



# NYT om KONTAKT - systemet

## kommunernes edb-centrals

### tidsdelingssystem

nr. 2

årgang 1

1969

## SIDEN SIDST

Ved redaktionens slutning den 29.8 havde vi modtaget bestilling på 624 eksemplarer af NOK.

Vi betragter dette som et fint resultat, især da et enkelt eksemplar i de fleste tilfælde dækker en hel skole, og så længe vi ikke modtager kritiske indlæg, vil vi ubeskedent antage, at man finder NOK's form og indhold udmærket. Siden sidst har IBM indledt en regelmæssig kursusvirksomhed i APL-programmering, og hos KEC har der været afholdt et for kursus til udarbejdelse af undervisningsmateriale til en generel undervisning af gymnasie-lærere.

Danmarks Forvaltningshøjskole påbegynder den 1. september et halvårskursus for administrativt personale, som indledes med et datalogikursus baseret på APL-terminaler.

Fra USA har vi erfaret, at man regner med, at der oprettes én ny APL-service om ugen.

I næste nummer af NOK håber vi, at vi kan bringe en rapport over, hvordan man bruger APL hos danske brugere.

## KONKURRENCE

I lighed med så mange andre blade vil NOK i de følgende numre udskrive konkurrencer og præmiere de efter redaktionens opfattelse bedste besvarelser.

Vi har tænkt os at indføre en 1. præmie på 1 times gratis terminaltid på en APL-terminal og 2 2. præmier på  $\frac{1}{2}$  times gratis terminaltid.

Normalt består konkurrencer i, at deltagerne skal besvare et af bladet formuleret problem. Vi har tænkt os at præmiere den bedste problemformulering og lade løsningen være underordnet. Vi håber at kunne opsamle en mængde gode ideer om, hvordan en datalogiundervisning skal gribes an og stiller naturligvis alt materiale til fri afbenyttelse.

Vi udskriver altså en konkurrence, hvor det drejer sig om at formulere ét eller andet problem, som er relevant for en datalogiundervisning. Problemet kan udtrykkes som en tekst evt. med diagrammer, APL-programmer eller lignende. Formuleringen må højst fylde én spalte i NOK og må gerne vedlægges en APL-løsning, som vi afprøver på terminalen. Hvis der ikke følger en løsning med, vil vi i redaktionen give et forslag til løsning, som vi naturligvis håber vil afføde protester og bedre løsninger fra læserne.

Bidrag til konkurrencen skal være redaktionen i hænde senest den 15. oktober for at kunne være med i næste nummer af NOK.

# NYHEDER

## Kontakt-systemet i værksat

Fra den 1. september 1969 har abonnenter på KONTAKT-systemet adgang til APL-service alle hverdage fra kl. 9.00 - 16.00. Der er truffet aftale mellem IBM-Datacentrum og KEC, at IBM midlertidigt stiller datamaskiner til rådighed. Afregningen foregår naturligvis til KEC og på vilkår for abonnement på KONTAKT-systemet.

## Iverson i Danmark

Midt i september måned afholdes hos KEC en international konference om EDB-spørgsmål. Dr. Kenneth Iverson, skaberen af APL-sproget, vil deltage som foredragsholder. Dr. Iverson vil derudover holde et specielt foredrag om EDB i undervisningen for en indbudt kreds af skolefolk.

## Kildeskat

Ved indvielsen af KEC's nye hus den 12. september 1969, vil der i forhallen blive opstillet en APL-terminal, som publikum selv får lov at betjene. Der findes blandt andet et program til beregning af den kommende kildeskat. På denne måde ønsker KEC at demonstrere de muligheder, man i kommunerne har for at yde borgerservice med terminalteknikken.

## Kontakt-systemet til Jylland og Fyn

KEC planlægger en demonstrationsrejse her i efteråret til Fyn og Jylland, så skolefolk vest for Storebælt får lejlighed til at se systemet, ligesom deres kolleger øst for Storebælt.

Vi rykker ud med terminal og TV-udstyr, og kan på denne måde holde en demonstration for ca. 100 deltagere pr. gang. Gode forslag til tid og sted modtages gerne.

