender i morgen, er ikke kun interessant for funktionens skyld, men især også for ideens skyld. Lad andre se dine resultater, og lad dem bygge videre på dem, og ikke mindst omvendt.

Tag som eksempel min labyrint-rutine i dette nummer. "*Den kan lave en labyrint, og hvad så?*" siger mange (men naturligvis ikke dig.). Næh, du tænker "*Den laver en labyrint ja, men hvordan?*" Så finder du ud af, at der er gjort brug af en software-stak i maskinkode, og straks myldrer ideerne frem. Nå ikke? Så kig på et af de andre programmer. Jeg garanterer for, at med lidt kreativitet i baghovedet, kan de fleste programmer (mit inklusive) enten laves mere effektive, eller ændres til noget **DU** har brug for. Hvis du laver rekursive programmer, har du sikkert allerede en masse ideer.

Jeg håber, at du er enig, så vi kan få et godt blad.

Men som det sig hør og bør, skal vi nu til

moraleme.

**Morale 1:** Brug bladet til noget!

**Morale 2:** Distribuer dine resultater via bladet, så vi andre kan følge morale 1.

Hvis du har læst denne smøre, ved du, at der venter dig en ny i næste nummer, hvis ikke jeg har fået noget af dit stof inden da.

Med håb om godt samarbejde,

**Thomas Olsson**

**Efterskrift.**

Hvis du har følt dig påvirket af ideerne i ovenstående tekst, og har forstået, at det er din pligt overfor dine egne interesser at indsende noget stof, kan du gøre det på følgende måder:

Pr. Brev:

**Thomas Olsson**

**Kagså Kollegiet 155/5**

**2730 Herlev**

Eller upload det i mit personlige directory på basen.

## TOTALLØSNINGER

Nøglefærdige systemer:

- **TWINHEAD PC**'ere
- Kontorautomatisering
- Programudvikling
- Maskinstyringer
- Programoptimering
- Systemplanlægning
- Tilbehør

Ring og få løsningen på dit problem

**CFM** Lundegaardsvej 7  
5672 Broby  
*computing* Tlf. 64 75 10 03

# GF-'89

## Generalforsamling i Odense ...

En del af generalforsamlingens deltagere mødte allerede op om lørdagen. Efter at have beundret **Klatt's Archimedes**, spist, snakket og indtaget nogle langnæsedede (meget hyggeligt) begyndte generalforsamlingen meget tidligt søndag morgen ...



Formanden vasker op

## Referat af GF'89 ...

Generalforsamlingen afholdtes hos **Kim Latt, Nørregade 68B Odense**. 18 var mødt eller ca. 21 % af medlemmene.

### Pkt. 1: Valg af dirigent og referent.

Som dirigent og referent valgtes henholdsvis **Kim Latt** og **Morten Kamp**.

Dirigenten konstaterede, at generalforsamlingen var lovligt indkaldt og gav derefter ordet til formanden.

### Pkt. 2: Formandens beretning.

**Olav Christiansen** konstaterede, at der i det forløbne år havde været god medlems-tilgang. Han rundgav en kurve, der viste udviklingen af tilgangen. Der er nu 86 medlemmer. **Thomas Christiansen** supplerede med at oplyse, at der nu er 56 personer tilsluttet bulletinboardet. Heraf er ca. 20 aktive. Der er enkelte ikke-Acom brugere på boardet. De får af en eller anden grund tit softwaren til at gå ned. **Thomas Christiansen (SysOp)** vil lejlighedsvis slette brugere, som overhovedet ikke er aktive. Formanden oplyste løvrigt, at året var forløbet uden større hændelser. Det havde været præget af udgivelsen af **Egebladet**.

### Pkt. 3: Kassererens beretning.

**Martin Søndergaard** oplyste, at regnskabet af tekniske grunde kun gik til 15. februar 1989.

Økonomien hænger nogenlunde sammen, men er præget af, at det er forholdsvis dyrt at producere bladet. Kassereren skønner som tidligere, at økonomien vil blive "sund" ved en medlemstilslutning på ca. 100, hvilket, jfr. ovenfor, ikke er urealistisk.

Det vedtoges efter diskussion forsøgs-mæssigt at udgive næste nummer af **Egebladet** i fotokopieret form (dette KAN ændres i sidste øjeblik, Red.).

### Pkt. 4: Godkendelse af regnskab.

Regnskabet godkendtes enstemmigt.

### Pkt. 5A: Kontingentforhøjelse.

Kassereren foreslog, at kontingentet med virkning for fremtiden forhøjes, evt. til 175 kr/år.

Efter nogen drøftelse heraf, herunder muligheden for at begrænse forhøjelsen til 150 kr. vedtoges kassererens forslag, idet der navnlig henvistes til, at dette ville muliggøre en væsentlig udvidelse af aktiviteterne ud over bladdudgivelsen.

Kassereren oplyste, at der snart forfalder en større portion kontingenter, således at virkningen af forhøjelsen kommer snart.

#### **Pkt. 5B: Finansiering af BBS'et.**

**Thomas Christensen (SysOp)** oplyste, at udgifterne til etablering af BBS'et udgør ca. 3.500 kr + telefonabonnement. Det vedtoges, at klubben betaler abonne-



Alle "Nødeme" på én gang

mentet; og **Rex-Data** vil antagelig efter forhandling dække størstedelen af de øvrige etableringsomkostninger. Det vedtoges, at der på BBS'et - efter nærmere forhandlinger - kan etableres en reklamedel, hvor indtægterne kommer foreningen til gode.

#### **Pkt. 5C 1: Ændring af vedtægternes § 6.**

Det vedtoges, at der til § 6 tilføjes:

*"samt **SYSOP** på klubbens bulletin board", således at denne også til enhver tid vil være "født"*

*medlem af bestyrelsen."*

#### **Pkt. 5C 2: Ændring af vedtægternes § 7.**

Det blev efter diskussion vedtaget, at valgperioden for bestyrelsen fortsat er et år for alle medlemmer.

#### **Pkt. 5C 3: Ændring af vedtægternes § 10, Regnskabsår som kalenderår.**

Punktet vedtoges enstemmigt.

#### **Pkt. 5C 4: Ændring af vedtægternes § 12 om eksklusion af medlemmer.**

Formanden redegjorde for en episode (omtalt i sidste nr. af **Egebladet**), hvor man havde måttet "blackliste" en person med navns nævnelse. Han skønnede, at der var tale om et ret enestående tilfælde.

Efter en ret omfattende diskussion vedtoges at ændre § 12, således at denne fremtidigt lyder:

#### **"§ 12**

**Stk. 1.** Eksklusion af et medlem kan ske efter bestyrelsens beslutning, dersom medlemmet efter fremsendelse af behørig rykker ikke betaler kontingent.



Kassereren aflægger beretning

**Stk. 2.** Eksklusion kan endvidere ske, dersom en enstemmig bestyrelse i særlige tilfælde vedtager dette. Efter tilsvarende regler kan andre personer omtales i Egebladet. En beslutning efter nærværende stykke kræver altid forudgående udarbejdelse af en redegørelse, der skal forelægges førstkommande generalforsamling."

#### § 6: Forslag fra medlemmerne.

Der var ikke indkommet nogle forslag.

#### § 7: Valg af ny bestyrelse.

Den nuværende formand, **Olav Christlansen**, tilbød at trække sig tilbage, blandt andet for at give plads til "nyt blod" i klubben, men lod sig dog opstille - som den eneste - til formandsposten.

Bortset fra kassereren, **Martin Søndergaard**, var der ingen af de øvrige i bestyrelsen, der genopstillede.

Til bestyrelsen valgtes:

1. **Formand :**  
**Olav Christlansen** (genvalg)
2. **Næstformand:**  
**Thomas Christlansen** (nyvalg)
3. **Kasserer:**  
**Martin Søndergaard** (genvalg)
4. **Sekretær:**  
**Morten Kamp** (nyvalg)
5. **Menligt medlem:**  
**Kim Latt** (nyvalg)
6. **Første suppleant:**  
**Jens Ovesen** (nyvalg)
7. **Anden suppleant:**  
**Thomas Olsson** (nyvalg)

#### § 8: Valg af revisor.

Som revisor valgtes **Torben Osted** (genvalg).

#### § 9: Eventuelt.

**Olav Christlansen**, som indtil nu både har været redaktør af "Egebladet" og formand for klubben, ønskede at afstå fra redaktørjobbet indtil videre. Der bliver derfor en del redaktionelle ændringer, hvad angår bladet:

**Archimedesredaktør bliver Kim Latt**

**B/Masterredaktør bliver Thomas Olsson**

**Chefredaktør bliver Bo M. Andersen.**

**Kim Latt** er villig til at stå for udskrivning af



teksterne, indsamling af stof m.v. løvrigt fordeles redaktørerne selv opgaverne mellem sig. Redaktørerne opfordrede til, at der bliver indsendt mere stof fra læserne, idet det er dette, som holder liv i bladet. Der var enighed om, at

det var ønskeligt, at så mange som muligt tilsluttes BBS'et, idet der sker en del på boardet, både i meddelelsesafsnittet og i fil-afsnittet.

Faciliteterne på boardet udvides løbende, og de tidligere fejl er nu så godt som afhjulpet.

#### § 10: Løst og fast.

Afsluttende diskussion. Der var ingen bemærkninger hertil.

Ordstyreren takkede for god ro og orden og afsluttede mødet.

# Nyheder

## Stort og småt om Acorn's fra ud- og indland ...

Tæsk til **80486**, **80586** osv! Med denne nyhed fra **Aleph Ltd.** er **Archle** stadig blandt "The big awesome boyz". **Aleph One** hedder nyheden, og lover med **ARM3** og 24MHz system-clock, 3-5 gange større hastighed. **ARM3** er en **ARM2** med 8Kbyte data og program on-chip cache-lager. Hardware implementeringen af cache-lageret er specielt optimeret til at presse de sidste dråber ud af **RAM'en** i **Archle**. **Archle's** grafik lægger beslag på en stor del af memory-båndbredden. Throughput (hastigheden) kan være reduceret med 60-70% afhængigt af hvilket mode man benytter, ligesom på **Electron** (ære være dens blakkede minde).

Prisen lader til at variere mellem "a few hundred pounds" og 500 £. Men selv til 500 £ + 900 £ for en **Archimedes**, er **Aleph One** og en **Archimedes** et ekstremt godt tilbud på en **Folkevogns-Cray**. Redaktørene af **Archimedes** sektionen venter spændt på at **Aleph Ltd.** sender et anmeldelses-eksemplar!!!!

Et nyt engelsk firma, **Mach Technology**, træder ind hvor **Acorn** har svigtet .

Med RAM-opgraderinger (2 Mb, 4 Mb), **Modula-2**, (endnu en) **BASIC**-compiler, billige harddiske, samt en **80286 2nd-processor** til **Archimedes'en**, træder dette nye firma ind i **Archimedes**-arenaen med et stærkt og alsidigt sortiment.

Yderligere oplysninger fås hos :

**Mach Technology**  
75 Hydewood Road  
Little Yeldham  
Essex CO9 4QX  
Tel. 00944 787 238125.

**RAM:** 340-520£ inc. VAT  
**Modula-2:** £89.95 inc VAT.  
**Mach BASIC:** £39.95 inc VAT.  
**20 Mb harddisk:** £249 excl VAT.

I øjeblikket er **Acorn's** danske forhandler **Rex Data** ved at undersøge muligheden for at hjemtage nogle af disse spændende ting.

