

PRISMA

Computerclub Deutschland e.V. • Postfach 11 04 11 • Schwalbacher Straße 50 • D-6000 Frankfurt am Main 1

November / Dezember 1989 Nr. 6

D 2856 F

```
void main(int argc, char **argv)
{ FILE *Stream1;
  char *Filename, far *ScreenAdr;
  int PageLength, DelayTime;

  printf("\nSHOW");
  printf("A.Wo");
  ScreenAdr = (far *) 0xB0000000;
  0xB8000000;
  if (!Read(argc, argv, &Filename, &PageLength, &DelayTime))
  { Print(1); }
  if (!OpenFile(Filename, &Stream1))
  { exit(2); }
  Read(ScreenAdr, OldScreen);
  while (Read(Stream1, Buffer, PageLength))
  { Write(ScreenAdr, Buffer, ScreenAdr);
    DelayTime; }
  Write(OldScreen, ScreenAdr);
  if (fclose(Stream1))
  { exit(3); exit(3); }

int ParseLine(int argc, char **argv, char **fn, int *pl, int *dt)
{
  if (argc < 2 || (argc > 4)) return(FALSE);
  *pl = atoi(argv[1]);
  *dt = atoi(argv[2]);
  if (*pl < 0 || *dt < 0) return(TRUE);
  if (*pl > 2; argc-)
  { *pl++; }
  if (*pl[0] == '/') /* p2[0] entspricht *p2 */
  { *pl[1] = *pl[1]; /* p2[1] entspricht *(p2+1) */
  }
  else if (*pl[0] == 'L')
  { *pl[1] = 'l';
    *pl = 4096;
    break;
  }
  else if (*pl[0] == 'D')
  { *pl[1] = 'd';
    *pl = atoi(p1+2, &p2, 10);
    if (*pl > 32000) || (*pl < 0))
    return(FALSE);
  }
  else
  { *pl[1] = 'y';
    return(TRUE); }
}

int OpenFile(char *filename, FILE **stream)
{ printf("Eröffne Datei %s\n", filename);
  if ((*stream = fopen(filename, "rb")) == NULL)
  return(FALSE);
  return(TRUE); }
```

Die Programmiersprache C ist Gegenstand einer Artikelserie von Dr.-Ing. Frank Rieg, die in diesem PRISMA beginnt.

Magazin

CCD 1990

Grundlagen

C - Programme - ein Überblick

Taschenrechner

28-Mini-Pack

Serie 70

CCDUTIL

Key-Files

Frequenzen im HP-71B

BASICLEX

Serie 40

Analytische Geometrie

Berechnung von
Himmelstrabanten

Grenzwertberechnungen

GEW + POP

Editorerweiterung für den
HP-41 CX

Suche deutsche Bedienungsanleitung oder Übersetzung für:

HP-75 I/O ROM (# HP 00075-90243)

HP-75 Text Formater (# 00075-15019)

HP ThinkJet Printer (# 02225-90032)

Wilfried Kern, Niederurseler Landstr. 153, D-6000 Frankfurt 50, Tel. (069) 57 55 72

Verkaufe: HP-41 Translator Pack (mit FORTH) für HP-71, Internal Design Specification IDS-I für HP-71,

Suche: PC TeX Metafont; Matthias Rabe, Tel. (0521) 32 44 74

Verkaufe: Mettenmeyer Interface HP-IL <-> RS232C/Centronics, netzunabhängig mit Lithiumakkus, Ladegerät und Lederhülle, 950,- DM; Grabau GR7 Grafik Video Controller (64 KB RAM für 4 Bildschirmseiten) zusammen mit 9" Monitor 1250,- DM;

Alf-Norman Tietze, Tel. (069) 789 3995

Suche für HP 41 die Programmsammlung / Solution Book: Cardiac/Pulmonary, falls nicht verkäuflich evtl. leihweise oder als Kopie.

Rainer Gillmann, Hamburg
Tel.: 040-726 12 87. Erreichbar ab ca. 19.30h, rufe zurück.

Verkaufe HP 71B + IL-Modul + ThinkJet Drucker + Digital Tape + HP Video, mit allen Handbüchern + 3 Jahre PRISMA usw., komplett DM 2200,-

Tel.: 0208/84 51 00 ab 17 Uhr.

Suche für HP 41 CX
Navigationsmodul, Magnetkartenleser und Lesestift.

Ernst Rauscher, 1Chemin des Tuilleries, F31770 Colomiers, Tel.: 61.78 80 18 privat / 61.935186/7

Suche HP 71 Speichermodule und HP 71 Translator-Modul, sowie einen nicht programmierbaren UPN-Rechner mit Winkelfunktionen.

Wer kennt Möglichkeiten, um mit dem HP 41 Temperatur, Luftfeuchte, Luftdruck und Benetzungsdauer einer bestimmten Fläche zu messen? Anhand der gemessenen Daten soll mit einem entsprechenden Programm die Pilzinfektionsgefahr und der Schädlingsbefall bei Obstbäumen vorausbestimmt werden.

Besteht Interesse an Programmen zur Berechnung von Dachstühlen und deren Holzlisten?

Bernhard Leiter, Mühlenweg 2, 7996 Liebenau

Verkaufe: HP DeskJet Printer mit 128 KB RAM, wenig gebraucht, für 1450,- DM, auf Wunsch mit Softfonts;

Thomas Kern, Tel. (02689) 67250

Suche HP-IL-RS232 Interface und Grabau GR7 Interface.

Verkaufe HP 75 mit 8k Extension für DM 600,-.
M.L. Hennessy, Feldbergblick 2, 6000 Frankfurt 56, Tel.: (06101) 48083

Verkaufe: Digitalkassettenlaufwerk HP 82161A mit Netzteil und 15 Digitalkassetten (mit Software für HP-71), 500 DM.

Dennis Foeh, Hermann-Hanker-Straße 17, 3400 Göttingen, Tel.: (0551) 792857

JPC-News

Heft Juli-September 89
Nummer 67

Die JPC-Redaktion bedauert, daß auch diese Nummer reichlich dünn ausgefallen ist, da immer noch zu wenig Beiträge eingehen. Für den HP-71 erstmalig überhaupt keine. Um wieder mehr Motivation zu wecken, ist beschlossen worden, Beiträge über ein Punktesystem anzulocken.

Erinnert wird an die immer noch laufende Fragebogenaktion.

Berichtet wird von einer Tagung am 3.6.89 in Chicago aus Anlaß zweier Jubiläen mit Beiträgen u.a. von Michael Markov und Richard Nelson:

- 10 Jahre HP-41 und
- 50 Jahre Hewlett Packard.

Erwähnt wird als neues Produkt:

- HP-28COM, ein HP28S mit Interface RS-232 der dänischen Firma WM Electronics.

Programm-Beiträge:

HP-28

- Neue Funktionen
- Virtueller Bildschirm
- Lösung von Differentialgleichungen erster Ordnung nach Runge-Kutta
- Orgelprogramm
- Histogrammgenerator
- Verschlüsselung von Zeichenketten
- Schnelle Berechnung von Fakultäten (z.B. 100!)
- "SWITCH"-Funktion

HP-41

- Packing der "assign"-Register

Georg Urbanski

Die NASE VORN mit TRANSFER

dem Übertragungssystem für die HP-Taschencomputer 17B, 19B, 27S, 28C/S, 41 C/V mit IR-Modul und 42S.

Mittels einem INFRAROT-Interface können Programme, Daten oder Grafiken an **IBM PC/XT/AT** kompatible Rechner oder **ATARI-ST** Computer übertragen werden (DM 280,-).

Dies und ähnliche Übertragungssysteme, wie zum Beispiel für die **SHARP** Taschenrechner oder den **CASIO FX 850P** sind erhältlich bei:

Packheiser Hardware Products
Rechtstraat 39
B-3770 Valmeir (Belgien)
Tel.: 0032 12 455617

oder

Hangweg 4
D-5800 Hagen 8

Rabatt für CCD Mitglieder!
Händleranfragen erwünscht.

Impressum

Titel:
PRISMA

Herausgeber:
CCD - Computerclub Deutschland e.V.

Postfach 11 04 11
Schwalbacher Straße 50
6000 Frankfurt 1

Verantwortlicher Redakteur:
Alf-Norman Tietze (ant)

Redaktion:
Klaus Kaiser (kk)
Michael Krockner (mik)
Martin Meyer (mm)
Dieter Wolf (dw)

Herstellung:
CCD e.V.

Manuskripte:
Manuskripte werden gerne von der Redaktion angenommen. Honorare werden in der Regel nicht gezahlt. Die Zustimmung des Verfassers zum Abdruck wird vorausgesetzt. Für alle Veröffentlichungen wird weder durch den Verein noch durch seine Mitglieder eine irgendwie geartete Garantie übernommen.

Druck und Weiterverarbeitung:
Reha Werkstatt Rödelheim
Biedenkopfer Weg 40 a, 6000 Frankfurt

Anzeigenpreise:
Es gilt unsere Anzeigenpreisliste 3 vom Juni 1987

Erscheinungsweise:
PRISMA erscheint jeden 2. Monat

Auflage:
2500

Bezug:
PRISMA wird allen Mitgliedern des CCD ohne Anforderung übersandt. Ein Anspruch auf eine Mindestzahl von Ausgaben besteht nicht. Der Bezugspreis ist im Mitgliedsbeitrag enthalten.

Urheberrecht:
Alle Rechte, auch Übersetzung, vorbehalten. Reproduktionen gleich welcher Art - auch auszugsweise - nur mit schriftlicher Zustimmung des CCD. Eine irgendwie geartete Gewährleistung kann nicht übernommen werden.

Inhalt

Magazin

JPC-News	2
CCD 1990	3

Clubnachrichten

UPLE, was war das denn gleich ?	37
Anwendungsbereich Naturwissenschaften und Technische Wissenschaften	48

Grundlagen

C-Programme - ein Überblick	4
-----------------------------	---

Taschenrechner

28-Mini-Pack	38
--------------	----

Serie 70

CCDUTIL	15
Key-Files	16
Frequenzen im HP-71B	16
BASICLEX	17

Serie 40

Analytische Geometrie	21
Berechnung von Himmels- trabanten	27
Grenzwertberechnungen	33
GEW + POP	34
Editorerweiterung für den HP-41 CX	37

Barcodes

POP	36
IP-Daten	39
IP	40
GEO	42
GEW	45
TST	46
EDFL	46
LIM	46

Kurz gemeldet	26+48
---------------	-------

Serviceleistungen	47
-------------------	----

Clubadressen	47
--------------	----

Vorschau

auf die nächsten Hefte

LR-Zerlegung
MCode ASRCH
Tag & Jahr
Einfaches Navigationsprogramm
Geschwindigkeitsmessung
Statik: Zweigelenkrahmen
IP-Ephemeriden
TI 58/59 Emulation
Mutliplot

CCD 1990

Die ständig wachsende Leistungsfähigkeit der Microcomputer und besonders deren Anwendungsprogramme erfordern immer umfangreichere Kenntnisse über Arbeitsweise und Einsatzmöglichkeiten der EDV.

Mit diesem Hintergrund wurde 1981 der CCD e.V. als unabhängiger Anwenderclub gegründet. Der "Dreh- und Angelpunkt" der ca. 500 Mitglieder aus der Gründerzeit war der HP-41C Taschencomputer von Hewlett-Packard. Tips und Tricks zur Programmierung, technische Informationen, Testberichte und vieles andere mehr rund um dieses handliche Computersystem sind noch bis heute ein Schwerpunkt innerhalb des Clubs. Die Mitglieder sind bundesweit "verteilt" und sind vereinzelt sogar aus dem Ausland. Aufgrund dieser räumlichen Mitgliederverteilung wird der Erfahrungsaustausch über unsere Clubzeitschrift PRISMA unterhalten.

Heute zählen wir ungefähr 2000 Mitglieder und es sind zahlreiche Themen und Anwendungsgebiete hinzugekommen: Hard- und Software für HP Taschencomputer der Serien 40 und 70 sowie für die neue Generation der HP Taschenrechner, für Personal-Computer unter MS-DOS und für ATARI-ST-Computer, Programmbibliotheken mit Diskettenservice, Literaturhinweise und Buchbesprechungen, eine CCD-Mailbox im GeoNet-System und vieles andere mehr.

Besonders die Vielfalt sowohl der Programmiersprachen als auch der unterschiedlichsten Softwarepakete haben einen deutlichen Bedarf an weiterführenden und umfassenderen Informationen mit sich gebracht. Die Erfahrungen von Mitgliedern im Umgang mit der EDV und Möglichkeiten zur Lösung spezifischer Hard- und Softwareprobleme werden in unserer Clubzeitschrift PRISMA veröffentlicht. Daraus ergeben sich auch häufig persönliche Kontakte unter den Mitgliedern.

In den Ballungsräumen Berlin, Hamburg, Karlsruhe, Köln, München und Rhein-Main finden sich Clubmitglieder zu regelmäßigen Regionaltreffs zusammen. Für die Themenkreise E-Technik, Geowissenschaften, Mathematik, Naturwissenschaften, Vermessungswesen und spezielle Hardware-Technik stehen Ansprechpartner zur Verfügung, die neben den EDV-Kenntnissen auch die entsprechenden Fachkenntnisse zum Anwendungsgebiet haben. Die einmalige Leistung des CCD und von PRISMA ist das Potential zum Erfahrungsaustausch unter unseren Mitgliedern.

Diese Leistung ist nur durch die aktive Mitwirkung unserer Mitglieder möglich gewesen und wird auch weiterhin nur dadurch möglich bleiben - vielen Dank an alle "Aktiven" im Club.

Auch im Jahr 1990 geht es weiter mit dem notwendigen Erfahrungsaustausch und die PRISMA-Redaktion wird stets bemüht sein, den größten Teil der zu übermittelnden Erfahrungen ansprechend in unserer Clubzeitschrift zu "verpacken". Wir beginnen ein neues Jahrzehnt, und der EDV wird auch weiterhin eine bedeutende Rolle sowohl im Berufsalltag als auch in der Freizeit zukommen. Sogar die Entwicklungen in der DDR stimmen positiv im Hinblick auf einen möglichen deutsch-deutschen Erfahrungsaustausch. Zur Zeit müssen wir uns jedoch noch gedulden, denn die Amerikaner haben ihre Ausfuhrsperrung für "Hochtechnologie" in Staaten des Ostblock noch immer nicht aufgehoben bzw. gelockert.

Lassen wir uns überraschen, was dieses Jahr für uns und unser Metier zu bieten hat. Ein gutes Jahr 1990 wünscht Euch

Alf-Norman Tietze
(Chefredakteur)

C-Programme - ein Überblick

erster Teil 1. Allgemeines

Diese mehrteilige Folge soll einen kleinen Einblick in das Programmieren mit C geben. Es ist günstig, wenn der Leser schon gewisse Erfahrungen mit einer Hochsprache sammeln konnte, aber es sollen auch unsere Clubfreunde, die bisher mit dem HP 41 arbeiten, angesprochen werden. Der Autor selbst programmiert hauptsächlich technisch-wissenschaftliche Probleme und Systemdienstprogramme mit FORTRAN 77, C und Assembler, für Messwerterfassung auch in HP-BASIC.

Die Programmiersprache C ist in den letzten Jahren neben die etablierten Hochsprachen BASIC, Pascal, FORTRAN und COBOL getreten. In den Computerzeitschriften ist ein regelrechter C-Boom zu verzeichnen. Diese Sprache wird oftmals als das Nonplusultra der Programmiersprachen von verschiedenen Autoren dargestellt, die alle bisherigen Hochsprachen überflüssig machen und sogar die Assemblerprogrammierung ersetzen soll.

Zu diesen Aussagen und Trends soll folgendes angemerkt werden: Es gab bislang keine Universal-Programmiersprache, es gibt sie zur Zeit nicht und aller Wahrscheinlichkeit wird es sie auch nie geben. Jeder, der mehrere Programmiersprachen beherrscht (und damit ist gemeint, daß er ständig mit mehreren Sprachen an ernsthaften Aufgaben arbeitet und nicht nur einige schlaue Bücher gelesen hat), wird bestätigen:

“Für jede Programmieraufgabe gibt es eine optimal geeignete Sprache“.

Um eine ganz grobe Einteilung zu geben, die sich auf Beobachtungen des Autors stützt (und daher nur Tendenzen zeigen kann), folgende Erläuterungen:

BASIC ist als erste Hochsprache ansich nach wie vor emp-

fehlenswert, da sehr leicht erlernbar. Aber auch eine Reihe von kommerziellen Programmen für PCs sind in BASIC geschrieben. Eine Domäne von BASIC ist die Messwerterfassung in technisch-wissenschaftlichen Labors. Das hängt damit zusammen, daß derartige Erfassungs- und Steuerungsprogramme oft schnell erstellt werden müssen, oft nur für eine Aufgabe oder Messserie verwendet werden und dann umgearbeitet werden. Hier wird in der Industrie sehr oft HP-BASIC verwendet. Ein schwerer Nachteil von BASIC: Praktisch ungenormt, zahlreiche Dialekte, die untereinander nicht kompatibel sind.

Pascal wurde von Prof. Wirth als Lehr- und Lernsprache konzipiert, die besonders Informatik-Studenten von Anfang an dazu bringen soll, sauber und strukturiert zu programmieren. Diesem Anspruch wird Pascal von allen Hochsprachen am besten gerecht. Als "Einstiegs"-Hochsprache rein theoretisch daher absolut richtig, aber doch schwieriger (weil strengere Regeln) als BASIC. In der Industrie für kaufmännische und technisch-wissenschaftliche Anwendungen auf größeren Maschinen ist es kaum verbreitet, im Gegensatz dazu aber auf PCs als Turbo-Pascal sehr weit. Pascal wird stark zu Ausbildungszwecken an Schulen und Universitäten eingesetzt.

FORTRAN ist die älteste Hochsprache überhaupt und die heute gängige Version ist FORTRAN 77. BASIC wurde in den 60er Jahren aus FORTRAN entwickelt. Nach wie vor die Standard-Sprache für den technisch-wissenschaftlichen Einsatz:

FORTRAN weist eine Unmenge an eingebauten mathematischen Funktionen auf und es gibt jede Menge von fertigen Unterprogramm-Bibliotheken für alle denkbaren mathematischen Problemen. Fast alle

CAD- und Finite-Elemente-Programme auf größeren Maschinen sind FORTRAN-Programme und die sog. Supercomputer werden zum großen Teil unter FORTRAN gefahren.

COBOL ist ebenfalls sehr traditionell und ist die Standardsprache für kommerzielle Anwendungen auf mittleren und großen Anlagen. In diesem Bereich ebenfalls verbreitet ist RPG II, eine IBM-Sprache. Rechnen im technisch-wissenschaftlichen Sinne ist mit diesen Sprachen praktisch unmöglich, sie sind für Massendatenverarbeitung bei vergleichsweise simplen Rechenoperationen ausgelegt. Ansich ist diese Sprache von sehr großer Bedeutung, da der kommerzielle Sektor der Datenverarbeitung in allen Industriefirmen und Verwaltungen anteilmäßig der größte ist. Auf PCs ist diese Sprache kaum verbreitet.

Assembler hat systemimmanent den schweren Nachteil, daß die Sprache von CPU zu CPU teilweise sehr stark unterschiedlich ist. Assembler ist unverzichtbar, wenn höchste Geschwindigkeiten und kompakteste Programme gefragt sind. Kein Compiler kann so effizienten Code erzeugen wie der Mensch selbst (das kann man leicht nachprüfen, wenn man sich den generierten Code eines C- oder FORTRAN-Compilers näher anschaut). Daher stark eingesetzt in der Systemprogrammierung und für Grafikroutinen. Der Anteil ist durch zunehmende C-Verbreitung sicher etwas rückläufig, aber bis heute nicht zu ersetzen. Assembler-Unterrou-tinen sind sehr dankbar, um kleine Teilprobleme (auf DOS-, BIOS- oder Hardware-Ebene) innerhalb eines Hochsprachen-Programms zu lösen. Nichts für Anfänger.

Kleine Anmerkung: Warum sind BASIC, FORTRAN und COBOL großgeschrieben? Weil

es im Gegensatz zu Pascal und Assembler Kunstworte sind.

Wie ist in diesem Szenario die Sprache C einzuordnen? Dazu betrachtet man am besten die Historie: Die allerersten Versionen des Betriebssystems UNIX waren noch in DEC-PDP 7 Assembler geschrieben. Es ist der Verdienst von Dennis Ritchie, mit der Sprache C in den 70er Jahren einen "portablen Assembler" (nach [1]) geschaffen zu haben. Das UNIX-System wurde dann praktisch zum größten Teil in C umformuliert und nach [2] ist es so, daß der UNIX-Systemkern ca. 13.000 Zeilen C-Code und nur noch ca. 800 Zeilen Assemblercode auf niedrigster Ebene umfaßt.