## STUDIEKREDS I APL

I dagene 4.-8. august samledes en række gymnasielærere og EDB-folk på KEC for i fællesskab i studiekredsform at diskutere EDB i undervisningen. Hovedvægten blev lagt på undersøgelse af APL's anvendelighed ved gymnasieundervisningen.

Deltagerne var:

Stig Bülow,  
Villum statsskole.  
Jørgen L. Jacobsen,  
Gladsaxe gymnasium.  
Willy Håkansson,  
Søborg gymnasium.  
Jørgen Jørgensen,  
Rønne Statsskole.  
Ingrid Stevnsborg,  
Køge Gymnasium.  
Viggo Nielsen,  
Køge Gymnasium.  
Per Gjerløv,  
IBM - NEUCC.  
Erik Mikkelsen, KEC.  
Erik Nielsen, KEC

Der var to terminaler til rådighed for deltagerne, og i ugens løb blev der skabt en eksempelsamling, som det er hensigten at anvende ved en kommende almindelig undervisning af gymnasielærere. Der var enighed om, at man i overskuelig fremtid vil se EDB i gymnasiet, både som selvstændigt fag og som en støtte til de andre fag. Der var også enighed om, at man i den kommende tid må eksperimentere med de forskellige muligheder, man har med hensyn til indhold, sprog, maskinsystemer o.s.v. Man bør nok begynde med studiekredse på frivillig basis - en undervisning flere af deltagerne havde udmærket erfaring med.

# APL - siden

## Data ind og ud af APL-programmer

Når APL-systemet befinder sig i kalkulatorstilstanden, vil aritmetiske udtryk, som skrives ind på terminalen, blive udregnet, og resultatet øjeblikkelig skrevet ud på terminalen igen.

I det følgende vil vi koncentrere os om, hvordan man under programdefinitionen kan få programmer til at forlange data ind fra terminalen, og hvordan man styrer udskrift af resultater.

INDDATA.

Programdefinitionstilstanden kommer man i ved at indlede en linie med V- tegnet.

Eks:

VFUNK

hvis man derpå definerer linie 1 således:

[1] A←J

vil virkningen være, at når man kalder programmet

FUNK

vil terminalen udskrive

J:

og afventer at vi indtaster en værdi for A. F.eks.

J:

5

Udtrykket  $\leftarrow J$  kan anvendes til indtastning af numeriske data og gerne i vektorform.

F.eks.

FUNK

J:

5 10 15 20

A vil da være en vektor indeholdende værdierne

5 10 15 20

således at f.eks.

A[3]

15

Man kan specificere flere ind-databestillinger i samme linie, hvis man har lyst.  
F.eks.

VFUNK

[1] A←J+2×B←J+C←J×3

FUNK

J:

2

B:

5

C:

3

og A, B og C har fået værdierne

A

25

Husk aritmetiske udtryk udføres fra venstre mod højre

11

C

6

J-symbolet behøver iøvrigt ikke altid at være ledsaget af en  $\leftarrow$ .

Eks:

[1] 2×J

vil fungere således

FUNK

J:

3

6

altså den indtastede værdi 3 erstatter J- symbolet. Værdien af det aritmetiske udtryk udregnes og resultatet udskrives.

Hvis det er alfabetiske informationer, man ønsker indtastet, ser linie 1 ud som følger:

[1] A←J

Eks:

FUNK

A←J

A udskrevet vil da blive:

A←J

altså en vektor med tre elementer indeholdende bogstaverne A, B og E.

Hvis man havde haft en variabel

Eks.:

```

      ABE
7

```

med værdien 7, da kunne man også indskrive ABE i eksemplet med numerisk information.

Eks.:

```

[1]  A+
      FUNK
[]:   ABE

```

Undersøger vi nu indholdet af A får vi

```

7
A

```

UDDATA.

Skriver man i programmet et aritmetisk udtryk eller en tekststreng uden symbolet  $\leftarrow$ , opfattes linien som en skriveordre.