## Play it again Sam n<sup>o</sup> 10 ...

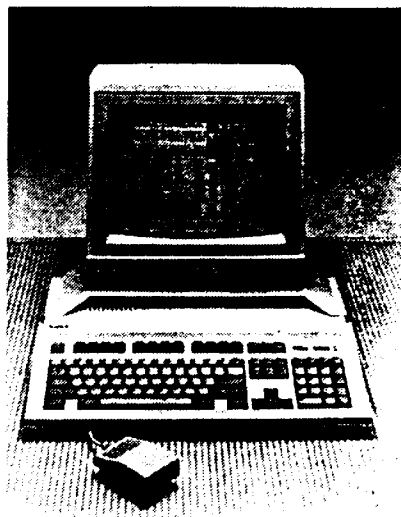
Traditionen tro har **Superior Software** barslet med endnu en kompilation. Denne gang er det nye release **QWAKI**. De gamle hits er denne gang **Zalaga**, **3D-Dotty** og **Repton Thru Time**.

Tape .....	9.95£
5.25" Disc .....	11.95£
3.5" Disc .....	14.95£

Kontakt **Superior Software** eller **Rex Data** for yderligere information.

## A3000 er i landet ...

En ny og billigere **Archimedes** er lige kommet til landet. **A3000** (potens nummerering !!) er en "alt i en"-box computer, ikke ulig **AMIGA 500**. Der er tale om en ulv i fåreklæder. Den er nemlig ca. 10% hurtigere end eksisterende **Archimedes**-modeller !!!



# Læserbreve

## Henrik Pedersen (6577) skriver:

I de sidste snart mange numre af **Egebladet** er der blevet efterlyst respons fra medlemmerne og jeg syntes nok det har været lidt tyndt med reaktioner. Så jeg lagde hovedet i blød og det er faktisk ikke så svært at finde emner for fremtidige indlæg fra læserne: Her er nogle af dem jeg fandt på. Der må bestemt være andre medlemmer der har ideer, så kom frem med dem. Det kunne jo være der var nogen der havde forstand på netop det du gerne vil vide mere om.

Jeg har netop fået en **Archimedes**, så jeg er ret positiv overfor forslaget fra sidste nummer om en artikelserie om **ARM**-assembler. Derudover kunne jeg godt tænke mig at se noget om programmering af computerens hardware, video- controlere, lyd kredse og **I/O**-enheder.

Eller hvad med en artikel om **3-D** computergrafik eller om de mange forskellige opgrader der findes, specielt til **BBC B**? Det kan heller ikke passe at der ikke er nogen der har lyst til at anmelde deres yndlingspil eller anmelde nyheder, hardware eller andet software. Se nu at komme igang. Egen erfaring viser at starten er den langt sværeste; er man først i gang går det faktisk rimelig nemt.

Med venlig hilsen  
**Henrik Pedersen.**

## Jan Vibe (6513 ) skriver:

Hej TV-folk og filosoffer (**BBC** og **Archimedes** ejere).

Det er utroligt som tiden går. Nu er **Quercus** allerede et år gammel, tillykke. Når et år er gået, er det jo skik at gøre

status.

Hvordan har klubben fungeret? Er der noget der blev gjort godt? Er der noget der kunne gøres bedre?

Jeg synes fa'me at det er **FLOT** klaret af klubbens bestyrelse m.v. at få medlemstallet banket op fra en 10-15 **EDB**-freaks til næsten hundrede medlemmer. Det viser at der faktisk stadig findes mennesker her i landet der interesserer sig for **Acorn** computerne.

Et andet punkt jeg gerne vil rose, er klubbens service til medlemmerne. Jeg har selv været yderst tilfreds med den service der er blevet ydet mig fra klubbens side, i forbindelse med køb af nyt anlæg. Faktisk har jeg kun én gang tidligere oplevet en lige så god service, og det var pudsig nok, da jeg købte min første hjemmecomputer, en **PET 2000**, for præcis 10 år siden.

Klubbladet 'Egebladet' er også et interessant og intelligent indslag i det efterhånden komplet **PC**-dominerede **EDB**-bladmarked. Lad os håbe at vi efterhånden får så mange medlemmer, at vi kan sende bladet ud i kioskerne.

Det var rosen. Risen er så at jeg synes at der er for få af medlemmerne der skriver til bladet, der var næsten ingen indlæg fra læserne i nr 7. Man risikerer at det hele tiden bliver "**Tordenskjolds soldater**", der får deres ting ud på tryk. Det kan hurtigt blive en trist affære, så derfor er dette en opfordring til "det tavse flertal" om at komme ud af busken.

Der må da være nogen flere, der har meninger, programmer, kritik, ros, ideer, spørgsmål, kommentarer, rettelser, forslag og smarte/beskidte tricks, som andre kunne have glæde af, eller hvad?

Jeg har sørme også nogle forslag.

1. Ved gennempløjning af diverse compu-

terblade m.v. kan man nemt få det indtryk at der endnu ikke er kommet en RISC baseret computer på markedet endnu. Der er simpelt hen ikke nogen i Danmark der aner at Archimedes'en eksisterer. Så var det jeg tænkte... Hvad nu hvis Quercus prøver at tage kontakt til f.eks. Ole Grunbaum, der er Politikens EDB-skrubent, og demonstrere Archimedes'en for ham. Det ville sandsynligvis skabe en del omtale af både Quercus og Archimedes.

Det er jo på den måde at andre firmaer får deres produkter omtalt. "Hint": Vi kunne jo prøve at sende ham nogle eksemplarer af bladet her.

2. Hvad med at lave en spørgsmål og svar brevkasse?

Hermed slut på status'en.

I forbindelse med desktop'en vil jeg som den første (tror jeg) rapportere en fejl. Prøv at indtaste dette lille program, og gem det på en diskette:

```
10 FOR N%=0 TO 500:CIRCLE 640,512,N%:NEXT
```

Når man så i desktop'en aktiverer det med musen, får man så åbnet et vindue, hvor det vist er meningen at programmets output skulle vises. det skulle ikke være muligt at komme til at tegne udenfor vinduet, men det er det alligevel!

Tilsyneladende bliver der ikke, ved åbningen af vinduet, afgivet en VDU 24 ordre der skulle sikre at grafikken holder sig inden for det åbnede vindue.

*(Dette er ikke en bug. Vinduet der åbnes er et tekstvindue. Ved den rette brug af desktoppen, d.v.s. hvis programmet fortæller desktoppen at det vil tegne eller skrive på skærmnen, vil det grafiske vindue blive sat når der skal skrives i det, samtidig med at man får oplyst hvor det er i arbejdsområdet. En noget længere forklaring er nok påkrævet. Se evt. referencemanualerne. JHO).*

Til sidst vil jeg spørge de øvrige medlemmer ude i samfundet, om de også har haft svært med at finde det engelske blad 'Acorn User' eller om det er et specielt københavnsk fænomen.

Til allersidst en vits:

**Programmør1:**

*Nå, du skriver dine programmer efter politikermethoden hva'?*

**programmør2:**

?????

**programmør1:**

*Ingen kommentarer.*

Grin Så!

Venlig hilsen  
Jan Vibe (6513).

(Jan Vibe har medsendt en del, som han selv kalder det, skøre programmer. En del af disse er på klubdisketten.).

Vi kan også en vits:

To æg kommer trillende på en bassinkant. Det ene æg falder i. Plopl siger det. Så falder det andet æg i. Det siger også plopl

Grin så. (Klatt og JHO)

**Ze Zomble (65007)** skriver:

Er tzer tznart nogne nye progz to haunt?

Redaktionen (???) har ingen kommentar.

**Jens "Clint" Ovesen (6545)** spørger:

Ska' vi så snart ha' noe kaf'?

**Ze Codemaster (6533)** svarer:

Lav ze zæl, zidder lige og koder.

Disse tilsyneladende imbicile og infatille læserbreve nåede igennem redaktions strenge censur (Desuden fylder det godt op).

# DisARMing

## Hans O. Lunde om ARM'en ...

*'It is commonplace for programmers to tick exclusively to high-level languages and never to look inside their machines. Alsing felt that they were missing something. He remembered learning assembly language during his time of midnight programming. 'It was neat to learn it. I could skip the middle man and talk right to the machine. It was also great for me to learn that priestly language. I could talk to God, just like IBM.'*

(Fra bogen "The soul of a new machine" af Tracy Kidder, s.92. )

Der er intet overjordisk ved at programmere i maskinkode, snarere tværtimod. Når maskinkode samtidig bliver brugt mindre og mindre og umiddelbart synes sværere end **BASIC** og andre højniveausprog, hvorfor så bruge tid på det? Denne artikel og forhåbentlig andre af diverse forfattere håber at kunne give dette spørgsmål et interessant svar.

Torben Mogensen har netop fuldendt sit kursus i **6502**-assembler, og mange har forhåbentlig fulgt de glimrende artikler. Vi vil dog ikke forsøge at sammenkæde denne serie med den tidligere, da formålet med **ARM**-assembler må siges at være et lidt andet. For det første bliver artiklerne en generel introduktion til maskinsprog, brug af en assembler, hvordan man laver programmer i maskinsprog på en Archimedes etc. Det skal være muligt for en bruger, der ingen maskin-sprogserfaring har, at starte med disse artikler og selvfølgelig nogle eksperimenter med maskinen derhjemme. Det er altså ikke en spalte, hvor erfarne assemblerprogrammører udgyder deres forskningsprojekter for den undrende læser. Der bliver ikke tale om en syste-

matisk gennemgang af alle instruktioner, instruktionstider, busforbindelser osv.

Vi henviser til litteraturen på området for en komplet indføring. Bogen "**Arm Assembly Language Programming**" af **Peter Cockerell** kan anbefales, men der findes også andre kilder.

Artiklerne vil blive centreret om nogle eksempler sammen med en gennemgang af et eller andet område. Samtidig håber vi at kunne gøre det til et sted, hvor programideer, algoritmestumper, datastrukturer, små irriterende fejl, brug af andre sprog, misforståelser og forliste projekter kan blive behandlet. En slags work-shop er måske en rimelig betegnelse for denne idé, men der skal kun køre et hovedtema af gangen. Evt. kan vi lave en slags brevkasse, sidst i artiklerne, til kommentarer, korrektioner af diverse bommerter, ros m.m. Hvis artiklerne når dertil, at programmerne på nogen måde kan være brugbare, vil vi forsøge at opbygge et veldokumenteret bibliotek af program-eksempler. Det er samtidig meningen, at flere personer skal skrive artikler, hvorved vi håber at opnå en levende spalte. **Quercus BBS** ville være et fint sted at sende indlæg og programmer til, men ellers er brev- eller telefonmetoden også brugbar. Detaljerne må vi finde ud af hen af vejen.

Citatet i begyndelsen af artiklen antyder, at der er noget særligt ved at bruge maskinsprog. Hvis du kan lide at programmere, vil dette næsten med sikkerhed også være tilfældet for dig. Det skyldes nok primært, at der en tilfredsstillende i at forstå, hvordan computeren fungerer, også på et lidt dybere niveau. Dertil kommer glæden ved at ud-trykke ideer, hvor simple de end måtte være, og føle at man behersker maskinen på dette niveau.

Din **Archimedes** er et meget kompliceret stykke teknik. Jeg sidder lige nu og skriver ved hjælp af editoren, der følger med

**RiscOS.** Ind imellem tager jeg fat i musen og foretager en operation, teksten ruller over skærmen og nogle gange kommer der vinduer til syne oveni teksten. Jeg gemmer teksten i en ramdisk; en "kunstig" diskette, der er afsat af computerens frie hukommelse. Alt dette er meget kompliceret, når man påtænker, at princippet i en computers centralenhed hviler på, at man kan sammenligne tal ved hjælp af elektriske signaler. Centralenheden i din **Archimedes** er **ARM, Acorn Risc Machine**, en 32 bits **RISC** microprocessor. Når editoren eller et hvilket som helst andet program kan udføres, skyldes det at denne microprocessor kan udføre simple instruktioner. Disse instruktioner er lagret i hukommelsen, hvor processoren henter dem og udfører dem. **ARM**-processoren kan kun udføre den slags instruktioner, og de kaldes derfor maskininstruktioner. Dette er det laveste niveau, man kan bevæge sig på som programmør, og vi vil ikke beskæftige os ret meget med elektronikken, der ligger under dette niveau.