C ist also untrennbar mit UNIX verbunden und es ist heute "die" Systemprogrammiersprache. Auch sehr große Teile des neuen PC-Betriebssystems OS/2 sollen in C geschrieben sein, wohingehend DOS noch weitestgehend in 8086-Assembler verfaßt ist. Auch das TOS/GEM des ATARI ST ist in C geschrieben. C hat zunehmende Bedeutung für Einplatinen-Computer- und Steuerungen, da es praktisch für jeden Mikroprozessor und Mikrocontroller (Mikroprozessor mit eingebautem RAM, serieller Schnittstelle etc.) heute C-Compiler gibt. C ist also eine Systemprogrammiersprache und liegt als solche hinsichtlich ihrer Komplexität zwischen Assembler und den eigentlichen Hochsprachen. Es als echte Hochsprache zu bezeichnen ist zwar heute üblich, trifft aber nicht den Kern der Sache ("portabler Assembler").

Ich wage sogar zu behaupten: Erst wenn man Assembler-Kenntnisse hat, kann man gewisse Konzepte in C richtig verstehen und nutzen. Ein kleines Beispiel möge zeigen, daß hier tatsächlich auf etwas niedrigerer Ebene operiert wird: Es soll eine ganze Zahl einge-

lesen werden, mit 3 multipliziert werden und wieder ausgegeben werden (Listing 1).

Wie man leicht sieht, ist das BASIC-Programm am einfachsten. Es brauchen nicht einmal I und K deklariert zu werden. Eingabeaufforderung und Lesen werden in einem Schritt erledigt. Formatangaben sind unnötig. Das FORTRAN-Programm hat schon mehr Schritte. I und K sind als INTEGER zu deklarieren (wenn man es nicht explizit tut, nimmt FORTRAN automatisch an, daß alle mit I, J, K, L, M, N beginnenden Variablen vom Typ INTEGER mit 4 Bytes sind). READ und WRITE sind eingebaute FORTRAN-Funktionen.

Das C-Programm ist äußerlich dem F77-Programm nicht unähnlich. Aber: die Deklaration "int" hat nicht mehr die feste Bedeutung 4 Bytes, sondern reflektiert die Maschinen-Wortlänge. Das sind bei einer 16-Bit-Maschine 16 Bit, bei einer 32-Bit CPU aber dann 32 Bit. Als nächstes, und das ist zunächst verblüffend: die Funktionen printf und scanf sind keine durch Norm definierten und in C eingebauten Funktionen! Es gibt sie gar nicht im C-Sprachkern. Sie werden vielmehr durch Bibliotheken bereitgestellt, die von Compiler zu Compiler mehr oder weniger stark differieren! Um sie näher zu deklarieren, muß das Header-File stdio.h eingefügt werden (was ebenso wie die entsprechenden Bibliotheken jedem C-Compiler beigelegt sind, denn ein Programm ohne I/O ist wenig wert). Die Funktionen printf und scanf sind noch am meisten "genormt", aber andere C-Funktionen weichen selbst bei PC-Compilern untereinander mehr oder weniger ab. Am deutlichsten wird der maschinennähere Level von C mit Eingabe von i und Ausgabe von k: Bei scanf steht "&i". Das bedeutet: Lies die Zahl auf die ADRESSE von i ein. Bei printf steht "k". Das bedeutet: Gib den WERT von k aus. Wie ersichtlich, bewegt sich das C-Programm auf relativ niedrigem Level.

Es soll hier in aller Deutlichkeit gesagt werden: Es gibt leicht-

ter zu lernende Programmiersprachen als C.

C ist keine typische Anfängersprache.

2. Hard- und Software

Welche Maschine ist für C-Programmierung geeignet? Prinzipiell jeder Computer, für den es einen C-Compiler gibt. Zu langsam sollte er allerdings nicht sein: Also ab ATARI ST, Amiga oder PC aufwärts.

Welches Betriebssystem ist optimal? Darauf gibt es eine ganz einfache Antwort: Ein virtuelles Betriebssystem, das von sich aus die Speicherverwaltung besorgt.

Dazu gehören OS/2 und UNIX. Gerade bei C ist es so, daß bei bestimmten Konzepten wie Pointeradressierung, dynamischer Speicherallokierung etc. leicht Abstürze vorkommen können. DOS und TOS reagieren darauf meist unfroh, während OS/2 und UNIX derartige Fehler abfangen. Für einen guten C-Start sind daher UNIX und OS/2 ideal. In der Praxis wird es aber so sein, daß die meisten unserer Leser ein DOS- oder TOS-System zur Verfügung haben. Damit läßt sich natürlich genauso gut programmieren, nur solche üblen Fehler können von diesen Betriebssystemen eben systemimmanent nur unzureichend abgefangen werden (damit muß man leben).

Bei den Compilern kann man unterscheiden: die klassischen Command-Line-Compiler und die Integrierten Entwicklungsumgebungen. Die Command-Line-Compiler werden von Hand gestartet, nachdem mit einem Editor die Sourcen erzeugt wurden. Diese Compiler sind meist besonders leistungstark und können mit einer Vielzahl von Compilerswitches auf alle denkbaren Situationen eingestellt werden. Dazu gehören beispielsweise MS-C für DOS und OS/2 und natürlich der klassische "cc" aller UNIX-Systeme (bei jedem UNIX-System ist dieser Compiler Teil des Lieferumfangs). Für einen C-Beginner ergibt sich die etwas unschöne Situation, daß er nicht nur mit C selbst kämpfen muß,

sondern auch gleichzeitig die nicht immer einfache Compilerbedienung lernen muß. Die meisten fortgeschrittenen Anwender dagegen bevorzugen dann solche Command-Line-Compiler, weil man den größtmöglichen Leistungsumfang hat und sehr viel durch Batch- und Make-Files automatisieren kann. Wer sich für einen solchen Compiler entscheidet, muß einen Editor oder ein Textverarbeitungsprogramm zur Verfügung haben. Bei manchen Compilern, wie MS-C, wird ein Editor mitgeliefert und UNIX-Systeme haben ohnehin mehrere Editoren im Lieferumfang enthalten.

Das Gegenstück sind die Integrierten Entwicklungsumgebungen. Bekannte Beispiele: DOS: Turbo-C und Quick-C, ATARI: Laser-C und Prospero-C. Hier findet der Anwender ein meist gewaltiges Menüsystem, das vom Editieren, Compilieren, Linken und ggf. Debuggen alles innerhalb eines Programms bietet. Bei manchen dieser Compiler kann ebenfalls wie oben der Compiler außerhalb der Integrierten Umgebung von der Command-Line aufgerufen werden. Solche Compiler haben für den Beginner mehrere gewichtige Vorteile: Sie übersetzen und linken meist extrem schnell (jedenfalls weitaus schneller als die großkalibrigen Command-Line-Compiler) und springen bei aufgefundenen Fehlern direkt auf die Fehlerstelle im Programm und aktivieren den Editor. Die Bedienung der Compiler ist oft spürbar einfacher als die der Command-Line Typen, wenngleich die heutigen Integrierten Entwicklungsumgebungen derart voll von Pull-Down-Menüs sind, daß dies auch wieder bereits verwirrt. Fast trauert man einfachen und extrem leicht zu erlernenden Umgebungen wie Turbo-Pascal 3.0 nach.

Es ist nun nicht so, daß diese Integrierten Entwicklungsumgebungen ausschließlich nur für Anfänger gedacht sind. Der erzeugte Maschinencode ist nicht zwingend langsamer als der der "großen" Compiler. Mir sind Fälle bekannt, bei denen

XX-C schnelleren Code erzeugte als der "Supercompiler" eines namhaften Herstellers.

Wer noch keinen C-Compiler hat, welchen soll er wählen? Dazu kann man lapidar sagen: Compilerkauf ist Vertrauenssache! Es ist leider so, daß nicht alle Compiler immer fehlerfreien Code erzeugen! Es gibt bestimmte Compilerversionen- bzw. Releases auch von bekannten Herstellern, die t.w. üble Fehler in das fertige Programm bauen (z.B. werden große Vektoren falsch adressiert). Ganz vorsichtig muß man nach meinen Beobachtungen mit ganz neuen Releases sein. Oft wurden hier bekannte Fehler beseitigt, aber dafür neue eingebaut. Hier hilft nur eines: Bei Bekannten, Clubkameraden nachfragen, was sie über Compiler XX wissen. Exotische Hersteller sollte man als Anfänger lieber auslassen, sondern ein Modell wählen, das als einigermaßen gut bekannt ist und das möglichst auch Bekannte haben. Sonst ist man bei Compilerproblemen allein auf weiter Flur; auf die sog. "Hotlines" der Hersteller braucht man sich gar nicht erst zu verlassen.

Weiter ist zu bedenken: Der C-Sprachkern ist sehr klein. Wie erwähnt, wird das ganze I/O über Bibliotheken abgewickelt. Diese Bibliotheken differieren stärker, als man glaubt. Es ist reine Glückssache, wenn bei DOS (vom Übergang auf andere Betriebssystem nicht zu reden) eine Source, die mit Compiler A einwandfrei übersetzt und gelinkt wurde, sofort und ohne Änderung mit Compiler B funktioniert. Völlig unübersichtlich ist die Situation beispielsweise bei Grafikprogrammen. Hier sind die Grafikaufrufe der beiden DOS-Bandleader total unterschiedlich. Es wird in allen möglichen Fachzeitschriften behauptet, daß C-Quellen höchst portabel sind. Dazu ist anzumerken: C-Programme sind genauso gut oder schlecht portabel wie BASIC-, Pascal- oder FORTRAN-Programme.

Es soll nicht der Eindruck erweckt werden, daß hier nur

```
BASIC (HP-75): 10 INPUT 'gib Zahl'; I
                20 K= I * 3
                30 DISP 'Ergebnis:';K
                40 END
```

```
FORTRAN 77:
  INTEGER I,K
  WRITE(*,'('' gib Zahl')')
  READ (*,'(I5)') I
  K= I * 3
  WRITE(*,'('' Ergebnis:'' ,I6)') K
  STOP
  END
```

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

main()
{
  /* Variable deklarieren */
  int i,k;
  /* hier beginnt das Programm */
  printf("gib Zahl ");
  scanf("%d",&i);
  k= i * 3;
  printf("Ergebnis:%d",k);
  return(0);
}
```

Listing 1 (b1.c)

```
/*
 * ein sehr langer Kommentar
 * weiter mit dem Kommentar
 * hier soll er fertig sein
 */
Kommentare dürfen nicht geschachtelt
werden:
/*
/* verboten
*/
*/
```

Listing 2

```
#include <stdio.h>

main()
{
  int i;
  printf("gib Zahl ");
  scanf("%d",&i);
  printf("\nfertig");
  return(i);
}
```

Listing 3 (b2.c)

```
b2
@echo off
if errorlevel 3 goto 13
if errorlevel 2 goto 12
if errorlevel 1 goto 11
:11
echo level 1
goto end
:12
echo level 2
goto end
:13
echo level 3
:end
```

Listing 4 (b2c.bat)

Negatives aufgelistet werden soll. C ist ganz ohne Zweifel als Sprache eine hervorragende Wahl für viele Programmieraufgaben und der Autor würde diese Artikelserie nicht verfassen, wenn er nicht von C im Rahmen seiner Zweckbestimmung überzeugt wäre. Mit C kann man exzellent Systemdienstprogramme schreiben (den 1001. Editor oder den 527. Compiler), aber auch für nicht zu große technisch-wissenschaftliche Probleme ist C gut geeignet. Zweifelsohne sehr gut geeignet als Allround-Sprache (aber nicht als Universal-sprache, die alles andere überflüssig macht !!) auch für den Hausgebrauch. Unschlagbar für Menüprogramme ala GEM, Windows oder OS/2 Presentation-Manager. Fein für Drucker- und Plottertreiberprogramme. Prädestiniert für UNIX-Programme, da sämtliche UNIX-Systemdienste als C-Functions erreichbar sind.

3. Ein erstes Programm

Betrachten wir nochmal das obige Beispiel b1.c (Listing 1).

Zunächst handelt es sich um ein Hauptprogramm. Dieses muß "main" heißen, andere Namen sind nicht zulässig. Die nachgeschalteten leeren Klammern nach main besagen, daß hier keine Argumente von der Commandline übernommen werden. C ist blockstrukturiert, die Klammern { und } entsprechen etwa dem Pascal begin/end. Jede Variable muß deklariert werden: i und k sind vom Typ int, der wie gesagt, maschinenabhängig ist. Jeder Befehl wird mit ; beendet. Kommentare werden in /* .. */ eingeschlossen; sie können über mehrere Zeilen gehen (Listing 2)

Gerechnet wird prinzipiell wie in BASIC, FORTRAN und Pascal, aber es gibt keinen Potenzoperator wie in FORTRAN oder BASIC. Das würde mit der Bibliotheksfunktion pow(x,y) erledigt werden.

printf() und scanf() sind die Standard-Functionen für Ein- und Ausgabe über Konsole.

Weitere Beispiele folgen.

Da selbst das Hauptprogramm main als Function behandelt wird, ist es gut (aber nicht zwingend nötig), es mit return(0) zu beenden. Damit liefert das Hauptprogramm den Wert 0 an das Environment des Betriebssystems zurück; dort kann man z.B. bei DOS bzw. OS/2 mit IF ERRORLEVEL .. weitere Aktionen einleiten. Beispiel b2.c (Listing 3).

Dazu kann nun ein Batchfile b2c.bat gehören, etwa wie folgt (Listing 4).

Was ist im Gegensatz zu anderen Sprachen zu beachten: Zweiganz entscheidende Dinge für den Beginner: C ist casesensitiv, I ist nicht i! Es ist üblich, daß in C generell kleingeschrieben wird, aber es geht auch:

```
int I,K;
K= I * 3;      usw.
```

Die Funktionen printf() und scanf() müssen kleingeschrieben werden, da sie so in der Bibliothek definiert sind. Das muß nicht immer so sein: Mitunter werden Funktionen heute "gemischt" definiert, z.B. heißt eine OS/2-Funktion, die einen String mit Bildschirmattributen an eine definierte Position schreibt: VioWrtCharStrAtt (kein Witz, muß genau so aufgerufen werden).

Zweitens: C läßt so ziemlich alles zu, ob sinnvoll oder nicht! Vereinfacht: Während Pascal praktisch jede Unsauberkeit aufspürt, ist FORTRAN bereits viel freizügiger und C erlaubt so gut wie jede auch unsinnige Konstruktion. Wenn daher ein C-Programm den Compiler einwandfrei passiert, heißt das noch lange nicht, daß es funktionsfähiger Code ist. Um Unsauberkeiten aufzuspüren, gibt es in UNIX das Dienstprogramm lint, das verschärften Syntaxcheck durchführt.

Es soll das obige Beispielprogramm übersetzt werden. Das geht beispielsweise so (Programmquelle möge b1.c heißen):

Turbo-C (DOS): mit EDIT editieren, mit Ctrl-K-D beenden, auf COMPILE gehen, dort

MAKE EXE-FILE, dann RUN MS-C (DOS und OS/2):
normal: cl b1.c (compilieren & linken)

oder zum Testen: cl /Od /W3 b1.c (Optimizer lahmlegen: Übersetzen geht schneller, erhöhten Warning Level)

mit b1 starten.

UNIX System V: cc -ob1 b1.c (compilieren & linken) mit b1 starten.

Nun kann man damit etwas experimentieren:

1) eine Zahl größer als 10922 eingeben. Es ist $10922 \cdot 3 = 32766$. Da k vom Typ int ist, also bei einem PC von -32768 bis + 32767 laufen kann, läuft k bei größeren Werten über. Beliebte Fehlerquelle bei C-Programmen. Abhilfe: i und k von Typ long definieren (= 4 Bytes bei DOS und OS/2). Dann muß Programm b1.c wie Beispiel b3.c aussehen (Listing 5).

Beachte, daß die Formatan-gabe %d nunmehr %ld heißen muß.

2) bei scanf das &i weglassen, also (falsch !!): scanf("%d",i); Gibt unter OS/2 eine Protection Violation, wird also korrekt abgefangen, führt unter DOS zu falschen Ergebnissen oder Rechnerabsturz. Ruhig mal ausprobieren.

3) die Zeile "#include <stdio.h>" weglassen. Hoffentlich kommen Warnings.

4) Mal ein Semikolon weglassen und einen Schreibfehler einbauen.

5) "return(0)" weglassen. Warning betrachten (es kommt doch hoffentlich eine ?). Programm ist damit aber lauffähig.

4. Variablentypen

Vorbemerkung: Ab jetzt werden alle Erläuterungen so gehalten, daß sie mehr oder weniger leicht verständlich sind. Dafür wird auf Vollständigkeit und Strenge verzichtet. Um das Studium eines guten C-Buches bzw. Studium der Compilerhandbücher kommt der Leser ohnehin nicht herum. Bestimmte Operatoren, wie die bitmanipulierenden Operatoren, oder komplexere Dinge wie structures/unions werden hier ausgelassen. Außerdem werden die Beispielprogramme ähnlich BASIC/FORTRAN-Programmen aufgebaut und nicht in der C-typischen, sehr kompakten Form, wie sie Kernighan/Ritchie (für Fortgeschrittene) propagieren.

In C gibt es Variablentypen mit 1, 2, 4 und 8 Bytes Länge. Die von DOS- und OS/2-Systemen sind in Tabelle 1 dargestellt.

Das sind die üblichen Benennungen. Aus Kompatibilitätsgründen zu älteren Compilern gibt es alternative Namen, z.B. für short: short int, signed short und signed short int, oder für double auch long float. Damit braucht man sich aber i.A. nicht zu befassen. Mehrere Datentypen können auch als "unsigned" deklariert werden, z.B. unsigned char geht von 0 ..255. Diese unsigned-Datentypen können bei manchen Compilern Probleme bereiten, sodaß man sie am Anfang besser völlig meidet.

Bemerkenswerterweise gibt es in C nicht den Datentyp des Strings. Strings werden mit char-Vektoren bzw. Pointern auf char abgebildet. Obwohl (oder weil ?) der eigentliche Datentyp String fehlt, ist C ganz besonders stark in der Byte-, Character- und Stringverarbeitung.

Zum Rechnen muß man wissen:

1. Alle Gleitkomma-Berechnungen werden in C immer mit doppelter Genauigkeit ausgeführt, auch wenn die Operanden nur vom Typ float sind. Die meisten Mathematik-Bibliotheks-Funktionen gehen vom Typ double aus.

2. Das Ergebnis einer Berechnung hat immer den Typ des höchstwertigen Operanden. Wird z.B. eine short-Variable mit einer long-Variablen multipliziert, so ist das Ergebnis long.

3. Ansonsten geht das Rechnen wie in BASIC, FORTRAN und Pascal. Im Zweifelsfall klammern. Lieber ein Klammernpaar zuviel als zuwenig.

4. Potenzoperator ist, wie erwähnt, nicht vorhanden. Durch Funktion double pow(double x, double y) ersetzen. Die Funktion pow() übernimmt also zwei double-Werte und liefert einen double-Wert zurück. Mathematische Funktionen wie sin(), fabs() etc. sind nicht im Sprachkern definiert, sondern Teil der mitgelieferten Bibliotheken. Nicht immer kompatibel! Für einen C-Compiler für einen 8-Bit Mikrocontroller werden i.A. weniger Funktionen mitgeliefert als für einen UNIX-C-Compiler.

5. C- Operatoren

Wie erwähnt, hier lediglich eine Auswahl:

Die C-Operatoren haben, wie andere Sprachen, eine definierte Hierarchie. Beispiel: bei $3 + 5 * 7$ hat * eine höheren Level als +. Klammern haben noch höheren Level: $(3 + 5) * 7$ (Tabelle 2)

6. Kontroll-Strukturen

6.1. Die if-Anweisung

Die Merkregel ist: do if true. Dazu ein Beispiel b4.c (Listing 6)

Die Zahlen i und k werden durch ein oder mehrere Blanks getrennt eingegeben, z.B. 4 2. Das Beispiel zeigt: Sind mehrere Statements nach einem if oder else auszuführen, dann muß dieser Block in { } eingeschlossen werden. Folgt nur ein einziges Kommando, so kann die Klammerung entfallen. Weitere Details: typischerweise wird zum Programm-Stop die Funktion exit() verwendet; sie verlangt das Header-File process.h (So etwas wird immer in den C-Libraries angegeben). Es ist UNIX-üblich, eine 0 bei fehlerfreier Ausführung zu übergeben. Beachte in den printf-Funktionen: Die sog. Escape-Sequenz \n sorgt für eine neue Zeile. Andere gern

```
#include <stdio.h>

main()
{
    long i,k;

    printf("gib Zahl ");
    scanf("%ld",&i);
    k= i * 3;
    printf("\Ergebnis:%ld",k);
    return(0);
}
```

Listing 5 (b3.c)

Name	wie FORTRAN & BASIC	Größe	Wertebereich	Bemerkungen
char	CHARACTER	1 Byte	-128 .. +127	
int	INTEGER*2	2 Bytes	-32768 .. +32767	maschinenabhängig !
short	INTEGER*2	2 Bytes	-32768 .. +32767	wie int bei PCs
long	INTEGER	4 Bytes	± 2147483647	
float	REAL	4 Bytes	ca. ± 3.4E±38	
double	DOUBLE PRECISION	8 Bytes	ca. ± 1.7E±308	

Tabelle1

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

main()
{
    int i,k;
    printf("\n\agib Zahlen i und k: ");
    scanf("%d%d",&i,&k);

    if(i > k)
    {
        if(i == 4)
        {
            printf("\ni ist größer k
                und dabei gleich 4");
            exit(0);
        }
        else
            printf("\ni ist größer k");
    }
    else
    {
        printf("\ni ist nicht
            größer als k");
        exit(0);
    }
}
```

```
    }
    printf("\ni und k waren:
        %d %d",i,k);
    return(0);
}
```

Listing 6 (b4.c)

gebrauchte Escape-Sequenzen:

\a Ton ausgeben, siehe obiges Programm

\b Backspace

\f Formfeed

\n schon bekannt: Newline

\r Carriage return, d.h. an Zeilenanfang ohne Vorschub

Wie gibt man einen Backslash selbst aus: Mit \\. Wozu soll das gut sein ?

z.B.: printf("\nDas Directory \\\neu wird angelegt");

Eine Hexadezimalzahl als Byte, z.B. zur Druckersteuerung wird durch \xddd, wobei ddd die Zahl ist, ausgegeben. Einen Ton kann man also mit \a oder \x007 ausgeben.