Eks.:

```

      FUNK
[1]  'RESULTATET ER'
[2]  2+2

```

da vil et kald af FUNK resultere i:

```

      FUNK
RESULTATET ER
4

```

Samme virkning vil  $\leftarrow$  symbolet med en venstrepil  $\leftarrow$  få. F.eks.:

```

[2]  A←A+2+2
      FUNK
RESULTATET ER
4

```

$\leftarrow$  symbolet er specielt bekvemt, når man ønsker et mellemresultat udskrevet.

```

[2]  25+□+7×A
      FUNK
RESULTATET ER
28
53

```

Man kan også blande tekst og beregnede værdier i en linie, når man blot adskiller tingene med ;.

Eks.:

```

[1]  'RESULTATET ER: ' ; 2+2
      FUNK
RESULTATET ER: 4

```

Udskrivning af tabeller foregår nemmest ved at fylde data ind i en matrix og derpå forlange den udskrevet.

Har man f.eks. i vektoren C værdierne (0, 1, 2, ..., 100) og i vektoren R værdierne (0, 0.8, 1.6, ..., 80) og ønsker man at udskrive en omsætningstabel mellem C og R definerer man en 2-dimensional matrix M lig med C og R sat sammen.

$M \leftarrow 101 \times C, R$

For at udskriften kan blive pæn må vi vende matricen:

```

      M
0      0
1      0.8
2      1.6
3      2.4
4      3.2
5      4
6      4.8
7      5.6
8      6.4
9      7.2
10     8
11     8.8
12     9.6
13    10.4
.      .
.      .
.      .
98     78.4
99     79.2
100    80

```

# SKOLE - siden

Et vigtigt formål med datalogi-undervisningen må være at give en orientering om, hvordan datamaskinerne anvendes i samfundet. Vi har i programbiblioteket et pædagogisk lagerstyringsprogram, som belyser de grundlæggende problemer, man har, når man skal holde styr på lageret i en større virksomhed.

## LAGERPROGRAM

For hver vare er man interesseret i:

navn  
stykkpris  
antal på lager  
antal bestilt  
gennemsnitligt forbrug pr. uge  
leveringstid (antal uger)

Vi henter lagerstyringsprogrammet ind i vores arbejdsområde i maskinen.

)LOAD 6020 LAGER

Kalder vi funktionen STATUS får vi lagerbeholdningen.

STATUS  
2604 DIMSER  
120 DINGENOTER

Vi kan specificere forbrug af en vare med funktionen:

FORBRUG  
SKRIV VARENS NAVN  
DIMSER  
SKRIV FORBRUG I STK.  
J:  
746  
RESTLAGER ER 1858

Vi kan rapportere at varer går ind på lageret med funktionen

INDGAAET  
SKRIV VARENS NAVN  
DINGENOTER

SKRIV ANTAL INDGAAET  
J:

137  
LAGERANTAL ER HEREFTER 257  
RESTORDRE PAA 363

STATUS  
1858 DIMSER  
257 DINGENOTER

Vi kan finde værdien af lagerbeholdningen med:

TOTALPRIS  
5932.84 KRONER

og oprette nye varer med

OPRETLAGER  
SKRIV VARENAVN  
JINSTREGINGER  
SKRIV PRIS  
J:  
2.78  
SKRIV ANTAL PAA LAGER  
J:  
1004  
SKRIV ANTAL I ORDRE  
J:  
230  
SKRIV ANSLAAET UGETORBRUG  
J:  
200  
SKRIV ANSLAAET LEVERINGSTID  
J:  
3  
SKRIV VARENAVN  
STATUS  
1858 DIMSER  
257 DINGENOTER  
1004 JINSTREGINGER

Varer tages ud af lagerbeholdningen med:

DROPPVARJ  
SKRIV VARENS NAVN  
DIMSER  
STATUS  
257 DINGENOTER  
1004 JINSTREGINGER

I en datalogitime kan man hente de enkelte funktioner frem, dissekere dem for at se hvordan de virker og evt. lade eleverne modificere dem.

# SKOLEEKSEMPEL

Ligningen:

$$x = 1 + \sin x$$

kan ikke løses med de konventionelle metoder man anvender i gymnasiet. Eksemplet skal vise, hvordan man simpelt får en numerisk løsning ved hjælp af sin APL-terminal.