At instruktionerne til en vis grad er simple gør, at processoren kan udføre dem med svimlende fart; på de øjeblikkelige **Archimedes**-modeller udfører processoren 8 millioner instruktioner pr. sekund, når det går bedst. At processoren er på 32 bit vil sige, at den kan behandle tal på 32 bits længde. Dette er ganske meget for en microprocessor, det normale har været 8 og til nøds 16 bit i årevis.

**ARM**-processoren hører også til hurtigløberne indenfor microprocessorverdenen, men det er ikke kun på grund af de 32 bits. **RISC** er forkortelsen for **Reduced Instruction Set Computer**, hvilket betyder at processorens instruktionssæt er reduceret. Andre processorer har mange flere instruktioner, de kaldes tit for **C(omplex)ISC**-processorer. **ARM**'en har få, men meget kraftfulde instruktioner.

Udviklingen af microprocessorer gik i

mange år mod at lave større og mere komplicerede instruktionssæt. Det har dog vist sig ikke altid at være det bedste. Compilere vil som regel producere maskinkode, hvor 20% af instruktionssættet står for 80% af de udførte instruktioner. Med **ARM**'en har man taget konsekvensen af dette og lavet en **RISC**-processor; det princip ser ud til at være sidste skridt, inden computerarkitekturen bevæger sig over mod, at mange processorer arbejder sammen i et netværk.

Kurset bliver altså baseret på absolut moderne teknologi, ingen der ejer eller har set en **ARM**-baseret computer er sikkert i tvivl om det!

Maskinkode er tal eller elektriske spændinger, som processoren kan afkode og udføre som instruktioner. En **ARM**-instruktion er 32 bits lang, 4 bytes. Instruktionerne hentes fra ram; faktisk har **ARM**'en hele tiden tre instruktioner under behandling. Hver gang en instruktion udføres, bliver der samtidig afkodet en og hentet en fra hukommelsen, dette kaldes "pipelining". Det er ikke nemt at huske tal, derfor bruger man en assembler("samler") til at lave maskinkode-programmer med. Med en assembler kan man tildele de enkelte instruktioner symbolske navne, som assembleren så oversætter til tal. **BASIC** på **ARC**'en har indbygget en assembler, og den vil vi anvende i dette kursus. Der er i modsætning til **BASIC** 1:1 korrespondens mellem de instruktioner, man skriver i sit assemblerprogram, og de instruktioner programmet rent faktisk kommer til at bestå af. Linien :

```
IF x%>0 THEN PRINT "x% er større end nul"
```

Vil kræve adskillige maskinkodeinstruktioner. Derfor har man opfundet høj-niveausprog som **BASIC**.

I maskinkode er man i princippet nødt til at gøre alting selv, og det kan synes meget besværligt. Efterhånden finder man dog ud af, at det ikke er så galt endda. Eksempel

på en assemblerlistning efter at programmet er "assembleret":

```
0094C4 E3A00002 MOV R0,#2 ; læg tallet 2 i R0
0094C8 E3A01002 MOV R1,#2 ; ditto i R1
0094CC E0812000 ADD R2,R1,R0; adder resultat
```

Disse tre instruktioner udfører regnestykket  $2+2=4$ . Den venstre kolonne viser hvor i hukommelsen den første instruktion ligger, den midterste; de fire bytes instruktionen er sat sammen af, og den højre vores assemblersymboler for instruktionerne, de kaldes også mnemonics.

Det er ikke noget overvældende program, og for at komme lidt videre må vi ind på **ARM**'ens opbygning. Den har en slags intern, meget hurtig hukommelse, dette kaldes registre; processoren har 27 registre, der alle kan indeholde 32 bits. Det er dog kun de 16, der umiddelbart har interesse for os. Vi vil indtil videre referere til registrene med assemblersymbolerne **R0-R15**. I registrene kan man opbevare og bearbejde tal, der er hentet fra ram eller andre registre eller er fremkommet efter en af **ARM**'ens logiske operationer, f.eks. **AND**.

Man kan ligeledes lægge tal fra registrene og ud i **RAM** igen. Der ud over findes der mulighed for at "branche" eller hoppe, det er **B**-instruktionen, en slags **GOTO**, der får programudførelsen til at fortsætte et andet sted i hukommelsen. Dertil kommer **SWI** (**SoftWareInterrupt**), som er meget vigtig. **Archimedes**-computeren har indbygget en masse rutiner, der bl.a. bruges af operativsystemet. Ved hjælp af **SWI** kan man "kalde" disse underprogrammer, der så udfører nogle opgaver for en. Dette er en stor hjælp, da man derved er fri for at opfinde hjulet i hvert eneste program, bl.a. er man fri for at programmere de andre processorer i Archimedes:

**IOC** (I/O controller),  
**VIDC** (videocontroller) og  
**MEMC** (memorycontroller).

**R15** er et specielt register, da det hele tiden indeholder adressen på den næste instruktion, der skal udføres. Den kaldes for **PC** (program-counter). Når vi fra f.eks. Basic med **CALL** kører vores maskinkodeprogrammer, vil **R14** indeholde Basic's returadresse. Vi kan derfor afslutte vores program med instruktionen **MOV R15,R14**; det vil give programcounteren **R15** **R14**'s værdi, nemlig det sted, som **BASIC** ønsker at fortsætte fra, når vores program er færdigt.

Køb eller lån en komplet beskrivelse af **ARM**-assembler, da der er mange detaljer vi ikke vil komme ind på nu. Du vil finde, at **ARM**'en har et lille antal instruktioner som alle kan udføres med betingelser, "rotationer" og andet godt fra havet. Disse er i stor udstrækning selvforklarende, så vi vil bruge eksempler til at illustrere dem med og kun gå i detaljer med de mere specielle ting.

Senere artikler vil gå i dybden med **SWI**-kald, som **Arc**'en har i tonsvis af, datastrukturer (hvordan kan jeg lave dette udmærkede basic-program om til maskinkode?!), optimering af maskinkoden, kommatil i assembler, brugen af betingelser og "shifts" og sikkert meget mere.

Vi afslutter med et kommenteret eksempel; vi vil bestræbe os på at kommentere alle programmer, måske endog finde en eller anden standard.

God fornøjelse og forhåbentlig mange kommentarer til ideerne med denne nye spalte.

---

```
10 REM >ARMchars
20 REM 18/7 '89
30 REM Hans Otto Lunde
40 REM Fra Quercus-artikel om ARM-
maskinekode
50
```

```

60 REM Programmet viser forstørrelser af
   ascii-tegnene fra 32 til 255.
70 REM For hvert tegn kaldes en romrutine, der
   har en 9-bytes blok med
80 REM asciiværdien forrest som argument giver
   bitmønsteret for tegnet
90 REM I de næste 8 bytes. Disse bytes læses og
   for hver byte roteres de
100 REM enkelte bits ud i carry-flaget. Er bitten
   sat(CS) skrives et helt
110 REM udfyldt tegn(255) på skærmen, ellers et
   mellemrum. Tegnet opfoto-
120 REM graferes ved at lave sådanne 8 linier og en
   tom linie inden næste
130 REM tegn.
140
150
160 DIM code% 200,L% -1
170 PROCassemble
180 VDU 23,255,255,255,255,255,255,255,255
190 MODE 4
200 OFF
210 CALL code%
220 END
230
240
250 DEF PROCassemble
260
270 pc=15      :REM særlige navne for R14-R15
280 link=14
290
300 line_feed=10 :REM ascii-konstanter
310 car_return=13
320 blank=32
330 box=255
340          :REM SWI-konstanter
350 WriteC=0  :REM skriv tegn i R0 til skærm
360 OS_Word=7 :REM generelt kald
370 OS_Byte=6  :REM ---
380 OS_ReadEscape=44 :REM sæt carry, hvis
   <esc> trykket
390 V_Sync=19  :REM vent på fly-back
400 Char_Def=10 :REM læs en tegndefinition
410
420
430 FOR pass%=0 TO 3 STEP 3
440 P%=code%
450 [
460 OPT pass% OR %1000
470
480 MOV R7,#32      \ start med ascii(32)
490 .ascii_loop
500  ADR R1,definition \ R1 pointer til
   parameterblok
510  STRB R7,[R1]    \ Ønsket tegn forrest

520  MOV R0,#Char_Def
530  SWI OS_Word     \ hent bitmønster
540  ADD R1,R1,#1   \ peg på første byte af
   bitmønster
550  MOV R6,#8      \ 8 linier pr. tegn
560  .char_loop
570  LDRB R4,[R1],#1 \ læs byte og incrementer
   pointer
580  MOV R4,R4,LSL#24 \ byte øverst i R4
590  MOV R8,R1      \ gem pointer
600  MOV R0,#V_Sync
610  SWI OS_Byte    \ vent på "frame
   fly-back"
620  MOV R1,R8
630  MOV R5,#8      \ klar til at teste 8 bits
640  .row_loop
650  MOVS R4,R4,LSL#1 \ roter bit ud i
   carry-flag
660
670  MOVCC R0,#blank \ vælg blank, hvis
   bit'en ikke er sat
680  MOVCS R0,#box  \ vælg kasse, hvis bit
   sat
690  SWI WriteC     \ skriv til skærm
700  SUBS R5,R5,#1  \ flere bits?
710  BNE row_loop
720  MOV R0,#car_return \ CR + linefeed inden
   ny række
730  SWI WriteC
740  MOV R0,#line_feed
750  SWI WriteC
760  SUBS R6,R6,#1  \ flere linier i tegn?
770  BNE char_loop
780  SWI WriteC     \ linefeed før nyt
   tegn
790  SWI OS_ReadEscape
800  MOVCS pc,link  \ tidligt brugerstop?
810  ADD R7,R7,#1
820  CMP R7,#255   \ er alle tegn
   forstørret?
830  BNE ascii_loop
840  MOV pc,link
850
860  .definition
870  EQUB 0
880  EQUW 0
890  EQUW 0
900
910 ]
920 NEXT pass%
930 ENDPROC

```

Artikelserien fortsættes i næste nummer af Egebladet.

# C-kursus

Jens Ovesen og Kim Latt begynder her på deres C-kursus.

C er et sprog udviklet i slutningen af 60'erne af Kernighan og Ritchie, begge fra AT&T, en af de store computergiganter i USA. Sproget, som er en videreudvikling af BCPL, blev udviklet med det formål for øje at skrive UNIX-operativsystemet. Sprogets store popularitet stammer også heraf; der er ingen grænser for alt det "gris" man kan lave i C. Nogle programmører sammenligner C med en avanceret macro-assembler. Det er dog et højniveau-sprog, som KAN skrives meget struktureret og overskueligt. Denne artikelserie vil efter bedste evne forsøge at lægge vægt på det sidstel Serien, som vil strække sig over nogle udgaver af Egebladet, kan læses med lige stort udbytte for så vel BBC- og Archimedes-ejere. For at artikelserien ikke skal blive for lang, forudsættes det at man har et godt kendskab til BBC-BASIC.