6.2 Die while- Anweisung

Beispiel b5.c (Listing 7).

Hier ist die Funktion der while-Bedingung einsichtig. Tatsächlich ist aber so, daß die while-Schleife ausgeführt wird, solange ihre Bedingung nicht 0 ist; bei C ist logisch-True eine Zahl ungleich 0, logisch-False ist 0 ! Das gilt genauso für if

Level	Operator	Benennung	Beispiel	
1	()	Funktion	pow()	An den Klammern erkennt C, daß pow eine Funktion ist
1	[]	Array	k[5]	k ist ein Vektor mit 5 Elementen
2	-	arithm.Minus	3 * (-5)	
2	!	logisches NOT	if(!i)	
2	*	Pointer	int *i	i ist ein Pointer auf integer
2	&	Adresse von	&i	für scanf oder call by reference
2	++	Increment	i++,++i	erhöhe i um 1
2	--	Decrement	i--,--i	erniedrige i um 1
2	sizeof	Groesse	sizeof(int)	z.B. für Speicherallokierung
2	(type)	Type-Cast	(long)i	i wird als long benutzt
3	*, /	Multiplikation und Division wie allgemein bekannt		
4	+, -	Addition und Subtraktion wie allgemein bekannt		
5	<	kleiner	if(i < j)	
5	>	größer	if(i > j)	
5	<=	kleiner gleich	if(i <= j)	
5	>=	größer gleich	if(i >= j)	
6	==	gleich	if(i == j)	wenn i gleich j
6	!=	ungleich	if(i != j)	wenn i ungleich j
7	&&	logisches UND	if(i == j && k == m)	
8		logisches OR	if(i == j k == m)	
9	=	Zuweisung	i = j + k;	
	+=	+ Zuweisung	i += 3;	entspricht i= i + 3
	-=	- Zuweisung	i -= 3;	entspricht i= i - 3
	*=	* Zuweisung	i *= 3;	entspricht i= i * 3
	/=	/ Zuweisung	i /= 3;	entspricht i= i / 3

Tabelle 2

und für alle weiteren C-Kontroll-Strukturen. Daher kann das Beispiel auch so verfasst werden (und nähert sich so schon dem typischen (unverständlichen) C-Stil an..), Beispiel b6.c (Listing 8).

Was ist hier neu:

1) Es wird die Eigenschaft ausgenutzt, daß logisch-true ungleich 0 ist: while(i) heißt: Solange i wahr (d.g. ungleich 0) ist, do ..

2) der Decrement-Operator wurde direkt in die printf-Anweisung verlegt. Jetzt ist aber wichtig, ob i— oder —i. i— heißt "nach Aktion" decrementieren, —i "vor Aktion". Also ist die Anzeige bei i— : 10..1, bei —i dagegen 9..0 ! Wenn der Operator alleine steht, wie bei dem längeren Beispiel b5.c, ist das belanglos.

3) Da mit while nur noch eine Anweisung ausgeführt wird, können wie immer die Klammern entfallen.

Nebenbei erlaubt C wie BASIC und Pascal, Beispiel b7.c (Listing 9).

Man kann beliebig viele Anweisungen in eine Zeile quetschen (Programm wird davon übrigens nicht schneller ..), damit es ja recht unverständlich wird. Empfehlung: 1 Anweisung pro Zeile.

Eine ähnliche Kontrollstruktur wie while ist in Listing 10 dargestellt.

Die Anweisung wird solange ausgeführt, bis der Ausdruck false (also 0) wird. Etwa wie Pascal's REPEAT..UNTIL.

6.3 Die for-Schleife

Sie weicht in ihren Möglichkeiten am meisten von ähnlichen Konstruktion anderer Sprachen ab, d.h. vom BASIC FOR..NEXT, von FORTRAN DO..CONTINUE und Pascal for..to..do. Ein simples Beispiel b8.c (Listing 11).

Die Wirkung ist wie bei b5.c und b6.c . Aber die Möglichkeiten sind viel weitergehend, denn die C-for-Schleife sieht wie Listing 12 aus.

Daher sind auch derartige

Konstruktionen möglich, b9.c (Listing 13).

Derartige for-Schleifen sind nicht gerade zu empfehlen, aber ohne weiteres funktionsfähig.

Die kürzeste Form der for-Schleife ist for(;;). Das ist zulässig und ist eine Endlosschleife.

6.4 Das ungeliebte goto

Hier werden die Puristen und Chef-Informatiker sofort aufschreiben, denn goto ist in diesen Kreisen hoffnungslos verpönt. Tatsache ist, daß goto in allen Programmiersprachen unverzichtbar ist. Eine typische Anwendung: Man stelle sich eine fünffach geschachtelte for-Schleife vor. Wenn in der innersten Schleife eine Fehlerbedingung auftritt, ein Grenzwert erreicht ist oder dgl., soll die g a n z e Schleifenkonstruktion verlassen werden. Ein Fall, der garnicht selten ist (Listing 14).

Wollte man nur von der innersten auf die nächste Schleife (die m-schleife) springen, so hätte man das anstelle goto mit break machen können. Aber um mehrere Schleifen auf einmal zu verlassen, hilft nur goto. Selbstverständlich kann man das auch mit allen möglichen Konstruktionen, die das goto vermeiden, ebenfalls "irgendwie" machen, aber dadurch wird das Programm nur langsam und unverständlich. Natürlich ist goto so sparsam wie möglich anzuwenden, aber es gibt in der Tat Aufgaben, bei denen ein goto die sauberste und schnellste Lösung ist. Übrigens können beim obigen Beispiel alle Klammernpaare bis auf Klammernpaar 1 entfallen. Sie sind nur der Übersicht wegen gezeigt. Es ist kein schlechter Stil, generell nach C-Kontrollstrukturen Klammernpaare vorzusehen, denn

```
for(i= 10; i > 0; i--)
{
    printf(" %d ", i);
}
```

ist in der Praxis einfach besser les- und erkennbar als

```
for(i= 10; i > 0; i-)
    printf(" %d ", i);
```

```
#include <stdio.h>

main()
{
    int i= 10;
    /*Variable können gleich bei der Typen-Deklaration initialisiert werden*/
    printf("\nI = ");
    while(i > 0)
    {
        printf(" %d ", i);
        i--;
        /* hier ginge auch --i */
    }
    printf("\nfertig..\n");
    return(0);
}
```

Listing 7 (b5.c)

```
#include <stdio.h>

main()
{
    int i= 10;
    /*Variable können gleich bei der Typen-Deklaration initialisiert werden*/
    printf("\nI = ");
    while(i)
        printf(" %d ", i--);
    printf("\nfertig..\n");
    return(0);
}
```

Listing 8 (b6.c)

```
#include <stdio.h>
main()
{int i= 10; printf("\nI = ");
while(i) printf(" %d ", i--);
printf("\nfertig..\n"); return(0);}

do
{
    ..Anweisungen, bei nur einer keine Klammern nötig
}
while(Ausdruck);
```

Listing 9 (b7.c)

Listing 10

```
#include <stdio.h>

main()
{
    int i;
    printf("\nI = ");
    for(i= 10; i > 0; i-)
        printf(" %d ", i);
    printf("\nfertig..\n");
    return(0);
}
```

Listing 11 (b8.c)

```
for (Zuweisung ; Endebedingung;
      Zuweisung)
{
    ..Anweisungen, bei nur einer keine
    Klammern nötig
}
Listing 12
```

```
#include <stdio.h>

main()
{
    int i,j;
    for(i= 10, j= 0; i > 0 && j < 12;
        i--, j+=2)
        printf("\ni= %d j= %d ",i,j);
    printf("\nfertig..\n");
    return(0);
}
Listing 13 (b9.c)
```

```
for(i=0; i < MAXI; i++)
{
    for(j=0; j < MAXJ; j++)
    {
        for(k=0; k < MAXK; k++)
        {
            for(m=0; m < MAXM; m++)
            {
                for(n=0; n < MAXN; n++)
                { /* Klammernpaar 1 */
                    .. Anweisungen..
                    if(epsilon < 0.00001)
                    {
                        goto epsilon_erreicht
                    }
                } /* Klammernpaar 1 */
            }
        }
    }
}
epsilon_erreicht:
.. weitere Anweisungen
Listing 14
```

```
#include <stdio.h>

main()
{
    char ctaste;

    printf("gib Taste: ");
    scanf ("%c",&ctaste);

    switch(ctaste)

    {
        case 'a': printf("\nein kleines A");
                  break;
        case 'A': printf("\nein großes A");
                  break;
        case '1': printf("\ndie Ziffer 1");
                  break;
        case '2':
        case '3':
```

```
        case '4':
        case '5':
        case '6':
        case '7':
        case '8':
        case '9':
        case '0': printf("\nein Ziffer von
                      2..0 ..");
                  break;
        default: printf("\nwar weder a, A
                      noch Ziffer ..");
    }
    printf("Ergebnis: %c",ctaste);
    return(0);
}
Listing 15 (b10.c)
```

```
#include <stdio.h>
#include <math.h> /* für z.B. pow() */

main()
{
    /* Variable deklarieren */
    double ergebnis;
    float ex;
    int i,imax
    /* einlesen */
    printf("gib einen Exponent an (float)
           und einen Zähler (int): ");
    scanf ("%f %d",&ex,&imax);
    /* rechnen und in Schleife ausgeben */
    for(i = 0; i < imax; i++)
    {
        ergebnis= pow((double)i, (double)ex);
        printf("\ni= %d, Exponent= %f,
              Ergebnis= %f",i,ex,ergebnis);
    }
    /* programm-ende */
    return(0);
}
Listing 16 (b11.c)
```

Da manche Editoren (z.B. Norton-Editor) zusammenhängende Klammernpaare finden, ist dies auch praktisch, um geschachtelte Strukturen besser zu überschauen.

6.5 Die switch-Anweisung
Entspricht grob dem Pascal "case of". Beispiel b10.c (Listing 15).

Das Beispiel versteht man am besten, wenn man damit etwas experimentiert.

7. Zusammenfassung und Ausblick
Die erste Folge der kleinen C-Reihe zeigte die wesentlichen Teile von C, (Variablentypen, Operatoren und Kontrollstrukturen) mit denen bereits gut gearbeitet werden kann. C kann man nur durch relativ viel Übung erlernen. Anhand der gezeigt-

ten Beispiele experimentieren. Dabei auch mit Gleitkommazahlen arbeiten: Beispiel b11.c (Listing 16).

In der nächsten Folge wird das I/O-System und Funktionen von C behandelt, die dritte Folge wird sich mit Vektoren und Pointern befassen und im vierten Teil wird gezeigt werden, wie mit getrennten Source-Files und Utilities größere Projekte angegangen werden können.

/1/ Plum, T.: Das C- Lernbuch. Carl Hanser, München- Wien und Prentice-Hall, London. 1985

/2/ Kernighan, B., Ritchie, D.: Programmieren in C. Carl Hanser, München- Wien. 1983

Dr.-Ing. Frank Rieg
Darmstadt
(2002)

CCD e.V.

ATLAHM

Funktion:

Verlangsamung eines AT-Rechners, z.B. für Spiele oder wenn Ausgaben zu schnell über den Bildschirm huschen und sich anders schlecht anhalten oder verlangsamen lassen, besonders wenn man während des Rollens Wichtiges aus einer Masse von Unwichtigem heraus erkennen will.

Format:

ATLAHM [dauer] [A] [H]

- dauer = Betrag der Verzögerung des PC-AT. Zulässiger Bereich 1-4000.
- A = Schaltet AT-LAHM ab und entfernt aus dem Speicher.
- H = Zeigt einen Hilfe-Text an.

Typ:

Speicherresident, wieder entfernbar

Hinweise:

Wenn ATLAHM geladen wird, dann benutzt es den hochauflösenden Zeitgeber, den es bei einem AT-kompatiblen Rechner gibt, um Programme zu verlangsamen, wenn sie auf einem AT oder 386er-Rechner zu schnell laufen. Deshalb ist ATLAHM hardware-abhängig, es läuft nur auf ATs und kompatiblen Rechnern, die diesen Zeitgeber (Timer) haben.

Der Zeitgeber ist Bestandteil der Echtzeituhr des AT. Wenn man diese Funktion freigibt, dann wird 1024 mal in der Sekunde die Interrupt-Leitung IRQ8 betätigt, die über den Interrupt-Controller den Interrupt 70H auslöst. Beim Auftreten des INT 70H wird in ATLAHM eine einstellbare Zeitschleife durchlaufen. Damit steht dem Hauptprogramm weniger Rechenzeit zur Verfügung und es wird langsamer.

Dieser hochauflösende Zeitgeber hat zwei Vorteile gegenüber dem gewöhnlichen Zeitgeber eines PC, der etwa 18,2 mal in der Sekunde den INT 1CH auslöst:

1. Dader Aufruf 1024 mal statt 18 mal in der Sekunde erfolgt, ist die Verzögerung wesentlich gleichmäßiger und die Programme lassen sich auch erheblich stärker verlangsamen.
2. Es ist recht verbreitet, daß Programme den INT 1CH abfangen und umlenken und so andere Verzögerungsprogramme unwirksam machen oder diese stören oder von ihnen gestört werden. Das kann bei ATLAHM kaum passieren, da es nur sehr wenige Programme gibt, die den INT 70H beeinflussen.

CCD e.V.

CommandPost

Funktion:

CommandPost ist eine MS-WINDOWS Applikation, die das MS-DOS Fenster von WINDOWS ersetzt. CommandPost bietet zahlreiche zusätzliche Funktionen, so kann z.B. die Menüzelle völlig frei erstellt werden.

Typ:

Microsoft Windows Applikation

Hinweise:

CommandPost kann durch Aufrufen von CMDPOST.EXE vom MS-DOS Fenster aus gestartet werden. CommandPost ersetzt daraufhin das MS-DOS Fenster, welches verschwindet. Wird in der WIN.INI Datei

LOAD=CMDPOST.EXE

angegeben, startet WINDOWS gleich mit CommandPost statt dem MS-DOS Fenster.

Bei deutschen WINDOWS-Versionen müssen in der WIN.INI Datei drei Zeilen ergänzt werden:

[CmdPost]

WinTitle=MS-DOS-Fenster

WinSession=Session

Außerdem sollten von den Programmen ABLAGE.EXE und SYSTEMST.EXE Kopien mit ihren englischen Originalnamen CLIPBRD.EXE und CONTROL.EXE angelegt werden (Das wird auch von vielen anderen englischen Applikationen verlangt).

Die Menüzelle ist in der Datei CMDPOST.MNU hinterlegt. Sie muß an das individuelle System angepasst werden. Am besten mit dem Notizblock (notiz.exe) von WINDOWS editieren, sonst gibt's Schwierigkeiten mit den Umlauten.

Funktion:

DCACHE ist ein residenter Festplattencache. Ein Cache beschleunigt die Festplattenzugriffe, indem er Teile des Platteninhalts im Speicher hält.

Format:

DCACHE [/OFF] [/ON] [/V] [/Mx] [/E] [/Hx]

Hinweise:

Mit dem Parameter /Mx läßt sich die Größe des Caches in Kilobyte einstellen, x kann die Werte 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 oder 8192 annehmen. Voreinstellung ist 64.

Wird der Parameter /E angegeben legt *DCACHE* den Cache im EMS-Speicher und nicht im Hauptspeicher an.

Der Parameter /Hx gibt an, für welche Festplatte der Cache angelegt wird: 0 für die erste Platte, 1 für die zweite Platte.

Mit /OFF und /ON läßt der Cache an- und ausschalten. Mit /U kann *DCACHE* deinstalliert werden.

DCACHE ? zeigt eine Hilfsmeldung an.

Quelle:

PC-Magazine (New York), Volume 7, Heft 17, Seite 255ff.

Wenn *ATLAHM* noch nicht eingerichtet ist, bewirkt der Aufruf *ATLAHM* die Anzeige einer kurzen Hilfestellung. Diese kann man jederzeit mit *ATLAHM* H erhalten. Ebenso bekommt man bei fehlerhaften Eingaben diese Hilfe.
Eingerichtet wird *ATLAHM* mit "ATLAHM dauer" z.B. *ATLAHM* 300

Damit kann man auch jederzeit die Verzögerung ändern. Der sinnvolle Bereich von "dauer" (Anzahl der Schleifen) hängt vom jeweiligen Rechner ab, je schneller der Rechner ist, um so größer sollte bzw. kann dieser Wert sein. Es sind eigentlich Vorkerbrungen getroffen, daß sich der INT 70H nicht selber überholt, aber bei zu großen Werten ist nicht auszuschließen, daß der Rechner extrem langsam wird, stehen bleibt oder sich aufhängt. Da hilft dann nur der RESET-Taster (so man hat) oder der Netzschalter. Also "dauer" langsam steigern und die Reaktion des Rechners beobachten.

Mit *ATLAHM* A kann man *ATLAHM* wieder abschalten und aus dem Speicher entfernen. Das sollte normalerweise ohne Probleme möglich sein. Falls nach *ATLAHM* weitere Programme Speicherresidentgemacht wurden, wird zwar der von *ATLAHM* belegte Speicher wieder frei gegeben, aber der kann von DOS nur eingeschränkt genutzt werden. Wenn man *ATLAHM* nicht wie vorgesehen abschaltet sondern mit Programmen zur Verwaltung Speicherresidenter Programme wie z.B. MARK & RELEASE entfernt, dann bleibt der Zeigeger und INT 70H aktiv und bewirkt eine dauerhafte Verlangsamung von ca. 10%. Das sollte man also vermeiden.

Wenn *ATLAHM* eingerichtet ist, dann bewirkt der Aufruf *ATLAHM* die Anzeige der eingestellten Verzögerung.

Die Anwendung von *ATLAHM* ist vor allem in folgenden beiden Bereichen sinnvoll: Zum einen gibt es Programme, die bewußt Wartezeiten enthalten, um die Reaktionszeit des Anwenders zu berücksichtigen. Aber statt einen Zeigeger (mit fester Zeit) zu benutzen, nehmen sie einfache Verzögerungsschleifen, die von der Geschwindigkeit des Rechners abhängen. Dies sind vor allem (ältere) Spiele.

Dann gibt es Programme, die (ohne bewußte Verzögerung) größere Datenmengen auf den Bildschirm schauen. Wenn man da neben einem schnellen Prozessor und schnellem Hauptspeicher auch noch eine schnelle Grafikkarte hat, kann man oft gar nicht schnell genug CTRL-S oder verwandte Tasten drücken, um die Ausgabe anzuhalten. Oder man will im Schnellgang eine Datei nach gewissen Merkmalen durchmusteren, und dann geht es ein bißchen zu schnell...

Quelle:

Vorlage war AT-SLOW von David Keil, Better Software Co. Davon ausgehend stark überarbeitet von Werner Dworak (607).

Funktion:

Emuliert Expanded Memory nach der EMS 4.0 Spezifikation im Extended Memory eines AT.

Format:

DEVICE = [d:] [pfad] EMS40.SYS [nnn]

Hinweise:

Um EMS40.SYS zu installieren, muß die obige Zeile im CONFIG.SYS plaziert (und neu gebootet) werden. Als Einheiten-treiber läßt sich EMS40.SYS nur deinstallieren, wenn es vorher aus dem CONFIG.SYS entfernt und neu gebootet wird. Mit dem Parameter nnn kann man einstellen, wieviel Kilobyte vom Expanded Memory in Expanded Memory umgewandelt werden sollen. Voreinstellung ist 384. Da EMS40.SYS den Expanded Memory nur emuliert, ist es wesentlich langsamer als echter Expanded Memory. Außerdem kann eine Emulation nicht 100% kompatibel zur EMS 4.0 Spezifikation sein.

Quelle:

PC-Magazine (New York), Volume 8, Heft 12, Seite 277ff.

Funktion:

DIRMATCH dient zum Abgleichen zweier Verzeichnisse. Die zwei Verzeichnisse werden nebeneinander angezeigt, mit den Dateien in alphabetischer Reihenfolge, wobei identische Dateinamen in der gleichen Zeile stehen. Die Datei mit dem neueren Datum ist hell dargestellt.

Die Dateien können kopiert und verschoben werden, eine Markierfunktion gibt es auch.