Funktionen skrives i APL som:

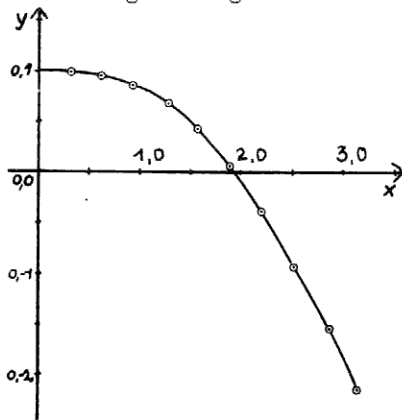
$$F \leftarrow (-X) + 1 + 10X$$

| X      |        |       |       |      |  |
|--------|--------|-------|-------|------|--|
| 0.314  | 0.628  | 0.924 | 1.26  | 1.57 |  |
| 1.88   | 2.2    | 2.51  | 2.83  | 3.14 |  |
| F      |        |       |       |      |  |
| 0.991  | 0.96   | 0.874 | 0.692 | 0.43 |  |
| 0.0726 | -0.392 | -0.92 | -1.52 |      |  |
| -2.14  |        |       |       |      |  |

Funktionsværdierne i intervalendepunkterne kan let kontrolleres ved en manuel beregning.

Man kan inddele intervallet mellem  $x=0$  og  $x=\pi$  i stadig flere delintervaller og med funktionsudtrykket finde en bedre og bedre tilnærmelse for  $x$ -værdien i nulpunktet. Det vil også være værdifuldt at vise en typisk algoritme for udregning af roden.

En meget almindelig fremgangsmåde vil være først at bestemme funktionsværdien i midten af intervallet  $(0;\pi)$ . Herefter beregner man funktionsværdien i midten af det delinterval, hvor funktionsværdierne har forskelligt fortegn.



Man fortsætter med denne intervalinddeling med efterfølgende beregning af funktionsværdien i midtpunktet, et på forhånd fastsat antal gange og derpå udskrives 0-punktet som middelværdien mellem de to sidst beregnede intervalendepunkter. Funktionsværdien i dette punkt findes i F.

Med 10 intervalinddelinger fås:

NULP 10

1.93

F

-0.00179

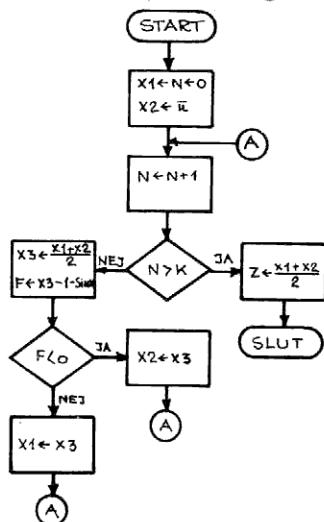
Algoritmen ser sådan ud:

```

X1 ← 0
X2 ← π
N ← 1
DO NULP [N]
  Z ← (X1 + X2) / 2
  F ← (-Z) + 1 + 10Z
  IF F < 0 THEN
    X1 ← Z
  ELSE
    X2 ← Z
  N ← N + 1

```

Det svarer til blokdiagrammet:



Man konstaterer overensstemmelsen med de tabellerede værdier af funktionen.

## netværksplanlægning

I sidste nummer af NOK omtalte vi nogle af de programmer, der findes i APL-systemets programbibliotek, såsom programmer for netværksplanlægning, almindelig statistik, plangeometri, lineær programmering m.v. I dette nummer vil vi gennemgå netværksplanlægningsprogrammet.

Grundlaget for netværksmetodens anvendelse er en grafisk afbildning af et projekt i form af et netværk, der viser enkeltoperationernes eller aktiviteternes indbyrdes sammenhæng. Når projektets struktur på denne måde er fastlagt, foretager man en tidsvurdering af aktiviteterne.

Disse data kan man lade en datamaskine regne på og man får udskrevet rapporter, der er et værdifuldt disponeringsgrundlag. På grund af datamaskinens hastighed kan man hurtigt simulere ændringer i arbejdsplanen, hvorved man har mulighed for at vurdere alternative løsninger og let kan få ajourførte arbejdsplaner.