C er i modsætning til BASIC et kompileret (oversat) sprog. D.v.s. at kildeteksten (programmet) oversættes en gang for alle til maskinkode af en kompiler. BASIC er jo en fortolker, der oversætter programmet til maskinkode efterhånden som det læses. I praksis betyder det at en fortolker er meget langsommere, idet den vil oversætte den samme linie mange gange. Eksempel, betragt følgende klassiske program i BASIC:

```
10 PRINT"Quercus er en god brugerklub"  
20 GOTO 10
```

Her vil fortolkeren først oversætte linie 10 til maskinkode, herefter udføre linien, gå til linie 20, oversætte den, udføre den, oversætte linie 10 etc. En tænkt BASIC-compiler ville oversætte programmet een

gang for alle til M/C, dernæst skulle denne objektkode køres. Objekt-koden kunne se sådan ud:

```
.linie10  
LDY #0  
.loop  
LDA streng,Y  
JSR OSASC!  
INY  
CPY #længde  
BNE loop  
.linie20  
JMP linie10  
.streng  
EQU$ "Quercus er en god brugerklub"+CHR$13
```

Det er i denne forbindelse ikke vigtigt at kunne forstå maskinkoden.

Lad os starte med et (meget) simpelt C-program:

```
#include <stdio.h>
```

```
int main()  
{  
    printf("Quercus er en god brugerklub\n");  
    return(0);  
}
```

Havelågen ("#") er en kommando til et specielt C-fænomen, nemlig præprocessor. Dette fine ord dækker over en funktion som kan sammenlignes med VIEW's macroer. I dette tilfælde vil standardbiblioteket **stdio.h** (STANDARD-InputOutput) blive flettet ind i kildeteksten. Når præprocessoren er færdig med at flette kildeteksten sammen, vil programmet blive kompileret.

I et hvert C-program skal der være funktionen **main()**. At det er en funktion tydeliggøres ved at sætte **int** foran funktionens navn. Herefter kommer {, dette betyder starten på en samling af ordrer (engelsk: compound-statement). Altså ligesom ved PASCAL's BEGIN-END-konstruktion. "printf" (print formatted) er en procedure fra **stdio.h** som skriver til skærmen.

"\n" laver et lineskift (ligesom ved **Acorn OS' [M]J**). Efter kommandoen skrives et ";". Dette skal altid skrives efter ikke reserverede kommandoer som f.eks. **"if"**. Efter som **main** er en funktion, returneres en værdi svarende til funktionshovedet, i dette tilfælde en integer, med **"return"**. **main** afsluttes slutteligt til slut med en tuborg slutparentes "}" (puhhh). Det skal til ";" bemærkes at i modsætning til **BASIC** afsluttes linier ikke med et vognretur, men med ";". Ovenstående program kunne derfor se sådan ud:

```
#include <stdio.h>
int main() {printf("Quercus er en god
brugerklub\n"); return(0);}
```

Det er dog sværere at læse. (Grimt!!!). Undtaget fra denne regel er dog præprocessor-kommandoer, som det ses af eksemplet.

Spændingen stiger til det uudholdelige, idet vi går over til C's løkkestrukturer:

```
#include <stdio.h>

#define ymax 10 /* definer en konstant "ymax=10"
*/
#define true 1
#define false 0
#define cls() putchar('V') /* definer en makro
"cls()"=CLS */

int main()
{
  int y; /* opret heltalsvariablen y */
  cls();
  printf("\nfor løkke: ");
  for (y=1;y<=ymax;y++) /* FOR Y%=1 TO ymax */
  {
    printf("%d ",y);/* Skriv y som heltal efterfulgt af
space */
  } /* NEXT */
  printf("\nrepeat løkke: ");
  y=1;
  do /* REPEAT */
  {
    printf("%d ",y);
    y++; /* Y%=Y%+1 */
  }
  while (y<=ymax); /* UNTIL Y%>ymax */
```

```
y=0;
printf("\nwhile løkke: ");
while (y<ymax) /* WHILE Y%<ymax */
{
  y++;
  printf("%d ",y);
}
y=1;
printf("\nloop løkke: ");
do /* LOOP */
{
  printf("%d ",y);
  if (y==ymax) break; /* Hop ud hvis Y%=ymax
*/
  y++;
}
while (true); /* ENDLOOP */
return(0);
}
```

Vi vil lige kort gennemgå de enkelte løkke-konstruktioners syntaks.

**For-sætningen :**

```
for (x=<start>;<slutbetingelse>;<step>)
{
  statements;
}
```

Bemærk at det der normalt ville følge efter **TO** i **BASIC**, er en meget stærkere konstruktion i **C**. I **BBC-BASIC** kører en **FOR** altid mindst én gang, det gør den ikke i **C**!

**Repeat-sætningen :**

```
do
{
  statements;
}
until (<slutbetingelse>);
```

Repeat kræver vist ingen yderligere uddybning.

**While-sætningen:**

```
while (<slutbetingelse>)
{
  statements;
}
```

For de fleste **BBC-BASIC** kendere er denne konstruktion ny. Den er konstruktionen man vælger hvis man vil have sin slutbetingelse checket først. Slutbetingelsen er for **repeat's** vedkommende **TRUE** og for **while's** **FALSE**.

#### Loop-sætningen:

```
do
{
  If (<exit-betingelse>) break;
}
while (TRUE);
```

En tilsvarende funktion findes ikke i **BASIC**.

Den ligeså unødvendig som den er grim. Kun dovne programmører bruger den!!!!

Vi kan konstatere, at **C** har alle tænkelige løkke-strukturer. Dog skal der lige knyttes en kommentar til **==**. En af de allerhyppigste fejl man kan lave, er at skrive **=** når man vil sammenligne to udtryk. **=** tildeler værdier og **==** sammenligner. Man gør klogt i hurtigt at skrive sig dette bag øret!!!

Vi kommer nu til **C's** operatorer :

Operator	Operation
++	Forøgelse med 1
--	Formindskelse med 1
*	Multiplikation
/	Division
+	Addition
-	Fortegnsminus eller
subtrak-	tion
%	Modulus (Rest ved division)
!	Logisk NOT
&&	Logisk AND
	Logisk OR
!=	Forskellig fra
==	Ens
>	Større end
>=	Større end eller lig med
<	Mindre end

**<=** Mindre end eller lig med

Dette er nok den sværeste del af **C** for nybegynderen at lære. Selv "udlærte" **C**-programmører laver ofte fejl her. Så fortvivl ikke hvis det ser mærkeligt ud. Vi har med vilje udeladt nogle af de mere eksotiske operatorer, men bare rolig, de kommer...

## Forgreninger ...

**C** har nogle meget stærke og potentielt meget farlige forgrenings-konstruktioner. Farligheden består i at, en smule uopmærksomhed mens man koder, kan fremprovokere subtile og svært gennemskuelige fejl.

Så hold derfor tungen lige i munden, når vi nu går videre!!

```
#include <stdio.h>
#define ymax 4
#define true 1
#define false 0
#define cls() putchar('\f')
int main()
{
  int y;
  cls();
  for (y=1;y<=ymax;y++)
  {
    if ((y%2)==0) /* "y MOD 2" i BASIC */
    {
      printf("%2d er lige\n",y);
    }
    else
    {
      printf("%2d er ulige\n",y);
    }
  }
  for (y=1;y<=ymax;y++)
  {
    switch (y%2)
    {
      case 0:
        printf("%2d er lige\n",y);
        break;
      case 1:
        printf("%2d er ulige\n",y);
        break;
    }
  }
}
```

# Hints og tips

## If-sætningen:

```
If (<betingelse>
{
  statements;
}
else
{
  statements;
}
```

If-konstruktionen er noget forskellig fra **BASIC IV**, men helt magen til **BASIC V** på **Archimedes'**en. Bemærk at **else**-delen ikke behøves. **THEN** er ukendt i C!

## Switch-sætningen (CASE):

```
switch (<udtryk>
{
  case <variabel=udtryk> :
    statements;
    break;
  .
  .
  case <variabel=udtryk> :
    statements;
    break;
  default : /* Otherwise */
    statements;
    break;
}
```

Det er overordenligt vigtigt at **HUSKE break'sne**. Hvis **break** ikke er der, vil alle **case-sætningerne** blive forsøgt matched. Det vil sige at, har man en **default-sætning** (Otherwise). Vil den blive brugt hvis man glemmer **break**.

Det var alt hvad i fik for den 25-øre!! Næste gang vil vi dels rette op eventuelle misforståelser, dels gå mere i dybden med det der eventuelt måtte være for svært, og fortsætte med funktioner,(procedurer) og scope. Er der noget I vil have at vi specielt kigger på, så skriv!!! Go' fornøjelse.

Hvis man f. eks benytter beregnede **RESTORE** sætninger, eller beregnede **GOTO** sætninger (FY!!), kan det være besværligt at man hele tiden skal lave om på beregningerne, når man indsætter og sletter linier i sit program, og omnummererer. Det funktionen gør, er at lede sit eget program igennem efter en **REM** sætning der har et bestemt tegn stående umiddelbart efter **REM** instruktionen. Når en sådan linie findes, returnerer funktionen det **NÆSTE** linienummer:

```
10010 DEF FNA(T$)
10020 LOCAL P%,T%,F%
10030 P%=-PAGE: T%=-ASCT$: F%=0
10040 REPEAT: IF P%?4<>244 THEN 10060
10050 IF P%?5=T% F%=1
10060 P%=P%+P%?3: UNTIL F%=1
10070 =P%?2+256*P%?1
```

Lad os sige at du har nogle **DATA** linier i linie 1320 i dit program. Du placerer så en **REM** sætning lige ovenover der ser sådan ud:

```
1310 REM1
1320 DATA 12,13,....
```

Bemærk at **REM** instruktionen skal stå som den første instruktion på linien. Man får så funktionen til at finde linie 1320 ved at skrive:

```
K%=FNA("1")
```

Man kan selvfølgelig udskifte 1 tallet efter **REM** sætningen med andre tal og tegn, men man skal være opmærksom på at man kun får returneret linienummeret efter **FØRSTE** forekomst af **REM** sætningen. Funktionen virker både på en **BBC** og en **Archimedes**.

Jan Vibe (6513)

# GIF-GAF

## Her beskriver Kim Latt nogle grafikformater ...

Som en følge af den store mangfoldighed der hersker på micro-computer området, er der et ligeså stort udvalg af grafikformater. Alle er de selvfølgelig inkompatible!!

Der er "GIF"-formatet udviklet af CompuServe i USA. "IFF" fra Electronic Arts, ligeledes fra USA. Til Workstations er der også en del mere eller mindre eksotiske formater, "TIFF" er et eksempel.

## GIF - Graphics Interchange Format ...

Dette format fra CompuServe er det mest udbredte indenfor PC-verdnen.

Formatet er i stand til at beskrive en **Bitmap** (et billede), således at grafik kan transporteres fra maskine til maskine.

Ideen er at et program, her **GIF-læseren**, tager en bitmap i en oplyst opløsning og oversætter

den til det grafikformat som målmaskinen har.

En analogi til dette findes i alle **Acorn** maskiner : Grafikvinduet i et hvilket som helst mode har altid opløsningen 1280 \* 1024. Operativ-systemmet sørger selv for at "oversætte" til den faktiske opløsning.

En **GIF**-fil har flg. oplysninger: Raster størrelse, antal farver (bitplanes), palettedata og forskellige data til nogle **GIF**-specialiteter.