Format:

DIRMATCH Quelle Ziel [/D] [/A]

Hinweise:

Quelle und Ziel stehen für das Quell- und Zielverzeichnis. Es kann jeder gültige Verzeichnisname angegeben werden. Dem Verzeichnisnamen kann noch eine Dateispezifikation angehängt werden.

/D, bzw. /A zeigen nur die verschiedenen (Different), bzw. gleichen (Alike) Dateien an.

Quelle:

PC-Magazine, New York, Volume 8, Heft 16

Funktion:

SMOOTH ist ein Datei-Browser für EGA- und VGA-Karten. Die Besonderheit von *SMOOTH* ist das fließende Scrolling. Normalerweise wird der Text Zeile für Zeile gescrollt, bei *SMOOTH* schiebt sich jede Zeile pixelweise ins Bild.

Format:

SMOOTH dateispez [/w] [/snn] [/cmmn]

Hinweise:

Die angegebene Datei wird von *SMOOTH* angezeigt.

Wird der /w Parameter angegeben unterdrückt *SMOOTH* das 8. Bit (nur für WordStar-Dateien).

Der /snn Parameter bestimmt die Scroll-Geschwindigkeit. Die Voreinstellung ist /S03.

Mit dem /cmmn Parameter lassen sich Vorder- und Hintergrundfarbe wählen.

mmm = Vordergrundfarbe + (Hintergrundfarbe * 16)

0 =	Schwarz	8 =	Dunkelgrau/Schwarz
1 =	Blau	9 =	Hellblau
2 =	Grün	10 =	Hellgrün
3 =	Türkis	11 =	Helltürkis
4 =	Rot	12 =	Hellrot
5 =	Violett	13 =	Helviolett
6 =	Braun	14 =	Gelb
7 =	Hellgrau/Weiß	15 =	Hellweiß

Mit den Pfeiltasten kann man im Text manövrieren.

SMOOTH ist nicht kompatibel mit Sidekick und der PrtSrc-Funktion.

Quelle:

PC-Magazine, New York, Volume 8, Heft 3

Funktion:

SPLIT teilt übergroße Dateien auf mehrere Disketten auf.

Format:

SPLIT [s:] [pfad] dateiname d:
UNSPPLIT s: [d:] [pfad]

Hinweise:

SPLIT kopiert die Datei [d:] [pfad] dateiname auf das Laufwerk d:. *SPLIT* arbeitet mit allen von DOS unterstützten Diskettentypen zusammen. Ist eine Diskette voll, verlangt *SPLIT* die nächste. Bis zu 99 Disketten kann *SPLIT* füllen. Auf der ersten Diskette legt *SPLIT* das 1 Kb große Programm *UNSPPLIT* an. *UNSPPLIT* wird zum Restaurieren der Datei benötigt. Dabei restauriert *UNSPPLIT* die Datei vom Laufwerk s: auf das Laufwerk d:. Bei dem Ziellaufwerk kann zusätzlich ein Pfad angegeben werden.

Quelle:

PC-Magazine, New York, Volume 8, Heft 9

CCDUTIL

CCD LEX-File für den HP-71

Teil 3

Die aktuelle Version von CCDUTIL ist CCD:2A. Ich bin dazu übergegangen, bei funktionellen Änderungen (neue Befehle) die Versions-NUMMER zu erhöhen und bei anderen Änderungen (Fehlerkorrekturen) den Buchstaben, wobei ich die Version "A" als "1A" betrachte (es gab auch eine Version 1B).

Bereits in der Version 1B sind zwei Bugs beseitigt worden: die "-1", die DSP, RPT und KBD zurückgeben konnten sah nur so aus; ein Vergleich wie "DSP=-1" lieferte immer Null als Ergebnis. Außerdem war es doch nicht der Weisheit letzter Schluß, wie von mir in Heft 5/88, Seite 5 vorgeschlagen wurde, zum Löschen der Anzeige in das Nibble an Adresse #2F442 einen Wert ungleich Null zu schreiben: beim nächsten INPUT-Befehl wird die Programmausführung unterbrochen.

Wenn man beim Studium der IDS mal genauer hinsieht, findet man eine mit NEEDSC bezeichnete Adresse (#2F94A). Ein Wert von Null bedeutet, daß keine Zeichen zum Display geschickt wurden, seit die ENDLIN-Taste das letzte mal gedrückt wurde. Dort schreibt SETSTS nun Null hinein, wenn die Option "Anzeige löschen" bei STS\$ gesetzt war.

Die Funktion FILE\$ ist (mal wieder) erweitert worden: ist die Länge des Stringparameters genau 32 Zeichen, so wird er nicht als Filename sondern als Massenspeicherkatalogeintrag interpretiert. Nun kann man z.B. das gesamte Directory einer Diskette mit einer ENTER-Anweisung einlesen, mit der FILE\$-Funktion die Katalogeinträge auswerten lassen und das ganze sortiert ausgeben.

Dieses Feature gab es auch schon in der letzten Version, jedoch war diese Erweiterung so kurzfristig, daß sie nicht mehr im Artikel in Heft 3/89 berücksichtigt werden konnte, da dieser bereits gesetzt wurde. Neu ist allerdings, daß nun bei exakter Schreibweise (nur Kleinbuchstaben und keine Leerzeichen) auch "workfile" und "keys" gefunden werden. Und jetzt - taaaaa... - ist es erstmals möglich, herauszufinden, welcher File einer Kanalnummer zugeordnet ist, wenn man FILE\$ keinen String sondern eine Zahl -eben die Kanalnummer- übergibt.

Und weil es billig zu haben war (etwa 100 Bytes), habe ich auch die Funktion FTYPE\$ aus dem DATACOMM-Modul übernommen. FTYPE\$ wird -genau wie FILE\$- ein Filename oder eine Kanalnummer übergeben und man erhält einen fünf

Zeichen langen String, der den Filetyp repräsentiert (zB. "TEXT", "BASIC" oder "KEY "). Kennt der HP71 den Filetyp nicht, so wird, genau wie bei CAT\$ eine bis zu fünfstellige Zahl in Stringform zurückgegeben. Wird der angegebene File nicht gefunden, erzeugt FTYPE\$ nicht wie FILE\$ einen Fehler, sondern gibt einen Nullstring zurück; das ist nützlich, wenn man herausfinden will, ob ein File bereits existiert.

Im Gegensatz zur Funktion im DATA-COMM-Modul findet dieses FTYPE\$ auch "workfile" und "keys". Es benutzt die gleiche LEX-ID und Tokennummer wie die Original-Funktion, sodaß CCDUTIL im Speicher vor DCLEX liegen muß (siehe HP71-Benutzerhandbuch Seite 112 - Suchordnung für Files) um in den Genuß der erweiterten Funktionalität zu kommen.

Wirklich neu sind die Funktionen REC# und RECLen#. Eigentlich sollten sie REC und RECLen heißen, aber daß muß HP wohl nicht gefallen haben: als ich die Antwort auf mein "permanent resource allocation request" erhielt, waren die beiden Funktionen einfach umbenannt.

Mit REC# kann man erfahren, wieviele Records in einem File sind und welches die momentane Recordnummer ist. Im Gegensatz zu den Funktionen FILESZR und DRECORDS von HP kann REC# nicht nur auf einen Filetyp angewandt werden, sondern auf TEXT, DATA und SDATA-Files und REC# ist etwa dreimal so schnell wie FILESZR.

RECLen# kann ebenfalls auf diese Filetypen angewandt werden und liefert die Länge des längsten Records (bei TEXT-Files), die Länge des momentanen Records bzw. den im momentanen Record verbleibenden Platz (DATA-Files) und wenn man am Ende des Files angekommen ist "-1". Eine genaue Parameterbeschreibung ist der untenstehenden Tabelle zu entnehmen.

In der folgenden Tabelle sind alle Befehle aus CCDUTIL aufgeführt. Bei den bereits an anderer Stelle beschriebenen Befehlen steht nur ein Hinweis auf den entsprechenden Artikel. Vor jedem Befehl sind in hexadezimaler Schreibweise die LEX-ID/Tokennummer des Befehls aufgeführt. Optionale Parameter sind in eckigen Klammern aufgeführt. Numerische Parameter werden durch "#" und Stringparameter durch "\$" dargestellt. Ist wahlweise beides möglich, so ist dieses durch "#/

"\$" dargestellt. Ich überlasse es dem geneigten Leser, sich Gedanken über die nicht benutzten Tokennummern zu machen.

CCDUTIL ist jetzt 2842 Bytes groß. Ich habe deshalb darauf verzichtet, den Hexdump mit abzdrukken; wer ist denn schon so wahnsinnig, 6000 Ziffern abzutippen? Zusammen mit KEYPOLL ist unser LEX-File nun immerhin schon fast 4k Bytes groß, etwa ein siebentel der Größe des JPC-ROM's (hätten wir sieben aktive LEX-File-Programmierer, dann ... - tja, wer weiß?).

Bei vielen Leuten sehe ich ein LEX-File namens CCDGP auf allen möglichen Disketten rumgeistern. Dieses ist eine uralte Vorversion von CCDUTIL, die ich auf der Mitgliederversammlung '86 unter's Volk gebracht habe. Sie hat noch tausend Bugs (können auch zwei oder drei mehr sein), verkehrte LEX-ID und Tokennummern und teilweise gleiche Befehle wie CCDUTIL, jedoch mit anderen Namen. Vergeßt am besten diesen File.

F6/01 FTYPE\$(#/\$) (Kanalnummer/Filename) gibt den Filetyp eines Files in Stringform zurück. Wird der angegebene File nicht gefunden, so wird ein Nullstring zurückgegeben. Als Parameter wird eine Kanalnummer oder ein Filename erwartet.

E1/D0 DDELAY
Siehe Prisma 4/88, Seite 7

E1/D1 PDELAY
Siehe Prisma 4/88, Seite 7

E1/D2 SWDSP AEON/OFFUE
Siehe Prisma 4/88, Seite 7

E1/D3 SWPRT AEON/OFFUE
Siehe Prisma 4/88, Seite 7

E1/D4 DSP Siehe Prisma 4/88, Seite 7

E1/D5 KBD Siehe Prisma 4/88, Seite 7

E1/D6 PRT Siehe Prisma 4/88, Seite 7

E1/D7 SETSTS \$ (Status)
Siehe Prisma 3/89, Seite 24

E1/D8 STS\$(#) (Statusoption)
Siehe Prisma 3/89, Seite 24

E1/D9 FILE\$(#/\$) (Kanalnummer/Filename/Massenspeicherkatalogeintrag) gibt den Katalogeintrag eines Files zurück (ähnlich CAT\$). Wird ein numerischer Parameter übergeben, so wird dieser als Kanalnummer aufgefaßt. Ein Stringparameter wird, solange er nicht genau 32

Zeichen lang ist, als Filename mit optionalem Einheitsspezifikator interpretiert. Ist die Länge des Strings genau 32 Zeichen, so werden diese als Katalogeintrag eines Massenspeichers aufgefaßt (siehe HP-IL Benutzerhandbuch Seite 234 - Anhang D: Massenspeicher-Kompatibilität). Siehe auch Prisma 4/88, Seite 7 und 3/89, Seite 24

E1/DA REC#(##) (Kanalnummer) Der Betrag des Parameters gibt eine Kanalnummer an. Bei einem positiven Vorzeichen wird zurückgegeben, aus wievielen Records der File besteht. Bei negativem Vorzeichen wird die Nummer des momentanen Records zurückgegeben. Als Filetypen sind TEXT, DATA und SDATA zulässig. Ein TEXT-File darf sich nicht auf einem Massenspeicher befinden.

E1/DB RECLN#(##AE,##UE) (Kanalnummer, Recordnummer) Der Betrag des Parameters gibt eine Kanalnummer an. Bei negativem Vorzeichen wird die Länge

des momentanen Records bzw. bei DATA-Files der bis zum Ende des momentanen Records verbleibende Platz zurückgegeben. Das ist ungleich der Recordlänge, wenn bereits aus dem Record gelesen bzw. in den Record geschrieben wurde. Der Wert ist bei SDATA-Files immer 8. Existiert der momentane Record nicht (es wurde z.B. der letzte Record gelesen), so wird -1 zurückgegeben. Ist das Vorzeichen positiv, so wird bei nicht vorhandenem zweiten Parameter bei Textfiles die Länge des längsten Records, bei DATA-Files die bei der Erzeugung spezifizierte Recordlänge und bei SDATA-Files 8 zurückgegeben.

Ist in dem File kein Record vorhanden, so wird -1 zurückgegeben. Mit dem zweiten Parameter kann eine Recordnummer angegeben werden. Gibt es diesen Record, so wird seine Länge, ansonsten -1 zurückgegeben. Als Filetypen sind TEXT,

DATA und SDATA zulässig. Ein TEXT-File darf sich nicht auf einem Massenspeicher befinden.

E1/DD MESSAGE #AE,##UE (Fehlernummer, Option)
Siehe Prisma 4/88, Seite 7

E1/DE KEYCTLAE(\$AE,##UE)UE (Tastenspezifikator, Option)
Siehe Prisma 4/88, Seite 7

E1/E3 COUNT(\$,\$) (Suchstring, gesuchter Teilstring)
Siehe Prisma 4/88, Seite 8

E1/E4 CUT(\$,\$,#AE,##UE) (Suchstring, Terminator, 1. Position, 2. Position)
Siehe Prisma 4/88, Seite 8

E1/E5 NPOS(\$,\$,#) (Suchstring, gesuchter Teilstring, Position)
Siehe Prisma 4/88, Seite 8

Matthias Rabe
Teichsheide 13
4800 Bielefeld
Tel. (0521) 32 44 74

KEY-FILES

mit Sonderzeichen
von H.R. Wuttke

Diese Sonderzeichen sind sowohl auf dem Display als auch auf dem ThinkJet möglich.

Erforderlich ist das LEX-File "ROMAN8LX" im Hauptspeicher oder in den Ports des HP-71B.

In kleiner Abwandlung zu den Handbüchern und zu J. Horn verwende ich nicht die Tasten ZEICHEN, sondern die Tasten NUMMERN (Benutzerhandbuch Seite 123). Um Sonderzeichen sowohl auf dem HP-71B als auch auf dem ThinkJet erscheinen zu lassen, müssen die CHR\$-Nummern des DRUCKER-Handbuches benutzt werden! Bei den Sonderzeichen, die mit "f" oder "g" USER aufgerufen werden können, gibt es zwei Arten:

Sonderzeichen, die mit CHR\$(...)-Nummern des DRUCKER-Handbuches abgeleitet werden, oder

Sonderzeichen, die aus einem String (Text) bestehen.

Als Beispiel wird folgendes Programm vorgeschlagen:

```
KEY '#1' ,CHR$(204); =ä
KEY '#2' ,CHR$(216); =Ä
KEY '#3' ,CHR$(206); =ö
KEY '#4' ,CHR$(218); =Ö
KEY '#5' ,CHR$(207); =ü
KEY '#6' ,CHR$(219); =Ü
KEY '#7' ,CHR$(222); =ß
KEY '#8' ,CHR$(254); =±
KEY '#9' ,CHR$(179); =°
KEY '#10' ,CHR$(172); =~
```

```
KEY '#15' ,CHR$(247); =1/4
KEY '#16' ,CHR$(248); =1/2
KEY '#17' ,CHR$(253); =»
KEY '#18' ,CHR$(251); =«
KEY '#19' ,CHR$( ); =CHR$(
KEY '#20' ,CHR$(189); =§
KEY '#21' ,CHR$(252); =■
KEY '#22' ,CHR$(124); =|
KEY '#23' ,CHR$(176); =`
KEY '#24' ,DESTROY; =DESTROY
```

Vergessen Sie nicht, danach CHARSET ROMAN8\$ ENDLINE in den HP-71B einzugeben. Und dann SECURE KEYS. Löschen:

KEY '#X' ENDLINE

X = Tastennummer die gelöscht werden soll.

Frequenzen im HP-71B

(Version 1.BBBB)
von H.R. Wuttke

Im HP-71B gibt es zwei Frequenzoszillatoren:

Einen freischwingenden Oszillator, der die CPU (U4) steuert und einen Quarz-Oszillator, der mit dem Master RAM (U1) das Display und die Uhr steuert.

Der CPU-Oszillator ist ein freischwingender Oszillator mit zwei Kondensatoren und einer Spule. Die Frequenz beträgt etwa 2.5 MHz, die intern durch 4 heruntergeteilt wird. Die Arbeitsfrequenz der CPU liegt somit zwischen 600 und 650 kHz. Sie kann mit dem LEX-File CLKSPD

(?) abgefragt werden (Für Bastler: Die Arbeitsfrequenz kann bis zu 1 MHz betragen). HP leistet für die CPU keinen Service.

Der Quarz-Oszillator im Master RAM (U1) schwingt auf der Frequenz von 32.768 kHz. Das Master RAM steuert das Display. Das Display wird mit 512 Hz getaktet. Ferner steuert der Quarz die Uhr und das Datum.

Der Quarz ist ein fernöstliches Elaborat, wie es auch in Armbanduhren zu finden ist. Der Quarz ist nicht gealtert, er schwingt

nicht im Vakuum und der Temperaturkoeffizient (TK) ist miserabel.

Da lohnt es sich nicht, die Uhr nach den exzellenten Abstimmungsanweisungen der Handbücher zu stellen, denn schon 1°C Temperaturveränderung macht die zeitaufwendige Feinjustierung zunichte.

Quellen: HP Hardware IDS, Wuttke-Quarze

H.R. Wuttke
Hainerweg 271
6000 Frankfurt 70

...UND NOCH EIN LEXFILE:

BASICLEX

von Michael Fiedler

Wie in meinem letzten Artikel zu STRUC2 angekündigt, möchte ich heute den File BASICLEX vorstellen. Dabei handelt es sich um eine neue Möglichkeit, BASIC-Programme zu listen, sodaß sie wesentlich übersichtlicher als bisher dargestellt werden. Hauptsächlich im Zusammenhang mit Strukturbefehlen des HP-71B (FOR...NEXT, DEFFN...ENDDEF), aber auch mit den neu hinzukommenden aus dem Lexfile STRUC2, zeigt sich die Stärke dieses Lexfiles, da es auf Wunsch automatisch Einrückungen entsprechend den Blockstrukturen vornimmt und damit die Übersichtlichkeit wesentlich erhöht.

Der File stammt von den Autoren Pierre David und Janick Taillan hier nach einer Idee von Jean-Pierre Bondu, alles Mitglieder des PPC Paris. Eine erste Version wurde in JPC 38 veröffentlicht, diese vorliegende in JPC 53. Die Filegröße ist 1838 Bytes (1819 laut CAT), es sind 4 Schlüsselwörter vorhanden:

BASICLEX ID=E1 SIZE=1838 Bytes

5A DBLIST Stmt 225090

5B PBLIST Stmt 225091

5C RENUMREM Stmt 225092

*

BASICLEX ID=E1 (see:ID=E1)

C8 INDENT Word 225200

*

Die korrekte Syntax lautet:

DBLIST [file][,start][,end] [INDENT x] [TO file]

Der Befehl gleicht also im ersten Teil genau dem normalen LIST-Befehl. Neu hinzu gekommen ist die optionale Angabe

INDENT x

wobei mit x angegeben wird, um wieviele Stellen ein Block eingerückt werden soll. Bei hierarchisch ineinander verschachtelten Blöcken addiert sich natürlich die Einrückungstiefe entsprechend auf, daher sollte der Wert bei komplexen Programmen nicht zu groß gewählt werden. Sonst bleibt unter Umständen nicht mehr genügend Platz für längere Zeilen.

TO file

bedeutet, daß man (optional) das Listing auch direkt in ein Textfile übertragen

lassen kann, statt es an das Display zu schicken. Das ist immer dann nützlich, wenn man das Listing anschließend etwa mit einem Texteditor weiterbearbeiten will, oder per Modem verschicken will, oder ähnliches.

Die Syntax des Befehls PBLIST ist identisch mit der von DBLIST. Hier wird lediglich die gerade aktuelle PRINTER IS.. Einheit angesprochen, während bei DBLIST das aktuelle Display angesprochen wird. Bei Verwendung der "TO file" Option besteht zwischen den Befehlen kein Unterschied.

Hier nun eine Auflistung, was bei Verwendung der DBLIST/PBLIST-Befehlen im Listing anders gemacht wird als beim LIST-Befehl:

- 1.) Sofern INDENT und ein Wert angegeben ist: Einrückung von Blockstrukturen.
- 2.) Blöcke mit DATA-Zeilen erhalten jeweils eine Leerzeile vor und nach dem Block.
- 3.) Vor jedem LABEL wird eine Leerzeile eingefügt.
- 4.) REM-Zeilen werden ohne Zeilennummern angezeigt. Die erste REM-Zeile wird jeweils mit einem Bindestrich "-" markiert.
- 5.) Unterprogramme (SUBs, also solche die mit CALL aufgerufen werden) stehen in einem mit einem waagrechten Strich abgetrennten Bereich am Ende des Listings.

Der Befehl RENUMREM wirkt im Prinzip wie der RENUMBER-Befehl, nur daß er bei der Neunummerierung alle Zeilen ausnimmt, die mit ! oder REM beginnen, also reine REM-Zeilen sind. Diese werden unabhängig von der angegebenen Schrittweite immer nur in 1-er Schritten neu nummeriert. Der Sinn liegt darin, daß REM Zeilen unter DBLIST und PBLIST ohne Zeilennummern angezeigt werden. Dadurch bleibt die Kontinuität der restlichen Zeilennummern erhalten.