PERT-netværket (se tegning) består af:

1. Aktiviteter, der er projektets enkeltoperationer (f.eks. støbning af fundament).

Et enkelt netværk i APL/PERT må højst indeholde 200 aktiviteter, men man kan opbygge større netværk ved sammenkædning af flere "subnetværk" på højst 200 aktiviteter.

2. Knudepunkter, der symboliserer begyndelse og afslutning af aktiviteter.

En aktivitet, der udgår fra et knudepunkt, kan ikke starte, før alle aktiviteter, der fører til knudepunktet, er afsluttede. Alle punkter nummereres.

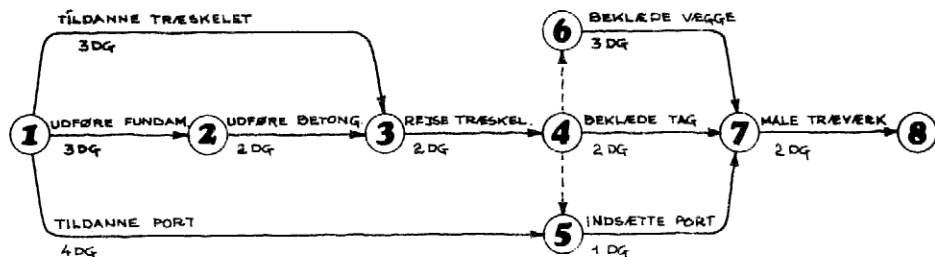
Blandt knudepunkterne skal fremhæves startknudepunktet, hvortil man knytter en dato for projektets start, og slutpunktet, der markerer hele projektets afslutning. Der må kun forekomme ét start- og slutknudepunkt. Man kan give knudepunktet en planlagt dato og ved beregning få at vide, om det ønskede kan lade sig gøre.

Alle data skrives ind over terminalen, og på dette grundlag får vi udregnet den tidligste slutdato for projektet, samt netværkets totale varighed.

For hvert knudepunkt beregnes to "tidspunkter", nemlig det tidligste tidspunkt og det seneste tidspunkt, knudepunktet kan passeres på. For en række knudepunkter vil de to tidspunkter være identiske.

Gennem disse punkter går en følge af aktiviteter, der udgør den kritiske vej gennem netværket, idet der ikke kan forekomme forsinkelser på den, uden at det vil få konsekvenser for den endelige færdiggørelse af projektet. For alle andre aktiviteter gælder, at der kun tillades en vis forsinkelse, uden at det medfører forsinkelse af hele projektet. Den tilladte forsinkelse af hver aktivitet kaldes slæk og udregnes af maskinen. Aktiviteterne på den kritiske vej har altså slæk = 0.

Når man har udført sin beregning af netværket, er det muligt over terminalen at få udskrevet rapporter og stavdiagrammer for det beregnede netværk.



Pildiagram

LOAD MINIPMS  
 SAVED 9.46.06 02/06/69

Programmet kaldes

```

INITIALIZE
**ARBEJDSDAGE I EN UGE?
[]:
  5
**DAGE PR. TIDSENHED?
[]:
  1
**UDSKRIFT AF KALENDER?
NEJ
**RETTELSE TIL KALENDEREN?
NEJ
  
```

Input:  
 Specifikation af netværket.

```

APLPERT
**BESKRIV AKTIVITETERNE VED START, SLUT, VARIGHED, BESKRIVELSE,
PLANLAGT DATO OG AKTUEL DATO
**VED SLUT SKRIV END
**BESKRIV AKTIVITET
1,2,3,UDFOERE FUNDAMENT
**BESKRIV AKTIVITET
7,8,2,MALE TRÆVAERK
**BESKRIV AKTIVITET
,1,0,START,,06/26/69
**BESKRIV AKTIVITET
8,,0,SLUT
**BESKRIV AKTIVITET
END
  
```

```

CALCULATE
**UNIQUE**
**TOP**
**TE**
**TL**
**STARTEN PAA NETVAERKET ER: 08/21/69
**SENESTE SLUT PAA NETVAERKET ER: 09/08/69
**NETVAERKETS TOTALE VARIGHED I UGER ER: 2.6
  
```

Beregning af netværket



**APL - kursus****2. LEKTION**

OPKALD.