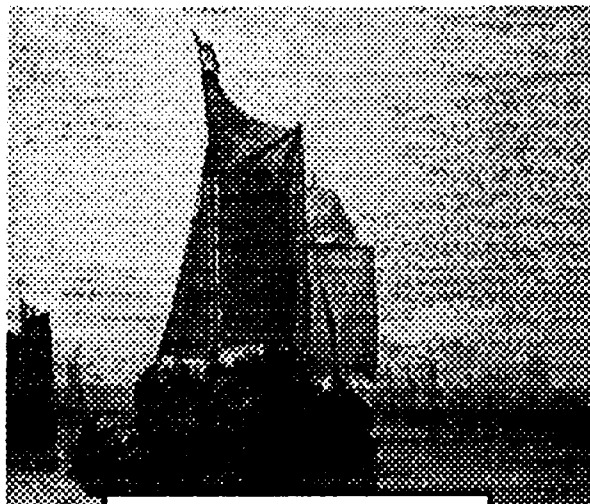
En af **GIF**-specialiteterne er:

Alle **GIF** billeder er pakket med en "**State of the art**" (**Creme de la creme**) metode. Metoden hedder "**Lempel-Ziv-Welch compression**" (**LZW-compression**), forklaring af metoden vil jeg ikke give, idet den

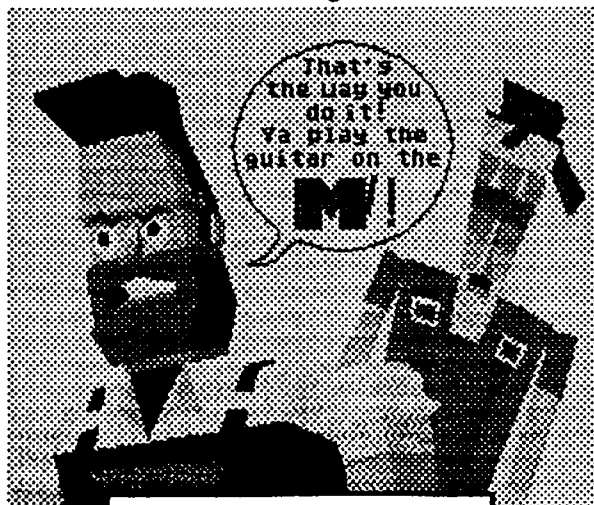
er vanskelig at forklare på lidt plads. Det skal bare bemærkes at metoden er som skabt til billeder, idet den er lige effektiv på mønstrede såvel som på ensfarvede billeder. Har man et modem og har man "**downloadet**" fra **BBS'er**, kender man utvivlsomt et eller flere arkiveringsprogrammer, såsom :**ARC, LHARC, PKXARC,**

**ZIP** etc., her bruges **LZW** også. **LZW** er så godt, at de meget dyre "**professionelle**" modemer bruger **LZW** til at vride højere hastighed ud af telefonledningen. **GIF** har nogle begrænsninger, som **Acorn**-ejere kan være ligeglade med. 256 farver er jo da mere end rigeligt, ikk'?

Til Workstations som



"Oldship" -et 256 farve GIFbil-



"MTV"- et 64 farve IFFbillede

f.eks. **SUN, Apollo** o.a. findes der formater som dækker op til **16777216** farver på een gang.

## IFF-Interchange File Format ...

Formatet som stammer fra **Electronic Arts**, er overvejende brugt til **Commodores AMIGA**'en og i mindre grad til **ATARI ST**'en.

Formatet har ikke samme begrænsninger som **GIF** idet man kan specificere op til **255** bitplanes, altså **5.7896\*10<sup>76</sup>** farver!!

Formatet er ikke kun tænkt til grafik, men dækker også musik, tekst, samples, fontdefinitioner, sequences og meget mere.

En **IFF-fil** består af nogle **"CHUNKS"**. Disse chunks, som er på 32-bit eller 4 tegn, signalerer hvad det efterfølgende data skal opfattes som.

Disse **"CHUNKS"** har **mnemotekniske** navne som f.eks. **ILBM** (InterLeaved BitMap), **ACBM** (Amiga Continuous BitMap), **SMUS** (Simple MUSIC) etc. og er af **Electronic Arts** beskrevet som en standard. **IFF** er en god og meget fremtidssikret standard, der er dog en indbygget svaghed.

**Electronic Arts** havde tænkt sig, at hvis softwarehuse fik brug for en ny **"CHUNK"**-descriptor, så ansøgte man om at få et unikt **"CHUNK"**-navn. Fordelene herved er selvfølgelig at man sikrer sig mod tvetydige definitioner og et hav af ukendte **"CHUNKS"**!! Dette har netop vist sig at være et problem. Når en programmør eller et softwarehus skal bruge en ny **"CHUNK"**-descriptor så laver de da bare en!

Når **"CHUNK"**'en hedder **ILBM** er der tale om grafik. Grafikken kan, som for **GIF**'s vedkommende, være pakket. Metoden til pakningen er meget simpel, den hedder **"Run-Length-Coding"** og er meget nem at forklare. Så derfor vil jeg lige kort skitsere metoden. Forestiller man sig nogle bytes fra skærm-lageret, så vil metoden tælle

hvor mange af disse der er ens i træk og gemme denne information som **<antal>** og **<byte>**.

Eksempel:

```
AA AA AA 44 11 11 11 11
```

Denne sekvens af hexadecimale tal vil af **"Run-Length-Coding"** blive pakket til :

```
03 AA 01 44 04 11
```

Som det kan ses er der ikke tale om nogen stor besparelse (ca. 25%), det er ikke svært at forestille sig at metoden kan blive meget dårlig hvis data f.eks. så sådan ud :

```
AA 55 AA 55 AA 55 etc. (et mønster i MODE 0)
```

Dette ville jo blive pakket til :

```
01 AA 01 55 01 AA osv., hvilket giver en besparelse på -100 %!!
```

**Electronic Arts** udgave af **"Run-Length-Coding"** har nogle finesser som gør metoden lidt bedre.

## Archie og GIF-IFF-grafik ...

Vi (**Brain Madsen** og **Kim Latt**) har i fællesskab brugt en del tid på at lave programmer, der kan læse både **GIF**- og **IFF**-filer. Disse programmer og forskellige billeder (**EGA**, **VGA** og **HAM**) + et 4096 farve modul, har vi besluttet os for at donere til **Quercus**.

Tanken er, at I kan købe et sæt klub-disketter med udpakke-programmerne samt billederne på, og det overskud der evt. måtte komme vil komme klubbens slunkne pengekasse til gode.

Er man fascineret af grafik, er dette **DISKETTERNE** at investere i!!!

Prisen for **3 Mb** billeder, d.v.s. 4 disketter,

bliver 80 kr. Prisen inkluderer selvfølgelig disketterne og forsendelse.

## BBC og GIF-IFF-grafik ...

Der er på indeværende tidspunkt ikke noget program til BBC'en. Har man imidlertid fået blod på tanden og har lyst til at skrive et program, kan jeg levere nødvendig dokumentation og hjælp.

Videregående læsning, som alt sammen kan lånes af mig :

### Om GIF:

#### Graphics Interchange Format (tm)

*"A standard defining a mechanism for the storage and transmission of raster-based graphics information"*

CompuServe, Incorporated. An H&R Block Company  
5000 Arlington Centre Blvd.  
Columbus, Ohio 43220

### Om Lempel-Ziv-Welch:

Terry A. Welch,

*"A Technique for High Performance Data Compression"*

IEEE Computer, vol 17 no 6 (June 1984)

### Om IFF:

*"EA IFF 85"*

Standard for Interchange Format Files  
Jerry Morrison, Electronic Arts

# Fraktaler

## H. Pedersen (6577) om fraktaler ...

Lidt om Fraktaler

Prøv nedenstående lille program, men vær forberedt på at lade computeren regne i nogen tid.

Programmet tegner fraktaler og er ca. 30 min om at blive færdig, på en Archimedes.

Måske skulle jeg lige forklare hvad fraktaler er.

De fleste har nok set billeder af fraktaler i aviserne eller på TV, men der har der aldrig været noget om gør-det-selv-fraktaler, så det vil jeg prøve at råde bod på.

Humlen i fraktaler er den rekursive formel  $Z(n+1)=Z(n)^2+C$ , hvor  $Z$  og  $C$  er komplekse tal, d.v.s. at istedet for at ligge på en linie som almindelige reelle tal gør, ligger de i et koordinatsystem og har altså to værdier tilknyttet sig; en for x-aksen og en for y-aksen, henholdsvis reel- og imaginærdelen.

Fraktalerne fremkommer så, når man for ethvert  $C$  i et passende område af koordinatsystemet afbilder hvor hurtigt formlen går mod uendeligt.

Ser man på programmet er dette måske lettere at forstå:

Linie 10 giver sig selv,

linie 20 bestemmer hvor stort billedet skal være,

30 springer vi over,

linierne 40 og 50 angiver hvor i koordinatsystemet vi lader  $C$  variere. Nederste venstre hjørne er her  $(-2.3, -1.4)$ , øverste højre hjørne er  $(0.8, 1.4)$

60 og 70 udregner hvor stor hver pixel på skærmen er

**FOR-NEXT-løkkerne** i 80 og 90 gennemløber alle pixels i det område af skærmen vi afsatte i linie 20.

I linie 100 til 130 beregnes  $C$ 's reel- og imaginærdel ( $P$  og  $Q$ ),  $Z(0)$  sættes lig 0 ( $X_0$  og  $Y_0$ ) og iterations-tælleren  $n$  sættes også til 0.

I løkken fra 140 til 190 beregnes  $Z(n+1)$  ( $X_0$  og  $Y_0$  i linie 160 og 170) indtil  $n=100$  eller  $Z(n)$  er uendeligt stor ( $Z(n)$  regnes for uendeligt stor hvis  $X_0^2+Y_0^2$  er større end 10).

(Artiklen fortsættes på side 34)

# 6502 MC-kursus

## Torben Mogensen afslutter sit maskinkodekursus ...

### 7. del.

Operativsystemkald.

De programmer vi hidtil har skrevet har været kaldt fra et **BASIC** program, der gav inddata og brugte uddata. Ofte er det dog mere interessant at lade maskinkodeprogrammet selv sørge for dette, f.eks. hvis maskinkodeprogrammet skal lave grafik. Til dette formål findes der nogle operativsystemkald til inddata og uddata. Derudover findes der kald, der svarer til \*-kommandoer så som **\*FX**, **\*LOAD** osv.

Det er også muligt at indsætte maskinkode, der bliver kaldt hver gang der sker noget bestemt i maskinen, f.eks. hver gang der skrives et tegn ud eller hver gang der er lodret sync på skærmen.

### Inddata og Uddata.

De vigtigste kald til uddata er **OSWRCH** (O.S. **WR**ite **CH**aracter) og **OSASCI** (O.S. **ASCI**II). begge disse kald vil skrive det tegn ud som står i register **A**. **OSWRCH** svarer fuldstændigt til **BASICs** **VDU** kommando, og gør det dermed muligt at lave al slags grafik. **OSASCI** gør næsten det samme som **OSWRCH**, men laver et lineskift når den skriver kode 13 ud, hvor **OSWRCH** bare rykker til begyndelsen af linien.

Til inddata kan **OSRDCH** (O.S. **ReaD** **CH**aracter) bruges. Den fungerer som **GET** i **BASIC**, mere præcist læser den et tegn fra tastaturet og lægger det i **A**.

Følgende lille program vil læse tegn fra tastaturet og skrive dem to gange, indtil der bliver trykket return.

```
.togange
JSR OSRDCH
JSR OSWRCH
JSR OSASCI
CMP #13
BNE togange
```

Ved at bruge **OSWRCH** den ene gang og **OSASCI** den anden, bliver der lavet ét lineskift når der bliver trykket return.

### \*-Kommandoer.

Et kald til **OSBYTE** svarer til **\*FX A,B,C**. Dvs. at

```
LDA #&90
LDX #0
LDY #1
JSR OSBYTE
```

svarer til **\*FX &90,0,1**, som jo er **\*TV 0,1**.