Als Beispiel für einen PBLIST-Ausdruck möchte ich gleich noch ein BASIC-Programm vorstellen, das auch für sich ganz nützlich sein kann. Es handelt sich um ein Basis-Umwandlungsprogramm, das Zahlen beliebiger Basis (zw. 2 und 16) ineinander umwandelt.

Das Programm BASUM wird einfach

gestartet und befindet sich danach in einer Endlosschleife, die immer am Eingabebefehl INPUT anhält. Dabei wird links im Display immer das Ergebnis der letzten Berechnung oder Eingabe angezeigt und rechts kann eine neue Eingabe gemacht werden. Als Eingaben sind zulässig:

- 1.) eine Zahl (wird als Dezimalzahl interpretiert). Sie wird gespeichert.
- 2.) eine Basis in der Form Bxx, also z.B. B16 für hexadezimal, B6 für Basis 6, etc. Zusätzlich versteht das Programm die Kürzel B für binär, O für oktal, D für dezimal (normalerweise nicht nötig, da Default Ein- u. Ausgabe ohnehin dezimal) und H für hexadezimal. Nach Eingabe einer Basis wird die vorhin gespeicherte Zahl in dieser Basis ausgegeben.
- 3.) eine Zahl mit Angabe der Basis dahinter. Diese Zahl wird nach dezimal umgewandelt, gespeichert und angezeigt.
- 4.) nichts (nur ENDLIN): die gespeicherte Zahl wird dezimal angezeigt.

Beispiele:

Umwandeln der Zahl 23 dez. nach hexadezimal:

23 (ENDLINE) H (ENDLINE)

Umwandeln der Binärzahl 10101 nach oktal:

10101B (ENDLINE) O (ENDLINE)

Umwandeln der Zahl 363A der Basis 12 nach dezimal:

363A B12 (ENDLINE)

Als Beispiel für PBLIST lege ich zwei Listings des Programms BASUM bei, eines herkömmlich mit PLIST erstellt, das andere mit PBLIST INDENT 2.

Beide Programme, BASICLEX und BASUM sind auch bei unserer Programm-bibliothek auf magnetischen Medien erhältlich. Ebenso gibt es dort jetzt eine neue, verbesserte Version von MAKEFILE, die etwas mehr Komfort bietet als die aus PRISMA 7/86. U.a. sind die Tasten rechts neben dem Ziffernblock mit den Buchstaben A-F belegt, und auch ein Unterbrechen des Eintippens ist jetzt möglich, mit Fortfahren zu einem beliebigen späteren Zeitpunkt.

```

BASUM      BASIC      888 Bytes,  HP-71B,  PLIST

10 !                      Basis - Umwandlungsprogramm
11 ! Wandelt beliebige ganze Zahlen zur Basis n in Zahlen zur Basis m um.
12 ! n und m: zwischen 2 und 16
13 ! Benötigt Lexfile STRUC2 (oder JPCROM)
14 !
20 LOOP
30 INPUT ' /?';I$
40 FOR I=LEN(I$) TO 1 STEP -1
50 IF I$[I,I]>='A' THEN
60 'SETBAS':
70 B$=I$[I]
80 SELECT UPRC$(B$)
90 CASE 'H'
100 B=16
110 CASE 'D'
120 B=10
130 CASE 'O'
140 B=8
150 CASE 'B'
160 B=2
170 CASE ELSE
180 IF B$[2]<'A' THEN B=VAL(B$[2]) ELSE B=0
190 IF B$[1,1]# 'B' OR B>16 OR B<=1 THEN DISP HTD('X')
200 END SELECT
210 I$=I$[1,I-1]
220 WHILE I$[LEN(I$)]=' '
230 I$=I$[1,LEN(I$)-1]
240 END WHILE
241 !
250 IF I>1 THEN
260 'EINUM':
270 D=0
280 FOR J=0 TO LEN(I$)-1
290 G=HTD(I$[LEN(I$)-J][1,1])
300 IF G>=B THEN DISP HTD('X')
310 D=D+B^J*G
320 NEXT J
330 I=-7
331 !
340 ELSE
350 'AUSUM':
360 O$='' @ D1=D
370 WHILE D1
380 D1=D1/B @ G=FP(D1)*B
390 O$[1,0]=DTH$(G)[5]
400 D1=IP(D1)
410 END WHILE
420 I=-8
421 !
430 END IF
440 END IF
450 NEXT I
460 'AUSGABE':
470 SELECT I
480 CASE -9
490 DISP O$;' ' ;CHR$(NUM(UPRC$(B$))+32);B$[2];' ' ;
500 CASE -8
510 DISP D;
520 CASE ELSE
530 IF LEN(I$) THEN D=VAL(I$)
540 DISP D;
550 END SELECT
560 END LOOP

```

BASUM BASIC 888 Bytes, HP-71B, PBLIST INDENT 2

Basis - Umwandlungsprogramm

Wandelt beliebige ganze Zahlen zur Basis n in Zahlen zur Basis m um.

n und m: zwischen 2 und 16

Benötigt Lexfile STRUC2 (oder JPCROM)

```

20 LOOP
30 INPUT '/?';I$
40 FOR I=LEN(I$) TO 1 STEP -1
50 IF I$[I,I]>='A' THEN

60 'SETBAS':
70 B$=I$[I]
80 SELECT UPRC$(B$)
90 CASE 'H'
100 B=16
110 CASE 'D'
120 B=10
130 CASE 'O'
140 B=8
150 CASE 'B'
160 B=2
170 CASE ELSE
180 IF B$[2]<'A' THEN B=VAL(B$[2]) ELSE B=0
190 IF B$[1,1]# 'B' OR B>16 OR B<=1 THEN DISP HTD('X')
200 END SELECT
210 I$=I$[1,I-1]
220 WHILE I$[LEN(I$)]=' '
230 I$=I$[1,LEN(I$)-1]
240 END WHILE
-
250 IF I>1 THEN

260 'EINUM':
270 D=0
280 FOR J=0 TO LEN(I$)-1
290 G=HTD(I$[LEN(I$)-J][1,1])
300 IF G>=B THEN DISP HTD('X')
310 D=D+B^J*G
320 NEXT J
330 I=-7

340 ELSE

350 'AUSUM':
360 O$='' @ D1=B
370 WHILE D1
380 D1=D1/B @ G=FP(D1)*B
390 O$[1,0]=DTH$(G)[5]
400 D1=IP(D1)
410 END WHILE
420 I=-8
-
430 END IF
440 END IF
450 NEXT I

460 'AUSGABE':
470 SELECT I
480 CASE -9
490 DISP O$; ' ';CHR$(NUM(UPRC$(B$))+32);B$[2]; ' ';
500 CASE -8
510 DISP D;
520 CASE ELSE
530 IF LEN(I$) THEN D=VAL(I$)
540 DISP D;
550 END SELECT
560 END LOOP

```

SERIE 70

Hexdump-Listing BASICLEX

(zum Abtippen benoetigen
Sie ein Hexdump-Lade-
programm wie z.B. MAKEFILE
aus Prisma 7/86!)

BASICLEX L ID#E1 1838 Bytes

0123 4567 89AB CDEF ck

```

000: 2414 3594 34C4 5485 AE
001: 802E 1000 0000 0000 E7
002: B3E0 01EA 5C52 6000 80
003: F920 0000 0000 0000 AE
004: 02D2 00DF 000B 200D 36
005: E106 8B00 DB44 24C4 B6
006: 9435 45A5 B052 4C49 7A
007: 4354 5B5F 2554 E455 90
008: D425 54D4 C51F F1E8 6A
009: C8C0 0000 F710 0000 F3
00A: 0000 0000 0000 000B A0
00B: 94E4 4454 E445 8C1F 7D
00C: FAE0 1361 BB98 F214 E2
00D: 8134 7690 4257 2114 63
00E: 1775 408F 5CC3 05A1 E9
00F: 867F 1181 313F 9623 B1
010: 38D2 0F20 8FDC 6305 15
011: D07E E08F B394 075C 7E
012: 0401 79C0 4828 D271 ED
013: 3034 B98F 27A8 08FB 61
014: EC20 8F5C C305 AC42 70
015: B34C 98F2 7B60 8F51 23
016: D208 FD96 3062 BF8F 78
017: B394 08F3 E320 0F61 34
018: 04F1 10CF C005 F700 6F
019: 0003 027C 1045 2736 DB
01A: 0869 018F DC63 0521 00
01B: 7050 038F E7A2 0400 9E
01C: 8D27 1301 3615 6494 DC
01D: EE0B 4615 4413 6018 C6
01E: DB2E 2031 3F96 2000 F5
01F: 135F E1E8 C259 1220 D9
020: 2001 8497 D008 FDC6 04
021: 305D A859 8F6E 8403 04
022: 1F09 6601 311F AEA8 D5
023: D51D 208D F3E2 0848 E2
024: 8FCE 2508 F3E3 201F 21
025: D103 F630 FE45 0008 C9
026: F957 5085 866D F868 6A
027: D031 C28F 8EC2 0171 7E
028: 8F51 1506 EDF7 A403 2A
029: 545F 4022 58F3 2450 56
02A: 1718 F957 506B BF77 EF
02B: 203D 94E4 4454 E445 5B
02C: 022D 8F32 4501 758F 86
02D: 2295 0609 F868 0031 16
02E: 028D 8EC2 08D9 3850 A1
02F: D4FF FCCD FF31 F168 10
030: 108D A939 043F FF3B C1
031: DFF3 1F01 F078 F214 9B
032: D1F5 C8F2 AF21 5DA1 24
033: DADE 6145 1743 3999 09
034: 9145 1DEE AF21 5D61 BD

```

```

035: ED55 F147 1E4E 8F14 96
036: 514A 8F50 4505 606E 30
037: D08F 3E32 01F8 30FE D3
038: 3603 F080 008F D2F9 9B
039: 04D1 8F36 F904 3113 7D
03A: 71F4 E8F2 1456 5BF6 B5
03B: 25F1 FAD8 F27C 5614 61
03C: 1174 1417 F464 5914 72
03D: 11C4 1478 BE78 8D02 1B
03E: 9E01 658F 681F 08FB 4E
03F: C631 57E1 F5C8 F214 3C
040: 16F5 F161 8FD2 F904 7D
041: 721F 178F 2AFB 1557 91
042: 17F1 5178 F36F 904C 74
043: 033B 3006 ACE1 FFC8 4F
044: F230 1155 0671 F1B4 08
045: E8F2 1461 3517 E143 75
046: F41C E8F1 4770 4D48 E7
047: FBF8 6043 417F 1371 C9
048: 3514 3C21 99E1 44D7 DD
049: 19AD 142D 817E 1371 2C
04A: 3517 1D08 FDFE F04F AB
04B: 0860 A063 C067 4E1B 03
04C: 4E8F 2137 1441 9FC1 BA
04D: 5249 4896 1F17 8F21 7A
04E: 577A F7D2 3192 8F4C C8
04F: 4804 6C11 91F5 D8F2 AA
050: 1451 F188 F215 3713 71
051: 5151 717F 3510 0004 3C
052: 15D5 17F1 74D2 CE15 83
053: D313 71F0 D8F2 1451 1A
054: B4E8 F214 2199 E146 40
055: 8BE7 2131 D015 B319 E2
056: FD14 68B6 2113 78FD 6D
057: A670 1354 908D 84A8 C4
058: 0171 7CA4 AE81 198A A9
059: A811 BFE8 F214 AB6A B1
05A: 550A E014 8D43 1509 A6
05B: 6252 1BEE 8F21 5A03 F2
05C: 0190 621D 215C 0193 6A
05D: FE61 5C01 9EE1 5203 0C
05E: 0290 6113 1709 6180 71
05F: D215 40D4 8F3E 3202 13
060: 0D10 30D6 0400 A050 C6
061: 0806 0690 0062 A01F B1
062: FE8F 2AE2 14D7 9031 56
063: FF85 F214 7135 3FD2 DC
064: D2D2 D2D2 D2D2 D2AF 23
065: AD23 1088 F2B0 B113 1F
066: 71F4 95F2 1457 OD26 22
067: B301 4B31 0F96 6041 5A
068: F4F8 F230 115D 05D1 4B
069: 1FEE 8F21 5B03 0115 74
06A: D090 6606 5101 F3F8 76
06B: F230 1155 0630 01B3 A2
06C: F8F2 1524 9486 0776 85
06D: 21BF E8F2 AF01 4A19 97
06E: 5CAF 2146 8FB8 CE0C A1
06F: 6195 F144 8F5A 2105 06
070: 606F FB1A F85F 1460 4D

```

```

071: 6135 3F02 0202 0202 B0
072: 0202 02AF AD98 F2B0 2B
073: B113 7134 3502 0202 A2
074: 15C5 1F4E 8F21 4713 40
075: 5D01 5B33 4000 108B 2C
076: E91F 68BE F0F6 8BE5 65
077: 0161 1611 6113 61BF 09
078: 85F2 1448 FF6F 4007 E5
079: D71F 5F8F 2143 CA13 12
07A: 0135 15A7 3702 0202 26
07B: 0215 C715 97DB 1FF8 A9
07C: 5F21 4314 5C0C 0174 8E
07D: 1411 B4E8 F214 2131 4F
07E: 1717 D42A E831 8096 49
07F: 6606 BF01 1A8A AF01 88
080: 9FE1 4AA6 A148 3170 E6
081: 9657 7137 061F F85F BE
082: 2147 1353 7020 2020 8B
083: 215D 7175 19EE 1520 1C
084: 3021 5409 0290 31D2 FE
085: 14D3 1121 4B96 2801 0F
086: 7154 F173 1331 311C 2D
087: 38F8 61B1 1B49 5F21 97
088: 3714 4071 3576 A1AE OD
089: 8447 3180 9624 51BF 95
08A: E8F2 14E1 11B6 2550 73
08B: AE21 12A6 214C 3120 FC
08C: 9658 0148 51C3 1109 FD
08D: 658B 14B3 10F9 66CA D9
08E: 3011 94F1 5C06 E9F1 8D
08F: F4E8 F214 3131 8F13 CB
090: 0015 5017 1137 1F4E 44
091: 8F21 451F 4F8F 2153 AA
092: 4AC2 1554 1D3F 1514 E9
093: 7B00 6A0C 8F53 4101 98
094: BFC8 F215 2494 C338 5B
095: F245 811A 495F 1421 A7
096: 8414 6EAA F1D8 81D8 6C
097: 548F F1E7 18D1 CD71 05
098: 1A49 5F14 2131 1731 8C
099: 8414 68F4 01B1 1B49 5C
09A: 5F2A F014 2184 146E 23
09B: A81C D88F 42A7 1146 6A
09C: 1351 5938 2281 DD08 64
09D: 3240 E4D2 304C 4CA1 28
09E: 6414 6C21 44D2 10B1 B2
09F: B5D8 F214 6190 D142 AA
0A0: 8F7F 3104 E01B 0D8F E6
0A1: 2140 016B E814 A311 BE
0A2: F966 0016 1D01 5A31 52
0A3: 6301 AF21 0910 A14B 2B
0A4: 310F 9620 0171 D214 0E
0A5: F133 C213 1061 7114 46
0A6: B8F3 E320 FE04 03CA 98
0A7: 804C 2C01 C6E0 9B3F 30
0A8: 06C6 116F C11A D9A0 CF
0A9: ABBB 02CF 90CF 311F 42
0AA: D911 006E 1117 114B 68
0AB: 311E 966F E171 14B8 FA
0AC: F3E3 2034 1300 6C20 C3

```

```

0AD: 4472 054F 5036 D104 7B
0AE: 6320 16C2 0269 1024 7D
0AF: E200 06FC 0D2E 610A 35
0B0: 64C0 D2E6 1091 0A66 1A
0B1: B0D2 3021 0A6A A017 0F
0B2: 114B 8F50 4504 A030 36
0B3: 1902 D0D2 E610 9678 51
0B4: 0D23 0210 96B7 0D2E FA
0B5: 6109 3110 AEA6 C60D C1
0B6: 2E61 0A31 20AE A6A5 8E
0B7: 0176 14B9 68D0 3130 56
0B8: AEA6 440D 2E61 0A31 2F
0B9: 40AE A623 0315 0AEA 48
0BA: 6720 3160 AEA6 C103 7C
0BB: 170A EA61 1031 80AE C1
0BC: A660 0AE0 0713 503A 9B
0BD: 17FF 2C5F F136 1087 A6
0BE: C027 E128 F39B 7085 B4
0BF: 1852 8F35 7A14 B11F 1C
0C0: 976F 211A 1457 7023 05
0C1: 1E16 BF11 1013 07FA A7
0C2: 1102 D2E6 F210 8109 A9
0C3: F2F2 F210 B797 1436 D6
0C4: 1007 F614 9510 1756 FE
0C5: 14F4 D613 2102 8F4E DE
0C6: FF08 6060 6119 1371 91
0C7: 2A13 47D3 1472 0523 BF
0C8: B142 0044 91D6 8F4E 8B
0C9: FF0D 015B 3870 5010 D3
0CA: 37B2 1137 1128 A232 F6
0CB: 1301 838F 7887 0D21 2E
0CC: 5F31 108B 6808 CF07 0C
0CD: F1FC 65F2 147D 711A 02
0CE: 135D 215F 3134 1180 D2
0CF: 611A 1351 11D8 1378 80
0D0: BFD2 135D 015B 311B 59
0D1: 8BEC 1110 8B28 1041 14
0D2: 3610 8D2E 6109 54C6 8A
0D3: 1701 5931 7514 B1C5 10
0D4: 31CF 962C 1316 E962 90
0D5: 3111 007D 605E 6066 CE
0D6: 9100 715D 305E 6061 7D
0D7: 1023 99E7 0C01 0004 CB
0D8: 2017 3D21 4F13 3CA1 AD
0D9: 3114 B171 90CC E6E5 27
0DA: F078 5184 28F3 57A1 6C
0DB: 8C8C 7F14 A161 311F 75
0DC: 9660 0D01 5A31 6303 D3
0DD: 1FC6 5F21 47D7 1CE1 AD
0DE: 43D2 3113 CA13 1017 12
0DF: 2108 FEEB 6044 1832 0F
0E0: 005C 08F6 2770 5008 3B
0E1: D393 901F D55F 2147 9E
0E2: 1351 7E14 3F43 4412 44
0E3: E08A 6618 F73B 908F CC
0E4: 7887 0147 540D 21B8 20
0E5: E7F2 15C3 03F 9D

```

Michael Fiedler
Friedrichstr. 17
6070 Langen

“J“: Berechnung des Abstandes eines Punktes P zu der Geraden 1.
Berechnung des Lotfußpunktes.

Ergebnis:
– d = ... R/S FDL: ...
Abstand Fußpunkt des Lotes von P auf die Gerade 1.

WICHTIGE BEMERKUNG:

Um eine gute Ablesbarkeit der errechneten Schnittpunkte und Geradengleichungen zu gewährleisten, klammert der Rechner automatisch den größten gemeinsamen Teiler aller drei Komponenten aus:

so wird zum Beispiel: $(3/3/3) = 3(1/1/1)$
oder: $(9/6/12) = 3(3/2/4)$
aber auch: $(1/1/3) = 1/3(3/4/5)$

Kann bei einem derartigen Ausdruck nichts ausgeklammert werden, so gibt der Rechner den jeweiligen Punkt in der Form $1(\dots)$ aus. Ist die auszuklammernde Konstante nicht ganzzahlig, wird der Punkt vom Rechner zwar in der Form $1/X(\dots)$ ausgegeben, doch ertönt gleichzeitig TONE O. Fährt man anschließend mit R/S fort, so steht die ausgeklammerte Konstante im FIX 9 – Format im X-Register. Mit dieser Zahl ist also der in Klammern angegebene Vektor zu multiplizieren, um den gesuchten Punkt zu erhalten:

Beispiel: Der gegebene Punkt sei $(3\sqrt{2} / -\sqrt{18} / \sqrt{2})$

Der Rechner gibt diesen Punkt als $1/1(3/-3/1)$ aus, teilt dem Benutzer jedoch durch den TONE / mit, daß die ausgeklammerte Konstante nicht geradzahlig ist. Wird das Programm jedoch mit R/S fortgesetzt, so hält der Rechner mit 1,414213561 in der Anzeige an. Der gesuchte Punkt ist folglich:

$$P: \sqrt{2} (3/-3/1)$$

Läßt sich auch durch Ausklammern kein ganzzahliger Klammerausdruck erzielen, so wird das Ergebnis im FIX 2 – Format ausgegeben:

z. B. $P(1,417,00/0,00)$ für $P(\sqrt{2}/7/0)$

Durchgerechnete Beispiele:

1. Man gebe eine Parameterdarstellung von $E: 2x - y + 3z = 4$ an.

Vorgehensweise:

i) Eingabe der Ebene in der Koordinatenschreibweise:

XEQ“a“ $“a_1 \uparrow a_2 \uparrow a_3 \uparrow a_4“$
2 ENTER –1 ENTER 3 ENTER 4 R/S

ii) Umrechnung in die Parameterform:

XEQ“A“ $“A_1 = “$
R/S $“E_1: (0,00:0,00:0,00)“$
R/S $“+ \lambda(-8,00:-2,00:8,00)“$
R/S $“+ \mu(58,00:64,00:74,00)“$

Eine Parameterdarstellung der Ebene E lautet demnach:

$E: (0/0/0) - \lambda(-8/-2/8) + \mu(58/64/74)$ oder gekürzt:
 $E: (0/0/0) - \lambda(-4/-1/4) + \mu(29/32/37).$

2. Man gebe eine Koordinatendarstellung von $E: (1/1/1) + \lambda(3/0/4) + \mu(4/4/0)$ an.

Vorgehensweise:

i) Eingabe der Ebene in der Koordinatenschreibweise:

XEQ“A“ $“A_1 = “$
1 ENTER 1 ENTER 1 R/S $“U_1 = “$
3 ENTER 0 ENTER 4 R/S $“V_1 = “$
4 ENTER 4 ENTER 0 R/S —

ii) Umrechnung in die Koordinatendarstellung:

XEQ“a“ $“a_1 \uparrow a_2 \uparrow a_3 \uparrow a_4“$
R/S $“-16,0000 X_1“$
R/S $“16,0000 X_2“$
R/S $“12,0000 X_3“$
R/S $“=12,0000“$

Eine Koordinatendarstellung der Ebene E lautet demnach:

$E: -16 x_1 + 16 x_2 + 12 x_3 = 12$ oder auch
 $E: -16 x + 16 y + 12 z = 12$

HINWEIS:

Wird XEQ“a“ oder XEQ“b“ ausgeführt und anschließend die R/S – Taste gedrückt, so wird die Koordinatenschreibweise der 1. bzw. 2. Ebene im oben erläuterten Format ausgegeben. Analog hierzu wird nach dem Aufruf von XEQ“A“ bzw. XEQ“B“ und anschließend gedrückter R/S – Taste die Parameterdarstellung der 1. bzw. 2. Ebene angezeigt.