Når man ønsker at etablere forbindelse med datamaskinen, drejer man datamaskinens telefonnummer på et ganske almindeligt telefonapparat. Vi kommer i kontakt med APL-systemet ved at skrive:

)4001

Højreparentesen ) definerer linien som en "kommando". Vi skal senere beskæftige os indgående med kommandoer, 4001 er et abonnentnummer. APL-systemet undersøger, om der er registreret en abonnent med det pågældende nummer og svarer i bekræftende fald med:

021) 9.13.37 09/05/69 KEC1

A P L \ 3 6 0

021) er en identifikation af den indgang, vi beslaglægger (port nr.). Derefter klokkeslet og dato for opkaldet og navnet på abonnenten. Kl. 13 min. og 37 sek. over 9, den 9. maj 1969. APL\360 indikerer, at systemet er klar til input. F.eks.:

2+2

4

**SIMPLE ARITMETISKE OPERATIONER**

I 1. lektion så vi opbygningen af simple aritmetiske udtryk. Operatorerne + - \* ÷ står for henholdsvis addition, subtraktion, multiplikation, division og potensopløftning. Vi fortsætter eksempel-rækken med: (Bemærk! En linie afsluttes ved at trykke på RETURN-knappen).

3+1

4

0.55-0.81

-0.26

7.2×6

43.2

1÷2

0.5

2\*3

8

3+

SYNTAX ERROR

3+

A

I det sidste eksempel manglede operatoren + noget at arbejde med. Dette registrerer APL-systemet som en fejl i syntaksen og markerer med A, hvor fejlen er fundet. Bemærk! I subtraktionseksemplet har facit et hævet minus-tegn. På denne måde skelnes imellem operatoren - og fortegn-tegn.

5+3

1.666666667

Antallet af cifre i et sådant resultat vil normalt blive 10. Maskinens interne regninger foregår altid med 16 cifres nøjagtighed. Ønskes et andet antal cifre end 10, indtastes kommandoen.

)DIGITS 16

WAS 10

Resultatet vil nu blive udskrevet med indtil 16 cifres nøjagtighed, indtil en ny DIGIT-kommando kommer.

5÷3

1.666666666666667

5÷0

DOMAIN ERROR

5÷0

A

Det konstateres, at 0 som divisor ikke er defineret. Se dog:

0÷0

1

## FEJLRETTELSER

Har man skrevet en linie og opdager en fejl, inden man afslutter linien ved tryk på RETURN-knappen, har man mulighed for at rette fejlen ved anvendelse af BACKSPACE- og A-knappen. Tryk på BACKSPACE-knappen rykker skrivemaskinens kuglehoved til venstre. Et derpå følgende tryk på A-knappen sletter alt, hvad der er til højre for kuglehovedets øjeblikkelige position på linien, samtidig med at der anbringes et v under det første tegn i linien, som er slettet. Derpå kan man fortsætte i linien med den korrekte skrivning.

På papiret kommer fortsættelsen i linien nedenunder, men i maskinen opfattes det som en sammenhængende linie.

$$2 \times 3.14169$$

$$v$$

$$59$$

$$6.28318$$

$$2 \times 1.2003$$

$$v$$

$$2.4$$

I det sidste eksempel er der blot fjernet noget fra en linie uden tilføjelse af nye tegn.

Det er ligegyldigt, i hvilken orden man skriver tegnene i en linie.

Har man f.eks. skrevet:

$$10 \times 5$$

med blanke foran 10, kan man backspace kuglehovedet og tilføje et ciffer foran 10

$$110 \times 5$$

og ved tryk på RETURN-knappen:

$$550$$

## VEKTORER

Skriver vi en række tal adskilt med blanke, får vi dannet en vektor.

$$1 \ 2 \ 3 \ 4$$

$$1 \ 2 \ 3 \ 4$$

Vi kan anvende de sædvanlige operatorer på vektorer:

$$1 \ 2 \ 3 \ 4 + 5 \ 6 \ 7 \ 8$$

$$6 \ 8 \ 10 \ 12$$

$$1 \ 2 \ 3 \times 4 \ 5 \ 6$$

$$4 \ 10 \ 18$$

Vi ser, at resultatet er en vektor med samme antal elementer, som de to vektorer vi går ud fra.