I **Users Manual** er der en liste over de vigtigste **OSBYTE** kald. Der er nogle af kaldene, der returnerer en værdi, og som derfor ikke kan bruges som **\*FX** kald. Dette gælder f.eks. **OSBYTE** med **A=&79**. Denne svarer til **INKEY(-x)**, dvs. hvis register **X** har en negativ værdi (**&80-&FF**), så testes der for tasten med den værdi. Ved retur er **X** negativ hvis tasten er trykket ned, ellers er **X** positiv. Hvis **A=&81** svarer et kald til **OSBYTE** til **INKEY** med positiv parameter. **X** og **Y** specificerer en tid i centisekunder (**X=tid MOD &100**, **Y=tid DIV &100**). Ved retur er **Y=0**, **C=0**, **X=ASCII** kode hvis en tast er trykket indenfor tiden, ellers er **Y=&FF**, **C=1**.

Et andet vigtigt kald er **OSCLI** (O.S. **Comm**and **Line** **Interpre**ter). Dette fungerer præcis som **BASIC** lls **OSCLI** kommando, dvs. det sender en tegnfølge til operativsystemets kommando fortolker. Alle kommandoer, der kan laves som \*-kommandoer kan også laves med **OSCLI**. Ved et kald peger **XY** på en tegnfølge, som er afsluttet med

ASCII kode 13. (X=adresse MOD &100, Y=adresse DIV &100). Følgende program vil læse kommandoer fra tastaturet og sende dem til operativsystemet indtil en tom linie tastes ind. Der testes ikke for om der tastes for mange tegn ind, men en sådan test kan let indføres.

```
.kommandoer
LDX #0
.laes \ læs tegn
JSR OSRDCH
JSR OSASCI
STA tegnfoelge,X
INX
CMP #13
BNE laes \ stop ved return
CPX #1
BEQ slut \ slut hvis linie er tom
LDX #(tegnfoelge MOD &100)
LDY #(tegnfoelge DIV &100)
JSR OS_CLI
JMP kommandoer
.slut RTS
.tegnfoelge \ her skal der være plads til tegnene
EQU STRING$(128,CHR$(0))
```

Jeg bruger navnet OS\_CLI fordi OSCLI er et BASIC kommandonavn.

## Vektorer.

Operativsystemet bruger tit indirekte hop igennem såkaldte vektorer i stedet for at gå direkte til en rutine. Det gør det muligt at lægge en ny adresse ind i vektoren, så ens egen rutine bliver kaldt i stedet for.

Den normale måde at bruge vektorer på, er at man laver en rutine, der kalder den oprindelige rutine, men gør et eller andet før eller efter kaldet.

Følgende rutine ændrer alle blanktegn til underlinier når de skrives ud. Der er to dele i rutinen: en initialiseringsdel, der ændrer vektoren og en del, der bliver kaldt hver gang der skrives et tegn ud.

```
.initialiser
SEI \ luk for afbrydelser
LDA WRCHV \ kopier gamle vektor
```

```
STA gammel_vektor
LDA WRCHV+1
STA gammel_vektor+1
LDA #(skift_blank MOD &100) \ indsæt ny
STA WRCHV
LDA #(skift_blank DIV &100)
STA WRCHV+1
CLI \ tillad afbrydelser igen
```

RTS

```
.gammel_vektor
EQUW 0
.skift_blank \ udføres ved hvert kald af OSWRCH
CMP #ASC(" ")
BNE hop_videre
LDA #ASC("_")
.hop_videre
JMP(gammel_vektor)
```

Efter et kald til initialiser, vil skift\_blank blive kaldt hver gang et tegn skrives ud. Her vil alle blanktegn blive ændret til understreg før de sendes videre til den oprindelige rutine. Der er ikke taget højde for at ASCII koden for blanktegn kan forekomme som parameter til f.eks. PLOT. Brugen af SEI og CLI i initialiseringsdelen sikrer at der ikke er afbrydelser mens vektoren bliver skiftet. Afbrydelser bliver ikke behandlet i dette kursus, da det meste af det man kan med afbrydelser kan gøres lettere med vektorer.

Hvis man vil gøre noget efter den almindelige rutine er færdig, f.eks. hvis man vil ændre indlæste blanktegn til understreger kan det gøres som vist her. Initialiseringsdelen er udeladt, da den er magen til den fra sidste eksempel på nær at det er RDCHV der bruges i stedet for WRCHV.

```
.skift_blank
JSR hop_videre
BCC ok
RTS \ retur hvis fejl
.ok
CMP #ASC(" ")
BNE retur
LDA #ASC("_")
.retur
CLC \ slet fejl
RTS
.hop_videre
JMP(gammel_vektor)
```

Brugen af **JSR** før kaldet til gammel\_vektor sikrer at man vender tilbage når den almindelige rutine er færdig. Man skal så afslutte med en **RTS**, så man vender tilbage til hvor **OSRDCH** blev kaldt. Behandlingen af **C** flaget er til for at sikre at fejlstatus bliver bevaret.

En anden interessant vektor er **Event**-vektoren. Den bruges når forskellige situationer opstår, f.eks. når der er lodret sync på skærmen eller når der bliver trykket på **escape**. Indholdet af register **A** viser hvilken hændelse der var årsag til kaldet:

- A=0 : Uddata buffer er tom
- A=1 : Inddata buffer er fuld
- A=2 : Nyt tegn i inddata buffer
- A=3 : **ADC** konvertering færdig
- A=4 : Lodret sync
- A=5 : Uret passerer 0
- A=6 : **Escape** er trykket
- A=7 : **RS432** fejl
- A=8 : **Econet** fejl
- A=9 : Bruger defineret

En event rutine fungerer ligesom de vektor rutiner der er vist ovenfor. Hvis **A** er et andet tal end det man venter på kalder man den gamle vektor med det samme, ellers gør man et eller andet først.

Når man har ændret en vektor, skal man sikre sig at den ikke er ændret igen siden, før man forsøger at ændre den tilbage. Man skal også passe på at man ikke bruger for meget tid i sine rutiner. Specielt gælder det event rutiner, som helst ikke må bruge mere end et par millisekunder.

## Adresser.

Navnene **OSWRCH**, **WRCHV** osv. er ikke foruddefinerede i **BASIC**, så de skal defineres i starten af de programmer der bruger dem. Værdierne er givet her:

<b>OSWRCH</b>	= &FFEE
<b>WRCHV</b>	= &20E
<b>OSASCI</b>	= &FFE3
<b>OSRDCH</b>	= &FFE0
<b>RDCHV</b>	= &210
<b>OS_CLI</b>	= &FFF7
<b>CLIV</b>	= &208
<b>OSBYTE</b>	= &FFF4
<b>BYTEV</b>	= &20A
<b>EVENTV</b>	= &220

Dette er så slutningen på kurset. Hvis man er interesseret i at lære mere, findes der mange bøger om maskinkodeprogrammering på **BBC** mikroen. En anden god kilde er "**The Advanced User Guide for the BBC Microcomputer**", hvorfra jeg har hentet informationerne om vektoradresser osv. Den fortæller også om hvordan man kan definere nye skærmmoder ved at programmere video kredsen og lignende ting.

## PCX TALEKORT

Lad **PCX** passe din telefon med klar naturlig tale.

Det menustyrede danske program kan automatisk ringe op til **OPS**-beeper eller mobiltelefon og meddele, at der er besked.

**PCX** kan også fungere som automatiseret ordremodtagelse og samtidig fortælle, om de ønskede varer er på lager.

**PCX** er et halvlængdeudvidelseskort til **IBM**-kompatible **PC**'ere.

Ring og hør nærmere om den nye slagkraftige løsning på telefonpasningen.

**CFM** Lundegaardsvej 7  
computing 5672 Broby  
Tlf. 64 75 10 03

# REX DATA

## HOLDER FLYTTEDAG

LØRDAG DEN 29. JULI HOLDER VI RECEPTION I NYE LOKALER PÅ ADRESSEN:

**BJERREGÅRDSVEJ 4, VRIDSTED, 7800 SKIVE**

Hermed inviteres alle til at fejre dagen sammen med os, idet der fra 13<sup>00</sup>-16<sup>00</sup> vil være åbent hus.

### Specielle åbningstilbud:

**Acom A3000 - den nye RISC computer (vil blive demonstreret ved receptionen): 9425**

**1st Word Plus - tekstbehandling til Archimedes: 795**

**Arcade 3 - spil til Archimedes: 155**

**Play It Again Sam 9 - spil til BBC/Master: 125**

Vi kan skaffe næsten alt til de forskellige Acorn computere.  
Ring evt. og hør nærmere. Ting, vi ikke har  
på lager i øjeblikket kan skaffes hjem fra England i løbet af kort tid.

OBS: Komplet pris-liste med end 500 titler kan rekvireres.  
Vi glæder os til at handle med netop DIG.

Med venlig hilsen, for **Rex Data I/S**  
**Flemming Jørgensen, Kresten Brelner, Olav Christlansen**

### REX DATA I/S:

**Butik/værksted:**  
**Bjerregårdsvej 4, Vridsted, 7800 Skive**  
**Tlf. 97 54 70 15 Fax: 97 54 70 19**

**Administration:**  
**Sundvej 34, Hvalpsund, 9640 Farsø**  
**Tlf. 98 63 81 50 Fax: 98 63 86 68**

# Listninger

## Labyrint-generator.

Her er en lille maskinkode-rutine(??), som kan generere en tilfældig labyrint med præcis een vej fra ethvert sted til ethvert andet sted i labyrinten.

Hvis man lægger mærke til, hvordan dimensionerne bruges, vil man se, at de kun bruges til at tegne en firkant med, og det vil altså også sige, at man med denne rutine vil kunne lave en labyrint i alle mulige mystiske figurer (f.eks en cirkel). Startpunktet for labyrinten sættes i linie 130. Det skal dog dertil bemærkes, at figuren skal være helt tom for 'huller', ellers vil der ikke komme nogen gang der. (Pas på med 'hjemmelavede' cirkler på B'eren.)

Jeg håber, at der er nogen, der kan bruge det til noget, hvad med f.eks. at vade rundt i 3D og real-time i en 50x50'er? Eller starte med nogen smart placerede gange, så man, når man er færdig, har skrevet 'God jul'?