3. Man bestimme Schnittwinkel und Schnittgerade der Ebenen $E_1: x + 2y + 3z = 4$ und $E_2: (2/3/4) + \lambda(1/2/3) + \mu(1/0/0)$

i) Eingabe der 1. Ebene in Koordinatenschreibweise:
XEQ“a“ $“a_1 \uparrow a_2 \uparrow a_3 \uparrow a_4“$
1 ENTER 2 ENTER 3 ENTER 4 R/S —

ii) Eingabe der 2. Ebene in Parameterdarstellung:

XEQ“B“ $“A_2 = “$
2 ENTER 3 ENTER 4 R/S $“U_2 = “$
1 ENTER 2 ENTER 3 R/S $“V_2 = “$
1 ENTER 0 ENTER 0 R/S —

iii) Ermitteln der Schnittmenge zwischen Ebene 1 und Ebene 2:

XEQ“F“ $“SCHNITT \sphericalangle = 90,0“$
R/S $“GS: 1/3(10:1:0)“$
R/S $“+ \mu(-13:2:3)“$

Die beiden vorgegebenen Ebenen schneiden sich also unter einem Winkel von $90,0^\circ$ und bilden so die Schnittgerade $g: (\frac{10}{3} / \frac{1}{3} / 0) + \mu(-13/2/3).$

4. Man bestimme den Abstand der Geraden $g: (0/0/0) + \mu(1/1/1)$ und der Ebene $E: (2/1/0) + \lambda(1/1/1) + \mu(0/1/0).$

i) Eingabe der Geraden:

XEQ“C“ $“A_1 = “$
0 ENTER 0 ENTER 0 R/S $“U_1 = “$
1 ENTER 1 ENTER 1 R/S —

ii) Eingabe der Ebene in der Parameterdarstellung:

XEQ“A“ $“A_2 = “$
2 ENTER 1 ENTER 0 R/S $“U_2 = “$
1 ENTER 1 ENTER 1 R/S $“V_2 = “$
0 ENTER 1 ENTER 0 R/S —

iii) Ermitteln der Schnittmenge zwischen Ebene 1 und Gerade 1:

XEQ“G“ $“PARALLEL“$
R/S $“ABSTAND: -1,41“$

Der Abstand der Geraden g zur parallelen Ebene E beträgt demnach 1,41 (also $\sqrt{2}$ Einheiten). Das negative Vorzeichen gibt an, das Gerade und Ursprung des Koordinatensystems auf verschiedenen Seiten der Ebene liegen.

5. Man bestimme den Schnittpunkt der Geraden $g: (1/1/2) + \lambda(1/7/2)$ und $h: (2/8/4) + \mu(3/0/2).$

i) Eingabe der Geraden 1:

XEQ“C“ $“A_1 = “$
1 ENTER 1 ENTER 2 R/S $“U_1 = “$
1 ENTER 7 ENTER 2 R/S —

- ii) Eingabe der Ebene Geraden 2:
 XEQ"D" "A₂="
 2 ENTER 8 ENTER 4 R/S "U₂="
 3 ENTER 0 ENTER 2 R/S —
- iii) Berechnung der Schnittmenge zwischen der Geraden 1 und der Geraden 2:
 XEQ"H" "S: 2(1:4:2)"

Der Schnittpunkt der Geraden g und h ist also S(2/8/4).

6. Man bestimme den Abstand des Punktes P(1/4/1) von der Ebene E: (1/2/1) + λ(3/3/1) + μ(1/0/0), sowie den Fußpunkt des Lotes von P und E.

- i) Eingabe der Ebene E in Parameterform:
 XEQ"A" "A₁="
 1 ENTER 2 ENTER 1 R/S "U₁="
 3 ENTER 3 ENTER 1 R/S "V₁="
 1 ENTER 0 ENTER 0 R/S —
- ii) Eingabe des Punktes P:
 XEQ"E" "P(X↑Y↑Z)"
 1 ENTER 4 ENTER 1 R/S —
- iii) Berechnung des Abstandes des Punktes P von der Ebene 1, sowie Ermittlung des Lotfußpunktes:
 XEQ"I" "d=-0,632456"
 R/S "FDL: 1/5(5:19:8)"

Der Abstand des Punktes P von der Ebene E beträgt also 0,632456 Längeneinheiten. Das davorstehende Minuszeichen gibt an, daß der Punkt P und der Ursprung des Koordinatensystems auf verschiedenen Seiten der Ebene liegen. Der Punkt, in dem das Lot von P auf E die Ebene E schneidet, hat die Koordinaten (1/9/5).

7. Man bestimme den Abstand der windschiefen Geraden g: (0/0/1) + λ(1/0/0) und h: (1/0/2) + μ(0/1/0).

- i) Eingabe der 1. Geraden:
 XEQ"C" "A₁="
 0 ENTER 0 ENTER 1 R/S "U₁="
 1 ENTER 0 ENTER 0 R/S —
- ii) Eingabe der 2. Geraden:
 XEQ"D" "A₂="
 1 ENTER 0 ENTER 2 R/S "U₂="
 0 ENTER 1 ENTER 0 R/S —
- iii) Bestimmung der Schnittmenge zwischen der 1. und der 2. Geraden:
 XEQ"H" "WINDSCHIEF"
 R/S "ABSTAND: 1,00"

Der kürzeste Abstand der beiden windschiefen Geraden beträgt also genau eine (1) Längeneinheit.

8. Man bestimme den Fußpunkt des Lotes vom Punkt P(√2/2√2/3√2) auf die Ebene E: (0/0/0) + λ(1/0/0) + μ(0/1/0).

- XEQ"A" "A₁="
 0 ENTER 0 ENTER 0 R/S "U₁="
 1 ENTER 0 ENTER 0 R/S "V₁="
 0 ENTER 1 ENTER 0 R/S —
- XEQ"E" "P(X↑Y↑Z)"
 √2 ENTER 2√2 ENTER 3√2 R/S —
- XEQ"I" "d=4,242641"
 R/S "FDL: 1/1(1:2:0)" (TONE 0)
 R/S 1,414213562

Der Fußpunkt des Lotes ist also F(√2/2√2/0). Der Vorfaktor ist, wie nach dem zweiten R/S ersichtlich wird, √2.

Mathematische Grundlagen zum Programm:

1. EBENE – EBENE

a) Ist eine Ebene in der Parameterform gegeben, so erfolgt als erstes eine Umrechnung in die Normalenform (Koordinatendarstellung).

$$E_p: \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \lambda \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} + \mu \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

$$\rightarrow E_n: n_1x + n_2y + n_3z = n_4$$

$$\text{mit } \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}$$

$$\text{und } n_1a_1 + n_2a_2 + n_3a_3 = n_4$$

b) Die beiden Ebenen haben nun also die Form:

$$E_1: m_1x + m_2y + m_3z = m_4 \quad E_2: n_1x + n_2y + n_3z = n_4$$

Jetzt kann der Schnittwinkel zwischen beiden Ebenen berechnet werden:

Es gilt nämlich:

$$\cos\varphi = \frac{\begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}}{|m| \cdot |n|} ; \quad |m| = \sqrt{m_1^2 + m_2^2 + m_3^2}$$

$$|n| = \sqrt{n_1^2 + n_2^2 + n_3^2}$$

Ist nun dieser Schnittwinkel φ=0 oder φ=180°, so sind die beiden zu untersuchenden Ebenen entweder parallel oder identisch. Dies kann folgendermaßen entschieden werden:

Falls m₁m₄ = n₁n₄ und m₂m₄ = n₂n₄ und m₃m₄ = n₃n₄, so sind die beiden Ebenen identisch. Sollte dies jedoch nicht der Fall sein, so sind sie zueinander parallel und haben den Abstand

$$d = D_{E10} - \cos\varphi \cdot \text{sgn}(m_4) \cdot \text{sgn}(n_4) \cdot D_{E20}$$

$$\text{mit } D_{E10} = \frac{|m_4|}{|m|} \text{ und } D_{E20} = \frac{|n_4|}{|n|} \quad \text{voneinander.}$$

Nimmt der Schnittwinkel jedoch einen Wert an, der von 0 und 180 verschieden ist, so schneiden sich beide Ebenen in einer Schnittgeraden, die sich folgendermaßen bestimmen läßt:

$$\text{Richtungsvektor: } v = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \times \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}$$

$$\text{Antragungspunkt: } n_1x + n_2y + n_3z = n_4$$

$$m_1x + m_2y + m_3z = m_4$$

Aus diesem Gleichungssystem wird der Antragungspunkt der Geraden durch Nullsetzen einer Koordinate und anschließendem Auflösen nach x und y gewonnen.

2. EBENE – GERADE

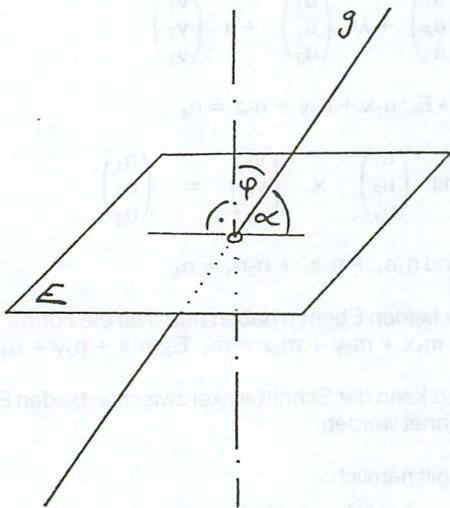
Die Ebene E und die Gerade g seien in folgender Form gegeben:

$$E: n_1x + n_2y + n_3z = n_4$$

$$g: \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} + \mu \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}$$

Es gilt:

$$\cos \varphi = \frac{\begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}}{|n| \cdot |u|} \quad \text{mit } |n| = \sqrt{n_1^2 + n_2^2 + n_3^2} \\ \text{und } |u| = \sqrt{u_1^2 + u_2^2 + u_3^2}$$



a) $\cos \varphi = 0$: In diesem Fall sind Gerade und Ebene entweder parallel oder haben unendlich viele Punkte gemeinsam. Ihr Abstand zueinander errechnet sich aus:

$$d = \frac{\text{sgn } n_4}{|n|} (n_1 e_1 + n_2 e_2 + n_3 e_3 - n_4)$$

Ist nun $d = 0$, so liegt die Gerade in der Ebene. Ist $d > 0$, so liegen Nullpunkt und Gerade auf verschiedenen Seiten der Ebene.

Ist $d < 0$, so liegen Nullpunkt und Gerade auf derselben Seite der Ebene.

b) $\cos \varphi \neq 0$: Die Gerade schneidet die Ebene in einem Punkt S unter dem Winkel α

Es gilt: $\alpha = \arcsin(\cos \varphi) = 90^\circ - \varphi$.

$$\text{und: } S(x_1/x_2/x_3) \text{ mit } \begin{aligned} x_1 &= e_1 + \lambda u_1 \\ x_2 &= e_2 + \lambda u_2 \\ x_3 &= e_3 + \lambda u_3 \end{aligned}$$

$$\text{wobei } \lambda = \frac{n_4 - n_1 e_1 - n_2 e_2 - n_3 e_3}{n_1 u_1 + n_2 u_2 + n_3 u_3}$$

3. EBENE – PUNKT

Es sei gegeben: $E: n_1 x_1 + n_2 y_2 + n_3 z_3 = n_4$; $P(p_1/p_2/p_3)$

Der Abstand des Punktes P von der Ebene E beträgt:

$$d = \frac{\text{sgn } n_4}{|n|} (n_1 p_1 + n_2 p_2 + n_3 p_3 - n_4)$$

Für $d > 0$ bzw. $d < 0$ gilt entsprechendes aus dem Kapitel EBENE – GERADE.

Der Fußpunkt des Lotes von P auf E hat die Koordinaten:

$$\vec{F} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} + \frac{d - \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}}{n_1^2 + n_2^2 + n_3^2} \cdot \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}$$

4. GERADE – GERADE

Es sei gegeben:

$$g: \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \lambda \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix}; \quad h: \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \mu \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

Dann gilt:

$$\cos \varphi = \frac{\begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}}{|u| \cdot |v|}$$

a) Ist nun $\cos \varphi = 0$, so sind beide Geraden entweder parallel oder identisch. Ihr Abstand voneinander errechnet sich aus:

$$d = \left| \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} b_1 - a_1 \\ b_2 - a_2 \\ b_3 - a_3 \end{pmatrix} \right| \cdot \frac{1}{\sqrt{u_1^2 + u_2^2 + u_3^2}}$$

Falls $d = 0$ gilt, so sind die Geraden identisch.

b) $\cos \varphi \neq 0$: Die Geraden schneiden sich in einem Punkt S unter dem Winkel φ oder aber sie sind windschief.

Der Schnittpunkt S bestimmt sich aus:

- (I) $u_1 - \mu v_1 = b_1 - a_1$ Aus diesem Gleichungssystem
- (II) $u_2 - \mu v_2 = b_2 - a_2$ wird μ berechnet und in die Gerade
- (III) $u_3 - \mu v_3 = b_3 - a_3$ dngleichung von h eingesetzt.

Der Abstand zweier windschiefer Geraden errechnet sich aus:

$$d = \frac{\left| \begin{pmatrix} a_1 - b_1 \\ a_2 - b_2 \\ a_3 - b_3 \end{pmatrix} \cdot \left[\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \times \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \right] \right|}{\left| \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \times \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \right|}$$

Ist dieser Abstand $d = 0$, so schneiden sich die beiden Geraden im besagten Punkte S.

5. GERADE – PUNKT

Der Abstand des Punktes P($p_1/p_2/p_3$) von der Geraden

$$g: \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \lambda \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

beträgt:

$$d = \frac{\left| \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} \times \begin{pmatrix} p_1 - a_1 \\ p_2 - a_2 \\ p_3 - a_3 \end{pmatrix} \right|}{\sqrt{v_1^2 + v_2^2 + v_3^2}}$$

Der Fußpunkt des Lotes von P auf g hat die Koordinaten:

$$\vec{X}_0 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \frac{\begin{pmatrix} p_1 - a_1 \\ p_2 - a_2 \\ p_3 - a_3 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}}{v_1^2 + v_2^2 + v_3^2} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

01+LBL "GEO"
SF 27 CF 29 RTH

05+LBL A
"A1=" CF 22 PROMPT
FC? 22 GTO 01 STO 02
RDN STO 01 RDN STO 00
"U1=" PROMPT STO 05
RDN STO 04 RDN STO 03
"V1=" PROMPT STO 00
RDN STO 07 RDN STO 06
5 8 XEQ 00 STO 09
RDN STO 10 RDN STO 11
RCL 02 * RCL 10
RCL 01 * + RCL 00
RCL 09 * + STO 12
RTH

50+LBL B
"A2=" CF 22 PROMPT
FC? 22 GTO 04 STO 15
RDN STO 14 RDN STO 13
"U2=" PROMPT STO 10
RDN STO 17 RDN STO 16
"V2=" PROMPT STO 21
RDN STO 20 RDN STO 19
10 21 XEQ 00 STO 22
RDN STO 23 RDN STO 24
RCL 15 * RCL 14
RCL 23 * + RCL 22
RCL 13 * + STO 25
RTH

95+LBL a
CF 22 "a1+a2+a3+a4"
PROMPT FC? 22 GTO 03
STO 12 RDN STO 11 RDN
STO 10 RDN STO 09 RTH

109+LBL b
CF 22 "b1+b2+b3+b4"
PROMPT FC? 22 GTO 07
STO 25 RDN STO 24 RDN
STO 23 RDN STO 22 RTH

123+LBL 07
FIX 4 CLA ARCL 22
"f X1" PROMPT CLA
ARCL 23 "f X2" PROMPT
CLA ARCL 24 "f X3"
PROMPT " =" ARCL 25
PROMPT RTH

141+LBL 00
X<>Y RCL IND X STO \
CLX SIGN - RCL IND X
STO J CLX SIGN -
RCL IND X STO a R↑ R↑
RCL IND X STO [CLX
SIGN - RCL IND X

STO ↑ CLX SIGN -
RCL IND X STO _ RCL]
RCL [* RCL \ RCL ↑
* - RCL \ RCL _ *
RCL a RCL [* -
STO \ RDN STO [RCL a
RCL ↑ * RCL] RCL _
* - RCL \ RCL [RTH

214+LBL 01
RCL 11 RCL 09 CHS +
CF 00 X=0? SF 00 X=0?
LASTX STO 04 RCL 10
STO 03 RCL 10 CHS
FS?C 00 CLX STO 05 5
11 XEQ 00 STO 06 RDN
STO 07 RDN STO 08 CLX
STO 00 STO 01 STO 02
RCL 09 X=0? GTO 08
RCL 10 X=0? GTO 09
RCL 12 RCL 11 /
STO 02 GTO 10

255+LBL 08
RCL 12 RCL 09 /
STO 00 GTO 10

261+LBL 09
RCL 12 RCL 10 /
STO 01

266+LBL 10
"E1: (" FIX 2 ARCL 00
"f:" ARCL 01 "f:"
ARCL 02 "f)" PROMPT
"+a(" ARCL 03 "f:"
ARCL 04 "f:" ARCL 05
"f)" PROMPT "+p(" ARCL 06
"f:" ARCL 07
"f:" ARCL 08 "f)"
PROMPT RTH

293+LBL 04
RCL 24 RCL 22 CHS +
CF 00 X=0? SF 00 X=0?
LASTX STO 17 RCL 23
STO 16 RCL 23 CHS
FS?C 00 CLX STO 18 18
24 XEQ 00 STO 19 RDN
STO 20 RDN STO 21 CLX
STO 13 STO 14 STO 15
RCL 22 X=0? GTO 08
RCL 23 X=0? GTO 09

RCL 25 RCL 24 /
STO 15 GTO 10

334+LBL 08
RCL 25 RCL 22 /
STO 13 GTO 10

340+LBL 09
RCL 25 RCL 23 /
STO 14

345+LBL 10
"E2: (" FIX 2 ARCL 13
"f:" ARCL 14 "f:"
ARCL 15 "f)" PROMPT
"+a(" ARCL 16 "f:"
ARCL 17 "f:" ARCL 18
"f)" PROMPT "+p(" ARCL 19
"f:" ARCL 20
"f:" ARCL 21 "f)"
PROMPT RTH

372+LBL C
"A1=" PROMPT STO 20
RDN STO 27 RDN STO 26
"U1=" PROMPT STO 31
RDN STO 30 RDN STO 29
RTH

380+LBL D
"A2=" PROMPT STO 34
RDN STO 33 RDN STO 32
"U2=" PROMPT STO 37
RDN STO 36 RDN STO 35
RTH

404+LBL E
"P(X↑Y↑Z)" PROMPT
STO 40 RDN STO 39 RDN
STO 38 RTH

413+LBL F
RCL 09 RCL 22 *
RCL 10 RCL 23 * +
RCL 11 RCL 24 * +
RCL 09 X↑2 RCL 10 X↑2
+ RCL 11 X↑2 + SORT
STO 41 / RCL 22 X↑2
RCL 23 X↑2 + RCL 24
X↑2 + SORT STO 42 /
FIX 8 RND ABS 1 X*Y?
GTO 05 LASTX RCL 12
SIGN * RCL 42 /
RCL 25 * CHS RCL 12
RCL 41 / ABS +
GTO 12