Det n'te element i resultatvektoren fås ved at beregne udtrykket bestående af det n'te element i de to vektorer adskilt af den pågældende operator.

En vektor med ét element er en skalar.

$$1 \ 2 \ 3 \times 2$$

$$1 \ 4 \ 9$$

$$2 \times 1 \ 2 \ 3$$

$$2 \ 4 \ 8$$

Skalaren virker i et udtryk med en vektor som en vektor med det samme antal elementer, alle af samme størrelse som skalar.

$$1 \ 2 \ 3 + 4 \ 5$$

$$\text{LENGTH ERROR}$$

$$1 \ 2 \ 3 + 4 \ 5$$

$$\wedge$$

Denne vektoraddition er ikke tilladt. De to vektorer skal have samme antal elementer.

VARIABLE.

Med sætningen:

$$A + 2$$

har vi defineret en variabel ved navn A med størrelsen 2.

Den variable A kan indgå i aritmetiske udtryk i:

$$A + 2$$

$$4$$

Man kan frit vælge navn til sine variable og ændre indholdet med + sætningen.

NIELS+0

NIELS

0

Sætningen opfattes her, som tidligere, som et aritmetisk udtryk, som maskinen beregner og udskriver på terminalen.

NIELS+NIELS+5

NIELS

5

I dette eksempel beregnes en ny værdi af NIELS ved at addere 5 til den gamle værdi.

Vi kan også definere en variabel som en vektor med + sætningen:

ERIK+1 2 3 4 5

og beregne

NIELS\*ERIK

5 25 125 625 3125

## FUNKTIONER.

Indleder vi en linie med tegnet  $\nabla$  efterfulgt af f.eks. navnet PI, indtræder systemet i en tilstand - programdefinitionstilstanden-, hvor den enkelte linie ikke eksekveres efter vognretur. I stedet defineres en funktion ved navn PI, og efter vognretur efter 1. linie udskrives en ny linie, som indledes med [1], som tegn på, at APL-systemet er klar til at modtage 1. linie af funktionen PI.

 $\nabla$ PI

[1] 3.14159v

I linie 1 har vi skrevet et aritmetisk udtryk, nemlig konstanten 3.14159 og derpå endnu engang  $\nabla$ -tegnet, som nu ophæver programdefinitionstil-

standen, så vi er tilbage i kalkulatortilstanden - vi har lukket funktionen.  
Kalder vi funktionen:

PI

3.14159

får vi udført de definerede linier én ad gangen i linienummerorden, her altså kun linie 1, som indeholder en ordre om udskrivning af konstanten 3.14159.

## fortsat fra side 6

Under programmeringen kan man have megen nytte af tracefunktion:

TANULP+110

Med denne sætning har man angivet, at man ønsker en udskrift for hver linie (1,2,3,...,10) af funktionen NULP, når denne kaldes næste gang.

Vi prøver med 3 intervaldelinger.

NULP 3

NULP[1] 1  
NULP[2]  
NULP[3] 1.57  
NULP[4] 0.429  
NULP[5]  
NULP[6] 1.57  
NULP[7] 1  
NULP[1] 2  
NULP[2]  
NULP[3] 2.36  
NULP[4] 0.649  
NULP[5] 8  
NULP[6] 2.36  
NULP[9] 1  
NULP[1] 3  
NULP[2]  
NULP[3] 1.96  
NULP[4] 0.0396  
NULP[5] 8  
NULP[6] 1.96  
NULP[9] 1  
NULP[1] 4  
NULP[2] 10  
NULP[10] 1.77  
1.77

KEC



I/S KOMMUNERNES EDB-CENTRAL

BRØNDBYØSTER BOULEVARD 22 BRØNDBYØSTER.

2650 HVIDOVRE TLF: (01) 75 45 11