God fornøjelse.  
Thomas Olsson.

```
10 MODE 4
20 INPUT "X,Y dimensions ? (2-79,2-63)";XD%,YD%
30 MODE 4
40 VDU23,1;0;0;0;0
50 MCS%=&380
60 I&74=&3000
70 DIM MC% MCS%,ST% I&74
80 PROC MC
90 IF RND(-TIME)
100 ISCR%=&RND
110 SCR%I2=&RND
120 MOVE0,0:MOVE0,YD%*16+4
125 PLOT85,XD%*16+4,0
126 PLOT85,XD%*16+4,YD%*16+4
130 X%=&8:Y%=&8:CALLMAZ%
140 VDU7
150 END
160 DEF PROC MC
170 SP%=&76
```

```
180 FORI%=&0TO2STEP2
190 P%=&MC%
200 [OPT1%
210 .PSH%
220 LDASP%+1:CMPI&75:BCCOV%
230 LDASP%:CMPI&74:BCCOV%
240 BRK:EQUB1:EQUStack full":BRK
250.OV%CLC:LDASP%:ADC#ST%AND255:STA&70
260 LDASP%+1:ADC#ST%DIV256:STA&71
270 LDY#0:LDA&78:STA(&70),Y:INY
271 LDA&79:STA(&70),Y
275 INY:LDA&7A:STA(&70),Y:INY:LDA&7B:STA(&70),Y
280 CLC:LDASP%:ADC#4:STASP%
290 LDASP%+1:ADC#0:STASP%+1
300 .OUT% RTS
310 .PUL%
320 LDASP%:BNEOV1%
330 LDASP%+1:BNEOV1%
340 BRK:EQUB2:EQU"Empty stack":BRK
350 .OV1% SEC:LDASP%:SBC#4:STASP%
360 LDASP%+1:SBC#0:STASP%+1
370 CLC:LDASP%:ADC#ST%AND255:STA&70
380 LDASP%+1:ADC#ST%DIV256:STA&71
390 LDY#0:LDA(&70),Y:STA&78:INY
392 LDA(&70),Y:STA&79:INY
395 LDA(&70),Y:STA&7A:INY:LDA(&70),Y:STA&7B
400 RTS
410 .RAN%
420 SEC:LDASCR%+1:ADCSCR%+4
425 ADCSCR%+5:STASCR%
430 LDX#4
440 .LP2% LDASCR%,X:STASCR%+1,X
450 DEX:BP LLP2%
460 RTS
470 .SCR%EQUStack"RANDOM"
480 .MAZ%
490 LDA#0:STA&78:STA&79:STA&7A:STA&7B
500 JSRPSH%
510 LDA&460:STA&78:LDA&461:STA&79:LDA&464
515 STA&7A:LDA&465:STA&7B
520 .MLP%
530 JSRRAN%
540 AND#3:TAX
550 DEX:BMIMU%
560 DEX:BMIMD%
570 DEX:BMIMR%
580 JMPML%
590 .MU%JMP_MU%
600 .MD%JMP_MD%
610 .ML%
620 SEC:LDA&78:SBC#16:STA&80
630 LDA&79:SBC#0:STA&81
640 LDA&7A:STA&82:LDA&7B:STA&83
650 JSRPNT%
660 CMP#1:BEQOK4%
```

670 JMPCOD%  
 680 .OK4%  
 690 LDA#25:JSR&FFEE:LDA#4:JSR&FFEE  
 692 LDA&78:JSR&FFEE:LDA&79:JSR&FFEE  
 695 LDA&7A:JSR&FFEE:LDA&7B:JSR&FFEE  
 700 LDA#25:JSR&FFEE:LDA#99:JSR&FFEE  
 702 LDA#&F0:JSR&FFEE:LDA#&FF:JSR&FFEE  
 705 LDA#4:JSR&FFEE:LDA#0:JSR&FFEE  
 710 LDA&80:STA&78:LDA&81:STA&79  
 720 JSRPSH%  
 730 JMPMLP%  
 740 .MR%  
 750 CLC:LDA&78:ADC#16:STA&80  
 760 LDA&79:ADC#0:STA&81  
 770 LDA&7A:STA&82:LDA&7B:STA&83  
 780 JSRPNT%  
 790 CMP#1:BEQOK3%  
 800 JMPCOD%  
 810 .OK3%  
 820 LDA#25:JSR&FFEE:LDA#4:JSR&FFEE  
 822 LDA&78:JSR&FFEE:LDA&79:JSR&FFEE  
 825 LDA&7A:JSR&FFEE:LDA&7B:JSR&FFEE  
 830 LDA#25:JSR&FFEE:LDA#99:JSR&FFEE  
 832 LDA#20:JSR&FFEE:LDA#0:JSR&FFEE  
 835 LDA#4:JSR&FFEE:LDA#0:JSR&FFEE  
 840 LDA&80:STA&78:LDA&81:STA&79  
 850 JSRPSH%  
 860 JMPMLP%  
 870 .\_MD%  
 880 SEC:LDA&7A:SBC#16:STA&82  
 890 LDA&7B:SBC#0:STA&83  
 900 LDA&78:STA&80:LDA&79:STA&81  
 910 JSRPNT%  
 920 CMP#1:BEQOK2%  
 930 JMPCOD%  
 940 .OK2%  
 950 LDA#25:JSR&FFEE:LDA#4:JSR&FFEE  
 952 LDA&78:JSR&FFEE:LDA&79:JSR&FFEE  
 955 LDA&7A:JSR&FFEE:LDA&7B:JSR&FFEE  
 960 LDA#25:JSR&FFEE:LDA#99:JSR&FFEE  
 962 LDA#4:JSR&FFEE:LDA#0:JSR&FFEE  
 965 LDA#&F0:JSR&FFEE:LDA#&FF:JSR&FFEE  
 970 LDA&82:STA&7A:LDA&83:STA&7B  
 980 JSRPSH%  
 990 JMPMLP%  
 1000 .\_MU%  
 1010 CLC:LDA&7A:ADC#16:STA&82  
 1020 LDA&7B:ADC#0:STA&83  
 1030 LDA&78:STA&80:LDA&79:STA&81  
 1040 JSRPNT%  
 1050 CMP#1:BEQOK1%  
 1060 JMPCOD%  
 1070 .OK1%  
 1080 LDA#25:JSR&FFEE:LDA#4:JSR&FFEE  
 1081 LDA&78:JSR&FFEE:LDA&79:JSR&FFEE

1082 LDA&7A:JSR&FFEE:LDA&7B:JSR&FFEE  
 1090 LDA#25:JSR&FFEE:LDA#99:JSR&FFEE  
 1092 LDA#4:JSR&FFEE:LDA#0:JSR&FFEE  
 1095 LDA#20:JSR&FFEE:LDA#0:JSR&FFEE  
 1100 LDA&82:STA&7A:LDA&83:STA&7B  
 1110 JSRPSH%  
 1120 JMPMLP%  
 1130 .COD%  
 1140 CLC:LDA&7A:ADC#16:STA&82  
 1150 LDA&7B:ADC#0:STA&83  
 1160 LDA&78:STA&80:LDA&79:STA&81  
 1170 JSRPNT%  
 1180 BEQNT1%:BMINT1%  
 1190 JMPMLP%  
 1200 .NT1%  
 1210 SEC:LDA&7A:SBC#16:STA&82  
 1220 LDA&7B:SBC#0:STA&83  
 1230 JSRPNT%  
 1240 BEQNT2%:BMINT2%  
 1250 JMPMLP%  
 1260 .NT2%  
 1270 CLC:LDA&78:ADC#16:STA&80  
 1280 LDA&79:ADC#0:STA&81  
 1290 LDA&7A:STA&82:LDA&7B:STA&83  
 1300 JSRPNT%  
 1310 BEQNT3%:BMINT3%  
 1320 JMPMLP%  
 1330 .NT3%  
 1340 SEC:LDA&78:SBC#16:STA&80  
 1350 LDA&79:SBC#0:STA&81  
 1360 JSRPNT%  
 1370 BEQNT4%:BMINT4%  
 1380 JMPMLP%  
 1390 .NT4%  
 1400 JSRPUL%  
 1410 LDA&78:BEQFIN%  
 1420 JMPMLP%  
 1430 .FIN% RTS  
 1440.PNT%LDX#&&80:LDY#0  
 1442 LDA#9:JSR&FFF1:LDA&84:RTS  
 1450 ]  
 1460 NEXT  
 1470 ENDPROC

Der vil i næste nummer af **Egebladet** være en artikel om strukturering og dokumentation af programmer. Denne artikel markerer samtidigt, at en ny type listninger vil præge bladet fremover. Der vil for fremtiden være nogle krav, listninger skal opfylde for at blive optaget. En rodet listning **KAN** ikke illustrere noget som helst.

# Annoncer

## TIL ARCHIMEDES KØBES :

PC-Emulatoren, en **Basic-compller** og en rigtig god tekstbehandling.

**Henrik Pedersen (6577)**  
Kolleglebakken 9,4009  
2800 Lyngby  
42 88 15 55 - vær 4009

## Til BBC'en sælges.

1 stk **Viglen Professional Console Unit (PCU)** til BBC model B. Pris 200 kr. + forsendelse.

1 stk **Solldisk Dual Floppy Disc Controller**. Består af en **WD1770** disc controller + kontakt til skift mellem de 2 disksystemer. Ydermere medfølger **DFS 2.1A** og **ADFS 2.1**. Pris 350 kr.

**ISO Pascal fra Acornsoft**. Består af 2 stk ROM plus 2 bøger. Pris 250 Kr.

**The Publisher fra PMS**. Pris 200 kr.

**(6503) Martin Søndergaard**  
Årestrupvej 76  
7470 Karup J  
Tlf. 97 10 21 95

## BBC B sælges.

**BBC B** med **WD1770+8271**, **Watford 32K** shadowmemory, 5,25" diskteststation, meget god **NEC 12"** farvemonitor, **AMX-mus** og diverse ragelse (software etc) sælges. Pris 2500,-

**(6545) Jens H. Ovesen**  
Tolderlundsvej 27, 4. tv.  
5000 Odense C.  
Tlf. 65 91 75 11

## Nightingale modem sælges.

**Nightinale** modem sælges, modificeret til **CCITT** og **Bell**. Kun 500,-

Henvendelse formanden (Valo).

# Addresseliste

## ADRESSEÆNDRINGER

6512  
Programmør  
Rene Nørskov  
Horseklildevej 16, st.th.  
2500 Valby  
Tlf. 31 17 93 29

6515  
Ingeniør  
Kristinn K. Sigurjonsson  
Romdrupvej 62  
9270 Klarup  
Tlf. 98 31 86 02

6511  
Elektronikmekaniker  
Ole Nebel Nielsen  
Lindegade 5B  
6823 Ansager

6545  
Stud.Scient.Tech.  
Jens H. Ovesen  
Tolderlundsvej 27, 4. tv.  
5000 Odense C  
Tlf. 65 91 75 11

6553  
Rune Christlansen  
Terpvej 69  
6630 Rødding

6587  
Senlorsergent  
Kresten Breiner  
Bjerregårdsvej 4  
Vridsted  
7800 Skive  
Tlf. 97 54 70 15

Følgende numre er slettet af medlemslisten:

6517 - 6520 - 6534 - 6550 - 6555 - 6572.

## Nye medlemmer.

Indmeldt i perioden 20/4 - 1/7, 1989

6591

**Programmør**  
**Brian Madsen**  
**Søvangen 3**  
**2635 Ishøj**  
**Tlf. 42 73 67 44**

6592

**Cand. Jur.**  
**Morten Kamp**  
**Næstvedgade 35, 4.tv.**  
**2100 København Ø**  
**Tlf. 31 xx xx xx**

6593

**Stud. Polyt.**  
**Torsten Lehmann**  
**Toftekærvej 166**  
**2860 Søborg**  
**Tlf. 31 56 17 34**

6594

**John Corell**  
**Hundshøjvej 15 Ganløse**  
**3660 Stenløse**  
**Tlf. 42 18 48 45**

6595

**Adjunkt**  
**Niels Elbrønd Hansen**  
**Høgevænget 14**  
**2791 Dragør**  
**Tlf. 31 53 63 61**

6596

**Arne Ohrt**  
**Carinaparken 28**  
**3460 Birkerød**  
**Tlf. 42 81 64 67**

# Klub siden

Abonnement på **Egebladet** (6 blade) + medlemskab i et år: kr. 175,00

**Klubdiskette nr. 1** (Fra blad nr. 1-3, samt **blorytme-programmer, raytracer** og **badge-program**): kr. 20,00

**Klubdiskette nr. 2** (Fra blad nr. 4-7, samt medlemmernes godbider.) kr. 20,00

Tidligere numre af **Egebladet**. nr. 1-7 pr. stk: kr. 12,00, nr. 8: kr. 20,00  
Hel årgang 1988 (1-4): kr. 40,00

Ekstra **Badges** (medlemskab inkluderer et gratis badge): kr. 10,00 pr. stk.  
Bemærk: Ovenstående priser er alle inklusiv porto.

### Særlige disketter:

Medlemsregister, **julekalender** (fra nr. 4), vedtægter. Send frankeret svarkuvert og formateret diskette.