468+LBL 05
LASTX ACOS XEQ 06 11
24 XEQ 00 STO 41 RDN
STO 42 RDN STO 43 CLX
STO 44 STO 45 STO 46

RCL 09 RCL 23 *
RCL 22 RCL 10 * -
X=0? GTO 08 RCL 12
RCL 23 * RCL 25
RCL 10 * - X<>Y /
STO 44 LASTX RCL 09
RCL 25 * RCL 22
RCL 12 * - X<>Y /
STO 45 GTO 10

515+LBL 08
RCL 09 RCL 24 *
RCL 22 RCL 11 * -
X=0? GTO 09 RCL 12
RCL 24 * RCL 11
RCL 25 * - X<>Y /
STO 44 LASTX RCL 09
RCL 25 * RCL 22
RCL 12 * - X<>Y /
STO 46 GTO 10

547+LBL 09
RCL 12 RCL 24 *
RCL 25 RCL 11 * -
RCL 10 RCL 24 *
RCL 23 RCL 11 * - /
STO 45 LASTX RCL 10
RCL 25 * RCL 23
RCL 12 * - X<>Y /
STO 46

575+LBL 10
RCL 44 RCL 45 RCL 46
XEQ 15 "GS: " INT
LASTX X=Y? GTO 14 1/X
RND INT LASTX "f-1/"

590+LBL 14
FIX 0 RND X<>L
ARCL L X*Y? TONE 0
FS?C 01 FIX 2 "f(" ARCL 47
"f:" ARCL 49 "f)"
PROMPT FIX 9 X*Y? RTH
RCL 41 RCL 42 RCL 43
XEQ 15 FIX 0 FS?C 01
FIX 2 "+p(" ARCL 47
"f:" ARCL 48 "f:"
ARCL 49 "f)" PROMPT
RTH

626+LBL 06
"SCHNITTz=" FIX 1
ARCL X PROMPT RTH

632+LBL G
CF 00 RCL 09 RCL 29 *
RCL 10 RCL 30 * +
RCL 11 RCL 31 * +

```
RCL 09 X↑2 RCL 10 X↑2
+ RCL 11 X↑2 + SQRT
STO 41 / RCL 29 X↑2
RCL 30 X↑2 + RCL 31
X↑2 + SQRT / X=0?
GTO 05 RCL 09 RCL 26
* RCL 10 RCL 27 * +
RCL 11 RCL 28 * +
RCL 12 X=Y? GTO 11 -
LASTX SIGN * RCL 41
/ GTO 12
```

```
689+LBL 05
ASIN XEQ 06 RCL 12
RCL 09 RCL 26 * -
RCL 10 RCL 27 * -
RCL 11 RCL 28 * -
RCL 09 RCL 29 *
RCL 10 RCL 30 * +
RCL 11 RCL 31 * + /
ENTER↑ ENTER↑ ENTER↑
RCL 31 * RCL 28 +
X<> Z RCL 30 * RCL 27
+ X<>Y RCL 29 *
RCL 26 + GTO 17
```

```
735+LBL 11
"IDENTISCH" PROMPT RTN
```

```
739+LBL 13
"WINDSCHIEF" GTO 05
```

```
742+LBL 12
"PARALLEL"
```

```
744+LBL 05
PROMPT "ABSTAND: "
FIX 2 ARCL X PROMPT
RTN
```

```
751+LBL H
CF 00 31 37 XEQ 00
STO 41 RDW STO 42 RDW
STO 43 R↑ R-P RCL T
R-P X=0? GTO 05
RCL 34 RCL 28 -
STO 46 RCL 33 RCL 27
- STO 45 RCL 32
RCL 26 - STO 44 31
46 XEQ 00 R-P X↑2
RCL Z X↑2 + SQRT
RCL 29 X↑2 RCL 30 X↑2
+ RCL 31 X↑2 + SQRT
/ X=0? GTO 11 GTO 12
```

```
801+LBL 05
RCL 26 RCL 32 -
```

```
RCL 41 * RCL 27
RCL 33 - RCL 42 * +
RCL 28 RCL 34 -
RCL 43 * + X=0?
GTO 08 ABS RCL 41 X↑2
RCL 42 X↑2 + RCL 43
X↑2 + SQRT / GTO 13
```

```
833+LBL 08
RCL 32 RCL 26 -
STO 41 RCL 33 RCL 27
- STO 42 RCL 34
RCL 28 - STO 43
RCL 29 RCL 36 *
RCL 30 RCL 35 * -
X=0? GTO 08 RCL 41
RCL 36 * RCL 42
RCL 35 * - X<>Y /
GTO 10
```

```
865+LBL 08
RCL 30 RCL 37 *
RCL 31 RCL 36 * -
X=0? GTO 09 RCL 42
RCL 37 * RCL 43
RCL 36 * - X<>Y /
GTO 10
```

```
885+LBL 09
RCL 41 RCL 37 *
RCL 43 RCL 35 * -
RCL 29 RCL 37 *
RCL 31 RCL 35 * - /
```

```
901+LBL 10
ENTER↑ ENTER↑ ENTER↑
RCL 31 * RCL 28 +
X<> Z RCL 30 * RCL 27
+ X<>Y RCL 29 *
RCL 26 +
```

```
919+LBL 17
"S: " FS7C 00 "FDL: "
XEQ 15 INT LASTX X=Y?
GTO 14 "↑1/" 1/X
```

```
930+LBL 14
INT LASTX X=Y? TONE 0
FIX 0 ARCL X FS7C 01
FIX 2 "↑(" ARCL 49
"↑:" ARCL 48 "↑:"
ARCL 47 "↑)" PROMPT
FIX 9 X=Y? 1/X RTN
```

```
951+LBL 15
FIX 5 STO 49 RDW
```

```
STO 48 X<>Y STO 47
XEQ 16 RCL 48 RCL 49
XEQ 16 XEQ 16 FIX 3
RND LASTX CF 01 X<>Y
X=0? GTO 08 SIGN X<>Y
SF 01
```

```
973+LBL 08
X<>Y ST/ 47 ST/ 48
ST/ 49 RTN
```

```
979+LBL 16
X<> L
```

```
981+LBL 18
X<> L MOD LASTX X<>Y
RND X=0? GTO 18 +
RTN
```

```
991+LBL I
RCL 09 RCL 38 *
RCL 10 RCL 39 * +
RCL 11 RCL 40 * +
STO 41 RCL 12 - LASTX
SIGN * RCL 09 X↑2
RCL 10 X↑2 + RCL 11
X↑2 + STO 42 SQRT /
"d=" FIX 6 ARCL X
PROMPT RCL 12 RCL 41
- RCL 42 / ENTER↑
ENTER↑ ENTER↑ RCL 11
* RCL 40 + X<> Z
RCL 10 * RCL 39 +
X<>Y RCL 09 * RCL 38
+ SF 00 GTO 17
```

```
1048+LBL J
RCL 30 RCL 26 -
STO 41 RCL 39 RCL 27
- STO 42 RCL 40
RCL 28 - STO 43 31
43 XEQ 00 X↑2 X<>Y
X↑2 + X<>Y X↑2 +
SQRT RCL 29 X↑2
RCL 30 X↑2 + RCL 31
X↑2 + STO 44 SQRT /
"d=" FIX 6 ARCL X
PROMPT RCL 41 RCL 29
* RCL 42 RCL 30 * +
RCL 43 RCL 31 * +
RCL 44 / SF 00 GTO 17
END
```

Kurz gemeldet

Bug im HEPAX-Modul

Im HEPAX-Modul mit der Versionsnummer 1C (taucht im CAT 2 beim Namen eines Moduls auf) liefert die Funktion HPOFL fehlerhafte Werte.

Das Ergebnis ist immer MOD 16 des tatsächlichen Wertes, d.h. wenn sich der gesuchte String in Record 33 befindet, so ist der in den Stack zurückgegebene Wert 33 MOD 16 = 1.

Natürlich kann man hergehen und jeden Record mit der Nummer 16 + x (hier 1) nach dem gesuchten String absuchen, dies ist aber mit Sicherheit nicht im Sinne des Erfinders.

Kepler Kepler ???

Der kleine feine Unterschied war es mal wieder. Wir haben leider in der Ausgabe 4/89 von PRISMA dem altherwürdigen Herrn Kepler ein zweites p verpaßt, es war natürlich der Astronom Johannes Kepler gemeint.

Infrarotdrucker HP 82240 A

Wie mir kürzlich übermittelt wurde, benötigt der Infrarotdrucker spezielles Thermopapier.

Verwendet man das Thermopapier der alten HP41-Drucker, so sieht das Ergebnis eher blaß aus, anscheinend ist das Papier für den Infrarotdrucker etwas wärmesensitiver.

Der Handel ist über dieses Faktum vermutlich noch nicht ausreichend informiert, man sollte also nachfragen, ob das Thermopapier ausdrücklich für den Infrarotdrucker geeignet ist.

Michael Schilli
Daucherstraße 2
8900 Augsburg

mm

Berechnung von Himmelstrabanten

von Dr. G. Heilmann

HP41CX, 1443 Bytes, SIZE 112, X-Memory

Dies nun folgende Programm dient der Berechnung verschiedener Daten von in unserem Sonnensystem befindlichen Himmelskörpern.

Das Programm bestimmt

- a) → die **Orte** der Planeten **Merkur, Venus** und **Mars**,
→ den Ort der **Sonne** und den des **Erdmondes** in äquatorialen Koordinaten und für jeden Erdort in in horizontalen Koordinaten für jede Epoche
- b) die zugehörigen **Phasen** dieser Planeten und die Phase des Erdmondes
- c) die **Sternzeit** und den **Stundenwinkel** eines Objektes mit gegebenen Rektaszension für jede Epoche und jeden Erdort
- d) die horizontalen Koordinaten für jeden Erdort bei gegebener Epoche aus den Koordinaten eines Objektes für diese Epoche

Das Programm "IP" benötigt das Datenfile "IPD" im X-Memory mit 120 Byte.

Vorbereitung des Rechners

- a) Alle Programme und Tastenzuordnungen müssen gelöscht sein, das Programm benötigt den gesamten Hauptspeicher des HP41.
- b) Im X-Memory muß ein Datenfile mit dem Namen "IPD" und 120 Registern Größe existieren:

```
"IPD"           (Name)
120             (Register)
XEQ "CRFLD"    (File erzeugen)
```

- c) Konfiguration DEG, DMY, CLK24
- d) die Uhr in HP41 muß nach UT laufen, d.h. MEZ (Mitteleuropäische Zeit) + 1 Stunde

```
1
XEQ "T+X"
```

Bitte nur beim ersten Mal eingeben, zum zurückstellen wird dann das Gleiche mit -1 gemacht.

Eingabe des Datenfiles IPD

- a) XEQ "SIZE" 120
- b) IPD in den Hauptspeicher einlesen oder von Hand eingeben:
0,119 XEQ "WNDDTX" für das Einlesen von Datenbarcodes

- c) Daten aus den 120 Datenspeicherregistern in das X-Memory kopieren:

```
0,119 XEQ "SAVERX"
```

Eingabe des Programms IP

- a) XEQ "SIZE" 000.
- b) Das Programm IP einlesen oder von Hand eingeben. Bei Handeingabe muß man allerdings synthetische Befehle eingeben können, wozu dann z.B. CCD-ROM oder der Byte Grabber nötig sind.
- c) Programm packen und das END erzeugen:
GTO . .
- d) Beim Einlesen sollte ein kompiliertes END erzeugt werden, dafür benötigt man eventuell das CCD-ROM oder das ZENROM oder die Funktion COMPEND aus SKWIDBC.
Empfehlenswert bei der Handeingabe ist die Funktion COMPILE aus dem gleichen Modul.

Format der Eingaben

- a) B, L Geografische Breite ENTER 1
 Geografische Länge
 des Beobachtungsortes in Grad, Minuten und Sekunden GG,MMSS
 → südliche Breite und östliche Länge bitte **negativ** eingeben !
- b) UT? Universal Time in Stunden, Minuten und Sekunden SS,MMSS
- c) DATUM? Tag, Monat und Jahr = TT,MMJJJJ
- d) AER? Bezugs-Äruiunktium, nur das Jahr JJJJ

Bedeutung der angezeigten Flags

Flag	Bedeutung	Funktion
00	Planetenflag	Gesetzt , wenn noch kein Planet gewählt ist Es können trotzdem die Daten für Sonne und Mond berechnet werden.
01	Zeitflag	Gesetzt , wenn die geografischen Koordinaten eines von der Voreinstellung verschiedenen Ortes eingegeben werden (Voreinstellung: Frankfurt am Main) Gelöscht: Jetzt Gesetzt: Epoche wählen (D2,D3) → XEQ a toggelt den Flag

- 02 Präzessionsflag Es erfolgt die Abfrage "AER ?" gefragt und der Ort des Körpers auf dieses Äruinoktums bezogen.
Gesetzt: Aktuelles Äruinoktum
Gelöscht: Äruinoktum kann gewählt werden (D4)
 → XEQ c toggelt den Flag
- 03 Wird bei der Berechnung der Mondphase benutzt, um Zeit-PROMPTS zu umgehen.
- 04 Ausgabesteuer. **Gesetzt**, wenn Ausgabe des Ortes in horizontalen Koordinaten, dezimalen Grad
 Azimut (A) von Nord über Ost 0° bis 360°; Höhe (H) negativ, Objekt unter Horizont
- 04 bei "' A 'H" → **Gesetzt**, dann erfolgt die Ausgabe in äquatorialen Koordinaten (Rektaszension (RA) in Stunden, Minuten, Sekunden: hh.mmss; Deklination (D) in Grad, Minuten: ±dd.mm
- 04 bei "*RA *D" **Gelöscht**, dann die angegebene Form
 → XEQ e toggelt den Flag

Initialisierung

XEQ "IP" eintippen, es erscheint die Anzeige "FM/B,L" (Flag 01 gesetzt).
 Der Beobachtungsort wird festgehalten, wie er zuvor festgelegt worden war. Er kann erst wieder durch erneuten Start mit XEQ "IP" geändert werden.

- a) Nun die geographischen Koordinaten des Beobachtungsortes (B,L) eingeben, dann R/S. Der Flag 01 bleibt dabei gesetzt.
 Bei Erreichen des Menüs kann der Flag 01 durch XEQ a aus- und eingeschaltet werden.

Drückt man R/S, ohne eine Eingabe gemacht zu haben, so wird automatisch Frankfurt/Main gewählt, Flag 01 wird gelöscht.

- b) Es erscheint "Me,Ve,Ma→U:V:W" (Flag 00 gesetzt).
 Mit einer der Tasten U, V, W können die verschiedenen Planeten Merkur, Venus, Mars ausgewählt werden.
- Erfolgt der U, V, W Tastendruck innerhalb von 10s, so erfolgt die Anzeige "L P H a.aa", Flag 00 wird gelöscht.
 a.aa = {0.39, 0.72, 1.52} dies sind die mittleren Entfernungen der Planeten in astronomischen Einheiten (IAE = 149 597 892 km).

- Erfolgt innerhalb von 10s kein Tastendruck, so erscheint die Anzeige "L H → A", Flag 00 wird gesetzt.
 Mond und Sonnenorte können nun berechnet werden, die Planetendaten befinden sich nicht im Hauptspeicher:
 L (Luna): Mond, H (Helios): Sonne
 XEQ A führt dann wieder zu b) zurück.
- Wird irgendeine andere Taste gedrückt, so wird sofort zu b) zurückgekehrt.

Das Menü

Das Menü ist mit **XEQ J** erreichbar, es läßt den Stack unverändert.

Das Menü enthält 3 Gruppen sehr verschiedener Anzeigen:

- a) "L H → A", Flag 00 gesetzt,
- b) "L P H 0.00",
- c) "L P H a.aa".

Diese Tasten beziehen sich auf die Tasten der **zweiten** Reihe.

a.aa gibt die mittlere Entfernung des angewählten Planeten von der Sonne in astronomischen Einheiten an:

Merkur:	0.39
Venus:	0.72
Mars:	1.52

XEQ F liefert den **Mondort**, die **Mondentfernung** in km und die **Mondphase**.

XEQ G liefert den **Ort des gewählten Planeten**, seine **Entfernung** von der Erde (in astronomischen Einheiten) und seine **Phase** (0° bis 180° = vollbeleuchtet bis unbeleuchtet).

XEQ H liefert den Sonnenort und die Entfernung Erde-Sonne in astronomischen Einheiten.

"T↑ P↑ A↑"

Dieses Menü ist vom vorhergehenden durch Drücken der Taste R/S erreichbar.

↑ bedeutet, das damit die **geschifteteten** Tasten der ersten Reihe gemeint sind: a, c und e

- T↑: XEQ a: Zeit-Toggle (Flag 01)
- P↑: XEQ c: Präzessions-Toggle (Flag 02)
- A↑: XEQ e: äquatorial-horizontal-Toggle (Flag 04)

" <,T A"

Dieses Menü ist vom vorhergehenden durch Drücken der Taste R/S erreichbar und bezieht sich auf die Tasten B und D der ersten Reihe.

✧,T: **Sternzeit** und **Stundenwinkel** für Beobachtungsort und Epoche nach dem Status von Flag 01.

Sternzeit: XEQ B → a=?
Diesen PROMPT mit R/S übergehen, es erscheint

- "EP" Zeitprompts bei gesetztem Flag 01
- "STZ= hh:mm" bei gelöschtem Flag 01

Zur Wiederholung XEQ b: ... STZ = hh:mm

Stundenwinkel: XEQ B → a=?
Hier nur von 0 verschiedene Reaktanzension eingeben des Beobachtungsobjektes eingeben

- "EP" Zeitprompts bei gesetztem Flag 01
- "STW= hh:mm" bei gelöschtem Flag 01

Zur Wiederholung XEQ d: ... STW = hh:mm; soll der Stundenwinkel des vorher berechneten Ortes eines Objektes ausgegeben werden, dann noch einmal XEQ d aufrufen.

Der Mond

XEQ F

- "EP" Zeitprompts bei gesetztem Flag 01
- "L" Berechnung des geozentrisch-ekliptikalen Mondortes
- "E-A" Transformation auf geozentrisch-äquatoriale Koordinaten
- "G-T" Transformation auf topozentrisch-äquatoriale Koordinaten

Ist Flag 04 gelöscht, so erscheint die Ausgabe äquatorial in der Form " *RA *D".

Ist er gesetzt:

- "A-H" Transformation auf horizontale Koordinaten, die Ausgabe erfolgt in der Form "'A 'H".

R/S betätigen

es erscheint "R=aaaaaa", die Mondentfernung in km

R/S betätigen

- "PHASE" die Berechnung der Mondphase
 - "K:H" (Flag 03 gesetzt) es wird die Kepler-Routine für die Berechnung des Sonnenortes relativ zur Erde in der Ekliptik benutzt
 - "E-A" Transformation der ekliptischen Sonnenkoordinaten auf äquatoriale Koordinaten
- Es erfolgt die Anzeige "PHASE: aaa"

R/S betätigen,

es erscheint das Menü "L P H a.aa".

Die Planeten

Wechsel durch XEQ A dorthin, dann XEQ G

- "EP" Zeitprompts bei gesetztem Flag 01
- "K:P" Berechnung des heliozentrischen Planetenortes in der Ekliptik nach Kepler
- "K:H" Berechnung des heliozentrischen Erdortes in der Ekliptik nach Kepler
- "D" Berechnung des geozentrisch-ekliptikalen Planetenortes
- "E-A" Transformation des geozentrisch-ekliptikalen Planetenortes auf geozentrisch-äquatoriale Koordinaten

Ist Flag 04 gelöscht, so erscheint die Ausgabe äquatorial in der Form " *RA *D".

Ist er gesetzt:

- "A-H" Transformation auf horizontale Koordinaten, die Ausgabe erfolgt in der Form "'A 'H".

R/S betätigen

es erscheint "R=aaaaaa", der Abstand in AE

R/S betätigen

Es erfolgt die Anzeige "PHASE: aaa"

R/S betätigen,

es erscheint das Menü "L P H a.aa".

Die Sonne

XEQ H

- "EP" Zeitprompts bei gesetztem Flag 01
- "K:H" es wird die Kepler-Routine für die Berechnung des Sonnenortes relativ zur Erde
- "E-A" Transformation des geozentrisch-ekliptikalen Sonnenortes auf geozentrisch-äquatoriale Koordinaten

Ist Flag 04 gelöscht, so erscheint die Ausgabe äquatorial in der Form " *RA *D".

Ist er gesetzt:

- "A-H" Transformation auf horizontale Koordinaten, die Ausgabe erfolgt in der Form "'A 'H".

R/S betätigen

es erscheint "R=aaaaaa", der Abstand zur Sonne in AE

R/S betätigen

es erscheint das Menü "L P H a.aa".

Anmerkung: Die Aufrufe der einzelnen Unterprogramme durch z.B. XEQ a erfolgt durch

SERIE 40

Drücken der Taste a, wenn der USER-Modus eingeschaltet ist.