**Programmer på tape** (se nummer 6 eller skriv eventuelt efter en komplet liste). Listen er også at finde på vores **BBS**.  
1 bånd: kr. 20,00, 4 bånd: kr. 50,00, 10 bånd: kr. 100,00, lykkepakke med 5 bånd: kr. 50,00  
Logo fra **Logotron**, komplet med **ROM** og instruktionsbog: kr. 200,00

Bestil idag, programmerne findes kun i et begrænset antal. Opgiv eventuelt alternative ønsker. Ved bestillinger på kr. 50,00 og derover betales der ikke forsendelsesporto. Opgiv navn og adresse samt eventuelt medlemsnummer.

**Martin J. Søndergaard**  
**Årestrupvej 76**  
**7470 Karup J**  
**tlf. 97 10 21 95**

# BBS-siden

## Co-SysOp om BBS'et ...

Under SysOp's fravær er det Co-SysOp'ens opgave at skrive om BBS'et.

Vi har i de forgangne måneder fået en stor brugertilgang, i skrivende stund er der 85 medlemmer af **Quercus BBS**. Af disse er ca. 30 faste og regelmæssige brugere. Tilgangen af brugere tog specielt fat efter at **BBS'et** kunne tilbyde 2400 baud modem til speciel klubpris.

Da der stadig er medlemmer af **Quercus** som ikke bruger den service som **BBS'et** tilbyder, vil jeg lige gennemgå disse.

**Den elektroniske postkasse:** Her kan du skrive om hvadsomhelst, spørge efter råd og vildledning, både hvad angår programmering, spilletips etc. Her vil man i de fleste tilfælde kunne få et eller flere hurtige svar (Bare ikke når SysOp og Co-SysOp går til eksamen!).

Postkassen har **SIG's** (Special Interest Groups) for mange forskellige emner, personlige meddelelser, køb & salg, **Archimedes**, **BBC** etc.

Udover postkassen er der muligheden for at hente programmer.

De programmer der er på **BBS'et** er alle bruger-programmer. Som et godt eksempel på et program udviklet af **Quercus BBS** brugere kan nævnes:

En **ANSI-terminal** (**ANSI:** American National Standard Institute).

**ANSI** er den standard alle de **PC**-orienterede **BBS'er** bruger. Programmet, som har undergået forskellige forandringer, er efterhånden det stærkeste terminal-prog-

ram der eksisterer til **BBC'en** (!).

**Quercus BBS** tilbyder også elektronisk korrespondence skak, hvor man kan udveksle træk så hurtigt som telefonen tillader det. Typisk vil man, hvis man ringer et par gange om ugen for at læse post, også kunne gøre et træk i de(t) parti(er) skak man har kørende.

For lukkede brugergrupper, f.eks. **Archimedes-folket**, har **SysOp** mulighed for at oprette specielle områder reserveret til gruppen. Der er allerede 4 **CUG's** (Closed User Groups).

Det er tydeligvis et "offer you can't refuse". **Quercus'** medlemmer kan lævrigt stadig købe et modem til klubpris.

## Announce:

**Quercus BBS**, proudly presents:

**Discovery 2400C Hayes** kompatibel modem with bells and whistles.

Prisen er i øjeblikket 1551 dkr. inkl. moms, indpakning og sommetider dag til dag ekspedition. Er du blevet interesseret så ring til:

**Co-SysOp Klatt 65 91 13 86**

**Quercus BBS** skal i høj grad gerne bære præg af at være brugernes **BBS**.

Så nu hvor indkøringsfasen med alle dens børnesygdomme (mange tak for jeres store tålmodighed!!!) er overstået, er det på tide at folk kommer med forslag til andre aktiviteter. Mulighederne er faktisk kun begrænset af 28K, **No room**, **SysOp & Co.'s** programmeringsfærdigheder og brugernes fantasi.

I næste nummer af **Egebladet** vil **Thomas Christlansen (SysOp)** give en anmeldelse af **ArcTerm 6.01**.

# Adventuresiden

## Martin Søndergaard bringer her hjælp ...

Eventyret begynder på en strand. Find en lille hytte. Smid det våde tøj og tør dig med håndklædet. Find en telefonboks. Tag den afrevne side og læs den. Gå tilbage til stranden og find en hule. Grav i sandet. Åben koden ved hjælp af et af tallene fra siden. Klæd dig på. Husk at undersøge dine overalls. Tilbage til telefonboksen og ring til et af numrene på seddelen.

Nu går turen til **Witherton**. Find en wire til en ostedskærer. Lav den om til en snare. Fang en kanin ved hjælp af snaren. Find et sted med en havelåge. Smid kaninen og åbn lågen. Find en blomst. Find nu en park. Husk blomsten i knaphullet. Læs avisen fra affaldsspenden. Medens du er i byen skal du udføre et stripnummer.

Begiv dig nu ud på **Forest Road**. Følg vejen så langt du kan komme. Gå så ind i skoven. Retningen er:

### **N-E-NW-W-JUMP.**

Find minearbejderens hytte tag batteriet og lagenet. Ved egetræet. Kravl op i det. Brug lagenet til at komme ned på den anden side af hegnet. Gem dig i truget for at undgå de vrede bier. Pas på batteriet ikke bliver vådt. Nu skal du prøve at få helikopteren til at hvirvle støvet op. Et dæksel kommer til syne, som åbnes med skruenøglen.

På gårdspladsen:

Kravl op af espalieren og ind gennem vinduet. Kryb ind i køkkenelevatoren og kør ned til bunden. Find sikkerhedskontoret og læs kortet. Brug kittel og maske for ikke at blive genkendt. Gem dig inde i alkoven og vent. Tag vægten og stå på den ved næste dør. Åben den. Dræb professoren med snaren og tag de hemmelige planer. Flygt via branddøren og gå op på klipperne, hvor du signalerer til U-båden. Derefter går turen

ned på stranden og gennem minefeltet og ud på molen, hvorfra du springer ned i den ventende motorbåd.

Du er nu klar til at begynde på **Myorem**.

Du står bundet til en pæl - **THINK**.

Derefter find tønden - **PUSH DRUM AND LIFT DRUM**.

Bær den hen til lianen. Kravl op efter grenen. Find dåsen - **TAKE CAN WITH TWIG**. Gå til posen og gør som følger: **PUT CAN INTO HOLE - TAKE BAG - TAKE LID - WEAR BAG ON HAND - TAKE CAN - PUT LID INTO HOLE**.

Kravl op af lianen igen. **CUT CREEPER WITH CAN - SWING WEST - SWING WEST - DOWN**.

Gå til tønden. **TIE CREEPER TO DRUM - TAKE LID - PUT LID INTO DRUM - HOLD DRUM**.

Gå hen til afvandingskanalen og **JUMP**. Nøglen til den lille kiste hænger på en krog bag døren. Planken fjernes ved brug af hammeren. Smid madrassen ved foden af klippen. Kryds kviksandet ved hjælp af planken. Husk at fylde krukken med rent vand. Lav en slynge af bandagen og grenen, den fra sumpen.

**MAKE CATAPULT. FIRE STONE WITH CATAPULT** - kan jage pumaen væk. For at komme ud fra hulen skal du springe ned i det mørke hul.

Kortet kan læses ved hjælp af krukken. **READ MAP WITH JAR**. Bær kun på et objekt ad gangen når du krydser hængebroen. Gem dig under kurven når du tager den store sten. Den gamle dør kan smadres ved at rulle den store sten ned mod den. Der er en hemmelig gang bag ildstedet.

En powermatic er en batterioplader. Brug ildslukkeren til at holde væggene i den hemmelige gang. Gem dig i Bilens bagagerum og vent. Har du problemer med et eventyr skal du ikke holde dig tilbage. Bare spørg alt hvad du lyster. Angående eventyr til **Archimedes**. Hvor langt er de forskellige nået. Næste gang vil jeg se på nogle af de gamle eventyr fra **Acornsoft**. Skriv, ring eller læg en besked til mig på vores **BBS**.

# Fraktaler (fort.)

(fraktaler fortsat fra side 23)

Hvorfor Z beregnes på denne måde, skal jeg ikke komme ind på her, det ville tage for lang tid. Til slut, i linie 200 og 210 sættes punktet på skærmen i en farve der på passende vis afhænger af n.

```
10 MODE 1
20 size_x%=320:size_y%=256
30 iterationer%=100
40 reel_min=-2.3:reel_max=0.8
50 imag_min=-1.4:imag_max=1.4
60 dx=(reel_max-reel_min)/size_x%
70 dy=(imag_max-imag_min)/size_y%
80 FOR x=0 TO size_x%
90 FOR y=0 TO size_y%
100 P=reel_min+dx*x
110 Q=imag_min+dy*y
120 X0=0:Y0=0
130 n=0
140 REPEAT
150 X2=X0*X0:Y2=Y0*Y0
160 TEMP=X2-Y2+P
170 Y0=2*X0*Y0+Q:X0=TEMP
180 n=n+1
190 UNTIL n=iterationer% OR (X2+Y2)>10
200 IF n=iterationer% THEN col%=0 ELSE col%=n
MOD 4
210 GCOL0,col%:PLOT 69,x*1280/size_x%,y*1024/
size_y%
220 NEXT x
230 NEXT y
```

Jeg er i gang med at lave et (efter egen mening) ret godt fraktal- tegnings-program til Archimedes, med mulighed for zoom, komprimering af billeder samt meget mere. Interesserede er velkomne til at kontakte mig for at få en kopi. Specielt ville jeg være glad for en henvendelse fra en med en Basic-compiler, eller fra en der har lyst til at lave linierne 80-230 i ovenstående program, i assembler. Fraktaltegning er utroligt tidskrævende, selv på en Archimedes.

Der er flere listige metoder hvorpå man kan få hurtigere Mandelbrot-billeder. Den bedste er: Betragt Mandelbrot mængden som den farve der skal fyldes til; med en flood-fill. Mens der fyldes er farven der fyldes med afh. af iterations dybden. Iøvrigt er 100 iterationer ikke nok, hvis man går tæt på randen. Mvh. Klatt

# ENDPROC

Nu hvor du forhåbentligt har læst hele bladet og syntes om det:

Har redaktionen lige et par bemærkninger at gøre ...

Ved redaktions slutning, det vil sige ved udskrivningen af herværende blad. Har vi observeret flere kedelige fejl. Fejlene er af både typografisk såvel som af orthografisk art. Vi beder mange gange om forladelse. I samme åndedrag lover vi at, næste nummer ikke skulle plages af disse fejl.

Iøvrigt om næste nummer:

Som skulle være på gaden omkring den første September, vil der være :

- Redaktørens tankedryp ...
- Fortsættelsen af "DisARMing" ...
- Yderligere afdækning af mysteriet C ...
- RISC OS applikationer ...
- Anmeldelse af DTP og ArcTerm 6.01
- Læserbreve ...
- Listninger ...
- Stil og dokumentation; en artikel om fremtidige listninger ...
- Og den sædvanlige blandede landhandel af ting og sager ...

Næste nummer af "Egebladet" vil blive udskrevet på laser-printer og skulle derfor gerne blive *meget* flot ...

# Salg og service

RADIO - TV - VIDEO  
DATAUDSTYR  
ELEKTRONISK STYRING  
WALKIE-TALKIE  
TV-OVERVÅGNING

# Breiner Electronic

Bjerregårdsvej 4, Vridsted  
7800 Skive  
Tlf. 97 54 70 15

Rabat til medlemmer af  
QUERCUS

10-15% afhængig af varegruppe

FRAPPORT  
I København

**B**



6556  
Fritidshjemmet Sundby Algaard  
Jens Warmingsvej 77  
2300 København S

**Afsender:**  
**"Egebladet"**  
v/ Kim Latt  
Nørregade 68 B, 2th  
5000 Odense C

Ved vedvarende adresseforandring  
bedes "Egebladet" returneret med  
oplysning om den nye adresse.