Literaturhinweise

- K. R. Lang Astrophysical Formulae, 2nd Edition
Springer Heidelberg, Berlin, New York 1980
- M. Schneider Himmelsmechanik
Bibliographisches Institut Mannheim, Wien, Zürich 1981
- O. Montenbruck Grundlagern der Ephemeridenrechnung
Verlag Sterne und Weltraum, Dr. Vehrenberg GmbH München 1984

01*LBL "IP"	45*LBL J	89 STO 14	151*LBL a	213 X<> 13
02 CLX	46 "L P H "	90 /	152 FC?C 01	214 +
03 X<>F	47 FIX 2	91 XEQ 03	153 SF 01	215 RCL 52
04 SF 01	48 ARCL 16	92 X<>Y	154 GTO J	216 X<>Y
05 CF 22	49 FIX 4	93 RCL 40	155*LBL c	217 X<0?
06 SF 27	50 FS? 00	94 /	156 FC?C 02	218 +
07 CF 29	51 " L H P-)A"	95 RCL 07	157 SF 02	219 STO 12
08 "FM/B,L"	52 PROMPT	96 +	158 GTO J	220 RCL 48
09 PROMPT	53 "T^ P^ A^"	97 RCL 14	159*LBL e	221*LBL 00
10 FC?C 22	54 PROMPT	98 /	160 FC?C 04	222 "K:H"
11 CF 01	55 " K,T A"	99 STO 07	161 SF 04	223 AVIEW
12 112	56 PROMPT	100 FC? 02	162 GTO J	224 RCL 49
13 PSIZE	57 GTO J	101 RTN	163*LBL H	225 +
14 RDN	58*LBL 61	102 RCL 37	164 SF 07	226 XEQ 21
15 "IPD"	59 CLX	103 "AEQ?"	165 GTO 00	227 RCL 08
16 36	60 GTO 00	104 PROMPT	166*LBL G	228 STO 06
17 SEEKPTA	61*LBL 62	105 FC?C 22	167 CF 07	229 RCL 35
18 28.111	62 RCL 49	106 RCL 41	168*LBL 00	230 RCL 07
19 GETRX	63 GTO 00	107 DDAYS	169 CF 05	231 *
20 R^	64*LBL 63	108 RCL 06	170 CLX	232 RCL 34
21 R^	65 RCL 40	109 -	171 STO 10	233 +
22 FS? 01	66*LBL 00	110 RCL 14	172 FC? 03	234 ST+ 09
23 GTO 00	67 CF 00	111 /	173 XEQ 16	235 "D"
24 RCL 58	68 "IPLD"	112 STO 14	174 FIX 8	236 FC? 07
25 RCL 59	69 SEEKPTA	113 RTN	175 RCL 48	237 AVIEW
26*LBL 00	70 RCL 36	114*LBL 01	176 FS? 07	238 FC? 07
27 HR	71 GETRX	115 DATE	177 GTO 00	239 XEQ 18
28 RCL 53	72 GTO J	116 DDAYS	178 "K:P"	240 XEQ 19
29 /	73*LBL 16	117 TIME	179 AVIEW	241 FS?C 03
30 STO 59	74 "EP"	118 RTN	180 XEQ 21	242 RTN
31 X<>Y	75 AVIEW	119*LBL 02	181 RCL 08	243 FS? 02
32 HR	76 CF 22	120*LBL 10	182 STO 11	244 XEQ 20
33 STO 58	77 RCL 37	121 "DATUM?"	183 RCL 48	245*LBL 17
34 SF 00	78 FC? 01	122 PROMPT	184 RCL 57	246 RCL 09
35*LBL A	79 XEQ 01	123 FC?C 22	185 +	247 RCL 52
36*LBL 06	80 FS? 01	124 GTO 10	186 REGMOVE	248 MOD
37 SF 25	81 XEQ 02	125 DDAYS	187 RCL 07	249 RCL 53
38 "Me, Ve, Ma-)U:V:W"	82 HR	126 "UT?"	188 ST* 01	250 /
39 AVIEW	83 X<>Y	127 PROMPT	189 ST* 03	251 STO 09
40 GETKEY	84 STO 06	128 FC?C 22	190 ST* 05	252 FS? 04
41 GTO IND X	85 RCL 38	129 .	191 RCL 05	253 GTO 00
42 FC? 25	86 +	130 RTN	192 RCL 04	254 HMS
43 GTO 06	87 STO 07	131*LBL 03	193 +	255 FIX 4
44*LBL 00	88 RCL 39	132 ENTER^	194 ST+ 09	256 "*"
		133 ENTER^	195 RCL 01	257 ARCL X
		134 RCL 42	196 ST+ 00	258 "}"
		135 *	197 RCL 02	259 FIX 2
		136 RCL 43	198 RCL 03	260 RCL 10
		137 +	199 +	261 HMS
		138 *	200 STO 13	262 ARCL X
		139 RCL 44	201 RCL 09	263 AVIEW
		140 +	202 RCL 28	264*LBL 00
		141 RCL 45	203 P-R	265 FS? 04
		142 /	204 RCL 00	266 XEQ 23
		143 X<>Y	205 RCL Z	267 RTN
		144 RCL 46	206 P-R	268 FIX 4
		145 *	207 RCL Z	269 RCL 57
		146 +	208 R-P	270 FS? 05
		147 RCL 40	209 RCL Z	271 RCL 11
		148 MOD	210 X<>Y	272 FC? 05
		149 STO 15	211 R-P	273 RCL 08
		150 RTN	212 RDN	274 X>Y?

275 FIX 0	337 R-P	399 RCL 09	461 FS? 10	523 -	585 GTO 00
276 "R="	338 STO 08	400 +	462 -	524 *	586*LBL D
277 ARCL X	339 FS? 05	401 STO 00	463 D-R	525 RCL M	587 "a=?,d=?"
278 AVIEW	340 RTN	402 RCL 14	464 RAD	526 +	588 PROMPT
279 RTN	341 RDN	403 RCL 55	465 STO 01	527 STO M	589 FS?C 22
280 FS? 07	342 STO 10	404 RCL 02	466 RCL 02	528 XEQ 04	590 XEQ 09
281 GTO J	343 X<>Y	405 Z E4	467 +	529 RND	591*LBL d
282 CLA	344 STO 09	406 /	468 PI	530 X=0?	592 SF 08
283 ARCL 47	345 RTN	407 -	469 X>Y?	531 RTN	593 XEQ 07
284 AVIEW	346*LBL 19	408 *	470 X<>Y	532 X<> L	594*LBL 00
285 FC? 05	347 "E-A"	409 STO 02	471 RCL 01	533 GTO 05	595 "A-H"
286 GTO 00	348 AVIEW	410 RCL 10	472 XEQ 04	534*LBL 04	596 AVIEW
287 SF 03	349 RCL 56	411 RCL 28	473 RCL 52	535 ENTER^	597 RCL 53
288 XEQ H	350 FS? 07	412 P-R	474 RCL M	536 SIN	598 *
289 RCL 88	351 ST+ 09	413 RCL 00	475 R-D	537 RCL 02	599 CHS
290 ST/ 11	352 RCL 50	414 X<>Y	476 DEG	538 *	600 RCL 10
291 XEQ 18	353 RCL 07	415 P-R	477 FS?C 10	539 -	601 RCL 46
292*LBL 00	354 RCL 28	416 X<>Y	478 -	540 RCL 01	602 INT
293 RCL 08	355 -	417 X<> 02	479 STO 03	541 -	603 P-R
294 X^2	356 RCL 51	418 STO T	480 RCL 38	542 RTN	604 RCL Z
295 RCL 11	357 *	419 X<>Y	481 *	543*LBL B	605 X<>Y
296 X^2	358 -	420 P-R	482 TAN	544 CLX	606 P-R
297 +	359 X<> 09	421 R^	483 RCL 28	545 "a=?"	607 RCL 58
298 RCL 06	360 RCL 10	422 R^	484 RCL 02	546 PROMPT	608 X<>Y
299 X^2	361 RCL 28	423 P-R	485 +	547 HR	609 P-R
300 -	362 P-R	424 ST+ T	486 RCL 28	548 STO 09	610 STO M
301 RCL 08	363 RCL Z	425 CLX	487 LASTX	549*LBL b	611 CLX
302 RCL 11	364 X<>Y	426 +	488 -	550 RCL 09	612 RCL 58
303 *	365 P-R	427 -	489 /	551 X=0?	613 R^
304 ST+ X	366 X<> 09	428 RCL 02	490 SQRT	552 SF 09	614 P-R
305 /	367 STO T	429 X<>Y	491 *	553*LBL 07	615 X<>Y
306 ACOS	368 X<>Y	430 R-P	492 ATAN	554 XEQ 16	616 ST+ M
307 FIX 0	369 P-R	431 X<>Y	493 ENTER^	555 XEQ 22	617 RDN
308 CLA	370 R^	432 RCL 01	494 +	556 FS?C 08	618 X<>Y
309 ARCL 47	371 R^	433 +	495 RCL 52	557 RTN	619 -
310 "}: "	372 P-R	434 STO 09	496 MOD	558 "STW= "	620 R-P
311 ARCL X	373 ST+ T	435 RDN	497 STO 09	559 FS?C 09	621 RCL M
312 AVIEW	374 RDN	436 R-P	498 RCL 28	560 "STZ= "	622 X<>Y
313 RTN	375 -	437 X<>Y	499 RCL 02	561 HMS	623 R-P
314 GTO J	376 RCL 09	438 STO 10	500 RCL 03	562 ATIME	624 RDN
315*LBL 18	377 R-P	439 RTN	501 COS	563 AVIEW	625 X<>Y
316 RCL 13	378 X<>Y	440*LBL 21	502 *	564 RTN	626 RCL 52
317 RCL 11	379 STO 09	441 REGMOVE	503 -	565 GTO J	627 MOD
318 P-R	380 RDN	442 RCL 07	504 RCL 00	566 RTN	628 FIX 2
319 RCL 12	381 R-P	443 ST* 02	505 *	567*LBL 22	629 ""
320 X<>Y	382 RDN	444 ST* 04	506 STO 08	568 RCL 15	630 ARCL X
321 P-R	383 STO 10	445 ST* 05	507 RTN	569 RCL 59	631 "}'"
322 X<> 10	384 RTN	446 RCL 01	508*LBL 04	570 -	632 ARCL Y
323 RCL 08	385*LBL 20	447 ST+ 02	509 STO M	571 RCL 09	633 AVIEW
324 P-R	386 "PZ"	448 RCL 03	510 XEQ 04	572 -	634 RTN
325 X<>Y	387 AVIEW	449 RCL 04	511 STO N	573 RCL 40	635 GTO J
326 ST- T	388 RCL 14	450 RCL 05	512 X<>Y	574 MOD	636*LBL F
327 X<> 09	389 RCL 54	451 +	513 STO 05	575 RTN	637 CF 05
328 X<>Y	390 RCL 07	452 +	514 XEQ 04	576*LBL 09	638 CF 07
329 P-R	391 RCL 28	453 RCL 52	515 RCL N	577 HR	639 XEQ 16
330 ST- 10	392 -	454 MOD	516*LBL 05	578 STO 10	640 "L"
331 RDN	393 STO 02	455 LASTX	517 RCL Y	579 X<>Y	641 AVIEW
332 -	394 4 E4	456 X<>Y	518 RCL Y	580 HR	642 RCL 51
333 RCL 10	395 /	457 RCL 56	519 -	581 STO 09	643 STO 05
334 R-P	396 +	458 X<=Y?	520 /	582 RTN	644 RCL 90
335 RCL Z	397 *	459 SF 10	521 RCL M	583*LBL 23	645 STO 06
336 X<>Y	398 STO 01	460 RDN	522 RCL 05	584 XEQ 22	646 RCL 07

SERIE 40

647 RCL 91	709 RCL 01	771 RCL 04	833 +	895 3
648 *	710 3	772 -	834 RCL 10	896 *
649 ENTER^	711 *	773 SIN	835 RCL 01	897 +
650 INT	712 SIN	774 RCL 70	836 -	898 RCL 45
651 STO M	713 RCL 62	775 *	837 SIN	899 /
652 -	714 *	776 +	838 RCL 80	900 SIN
653 RCL 91	715 +	777 RCL 11	839 *	901 RCL 89
654 ST/ M	716 RCL 00	778 RCL 04	840 +	902 X<>Y
655 *	717 RCL 03	779 -	841 RCL 10	903 /
656 ENTER^	718 -	780 SIN	842 RCL 04	904 STO 08
657 INT	719 STO 06	781 RCL 71	843 +	905 XEQ 19
658 STO N	720 SIN	782 *	844 SIN	906 "G-T"
659 -	721 RCL 63	783 +	845 RCL 81	907 AVIEW
660 RCL 91	722 *	784 RCL 01	846 *	908 RCL 08
661 X^2	723 +	785 RCL 04	847 +	909 STO 11
662 ST/ N	724 RCL 06	786 +	848 RCL 10	910 RCL 09
663 /	725 ST+ X	787 SIN	849 RCL 04	911 STO 12
664 STO 0	726 STO 06	788 RCL 72	850 -	912 RCL 10
665*LBL 11	727 SIN	789 *	851 SIN	913 STO 13
666 RCL IND 06	728 RCL 64	790 +	852 RCL 82	914 RCL 89
667 RCL M	729 *	791 RCL 01	853 *	915 STO 08
668 *	730 +	792 RCL 04	854 +	916 RCL 15
669 RCL 52	731 RCL 04	793 -	855 RCL 05	917 RCL 59
670 MOD	732 SIN	794 SIN	856 RCL 01	918 -
671 RCL IND 06	733 STO 08	795 RCL 73	857 -	919 RCL 53
672 RCL N	734 RCL 65	796 *	858 SIN	920 *
673 *	735 *	797 +	859 RCL 76	921 STO 09
674 +	736 +	798 RCL 45	860 *	922 RCL 58
675 RCL IND 06	737 RCL 00	799 /	861 +	923 STO 10
676 RCL 0	738 RCL 02	800 RCL 00	862 RCL 05	924 XEQ 18
677 *	739 -	801 +	863 RCL 13	925 SF 05
678 +	740 STO 05	802 STO 09	864 -	926 GT0 17
679 ISG 06	741 ST+ X	803 RCL 08	865 SIN	927 END
680 RCL IND 06	742 SIN	804 RCL 74	866 RCL 77	
681 +	743 RCL 66	805 *	867 *	
682 ISG 06	744 *	806 RCL 10	868 -	R00= 0.387099000
683 RCL IND 06	745 STO 10	807 +	869 RCL 45	R01= 0.205614000
684 RCL 07	746 -	808 RCL 45	870 /	R02= 0.000020000
685 *	747 RCL 06	809 /	871 STO 10	R03= 102.2794000
686 ISG 06	748 RCL 13	810 +	872 RCL 84	R04= 72.515300000
687 RCL IND 06	749 -	811 RCL 02	873 RCL 01	R05= 149400.00000
688 +	750 SIN	812 -	874 COS	R06= 7.002900000
689 RCL 07	751 RCL 67	813 SIN	875 RCL 85	R07= 0.001900000
690 *	752 *	814 RCL 75	876 *	R08= 47.146000000
691 +	753 +	815 *	877 +	R09= 1.185000000
692 RCL 52	754 RCL 06	816 RCL 03	878 RCL 13	R10= 28.753700000
693 MOD	755 RCL 01	817 ST+ X	879 COS	R11= 0.370500000
694 STO IND 05	756 -	818 RCL 00	880 RCL 83	R12= 0.723332000
695 ISG 05	757 STO 11	819 -	881 *	R13= 0.006821000
696 ISG 06	758 SIN	820 RCL 02	882 +	R14= -0.000048000
697 GT0 11	759 RCL 68	821 -	883 RCL 11	R15= 212.6032000
698 RCL 01	760 *	822 STO 10	884 COS	R16= 197.8039000
699 SIN	761 +	823 SIN	885 RCL 86	R17= 58320.00000
700 RCL 60	762 RCL 06	824 RCL 78	886 *	R18= 3.393600000
701 *	763 RCL 01	825 *	887 +	R19= 0.001000000
702 RCL 01	764 +	826 +	888 RCL 06	R20= 75.780000000
703 ST+ X	765 STO 12	827 RCL 10	889 COS	R21= 0.900000000
704 STO 13	766 SIN	828 RCL 01	890 RCL 87	R22= 54.383800000
705 SIN	767 RCL 69	829 +	891 *	R23= 0.508000000
706 RCL 61	768 *	830 SIN	892 +	R24= 1.523692000
707 *	769 +	831 RCL 79	893 RCL 12	R25= 0.093313000
708 +	770 RCL 06	832 *	894 COS	R26= 0.000092000

R27= 319.5294000	R51= 8640184.500	R75= 212.0000000	R99= 1000.000000
R28= 59.85850000	R52= 23925.83600	R76= 4586.000000	R100= 481267.0000
R29= 19080.00000	R53= 3600.000000	R77= 192.0000000	R101= 270.4341640
R30= 1.850300000	R54= 1.002740000	R78= 165.0000000	R102= -0.001133000
R31= -0.000700000	R55= "PHASE"	R79= 206.0000000	R103= 0.883142000
R32= 48.78600000	R56= 16.00000600	R80= -110.0000000	R104= 477198.0000
R33= 0.771000000	R57= 12.00000000	R81= 148.0000000	R105= 296.1046080
R34= 285.4322000	R58= 23.43930000	R82= 541.0000000	R106= 0.009192000
R35= 1.069800000	R59= 0.013000000	R83= 18520.00000	R107= 0.849108000
R36= 1.000000000	R60= 360.0000000	R84= 21.00000000	R108= -1934.000000
R37= 0.016751000	R61= 15.00000000	R85= 25.00000000	R109= 259.1832750
R38= -0.000042000	R62= 0.640600000	R86= -526.0000000	R110= 0.002078000
R39= 358.4758000	R63= 0.556800000	R87= 44.00000000	R111= -0.142008000
R40= 359.0498000	R64= 180.0000000	R88= -31.00000000	R112= 36000.00000
R41= 35640.00000	R65= 6.000000000	R89= -23.00000000	R113= 279.6966780
R42= 101.2208000	R66= 50.06000000	R90= 11.00000000	R114= 0.000303000
R43= 1.719200000	R67= -8.420000000	R91= 10.00000000	R115= 0.768925000
R44= 16.02700000	R68= 22640.00000	R92= 3423.000000	R116= 35999.00000
R45= 1.011900000	R69= 769.0000000	R93= 187.0000000	R117= 358.4758330
R46= 0.500000000	R70= 36.00000000	R94= 34.00000000	R118= -0.000150000
R47= 36525.00000	R71= -125.0000000	R95= 28.00000000	R119= 0.049750000
R48= 24.00000000	R72= 2370.000000	R96= 149597892.0	
R49= 1.011950000	R73= -668.0000000	R97= 6378.140000	
R50= 0.092900000	R74= 412.0000000	R98= 92.11100000	

Dr. G. Heilmann
Obernhofstraße 15
5408 Seelbach

Grenzwertberechnungen

32 Zeilen, 59 Bytes, SIZE 004 +
der im Unterprogramm verwendeten Register, HP-41C

Dieses Programm erleichtert die Bestimmung des Grenzwertes einer vorgegebenen Funktion gegen einen Wert X_L .

Vor dem Aufrufen des Programmes wird die zu untersuchende Funktion unter einem globalem Label in den Programmspeicher eingegeben und anschließend ein GTO... ausgeführt. Sodann wird das eigentliche Grenzwertprogramm mit XEQ"LIM" aufgerufen und bei der Abfrage "FUNCTION NAME ?" das Label der vorher eingegebenen Funktion eingetippt. Nach dem Abschluß dieser Eingabe mit R/S erscheint "XL=" in der Anzeige, worauf der Grenzwert, gegen den die Funktion zu untersuchen ist, eingegeben wird. Als letztes wird nun noch der Startwert verlangt ("X0="), von dem aus das Programm gegen den Grenzwert X_L streben soll. Wird auch diese Eingabe mit R/S abgeschlossen, so erscheinen nun die Werte, die die Funktion annimmt, wenn X von X0 gegen X_L läuft. Ist ein konstanter Grenzwert erreicht, so hält das Programm an.

Beispiel:

$$\text{Es soll } \lim_{x \rightarrow 0^+} \frac{\sin x}{x}$$

berechnet werden.

Als erstes wird die Funktion unter einem Label, das wir in diesem Fall LBL"FU" nennen wollen, im Programmspeicher abgelegt:

```
PRGM
GTO ..
LBL"FU" SIN LASTX / RTN
GTO
PRGM
XEQ"RAD"
```

Anschließend wird "LIM" aufgerufen:

```
XEQ"LIM" "FUNCTION NAME"
"FU" R/S "XL="
0 R/S "X0="
1 R/S "0,958851077"
"0,989615837"
"0,997397867"
"0,999349085"
:
"0,999999960"
"0,999999990"
"0,999999998"
"0,999999999"
"1,000000000"
"1,000000000"
```

An dieser Stelle hält das Programm dann an.

```
01 LBL"LIM"
02 SF 00
03 "NAME ?"
04 AON
05 PROMPT
06 ASTO 00
07 AOFF
08 "XL="
09 PROMPT
10 STO 01
11 "X0="
12 PROMPT
13 STO 02
14 LBL 00
15 RCL 01
16 RCL 02
17 +
18 2
19 /
20 STO 02
21 XEQ IND 00
22 VIEW X
23 RCL 03
24 X<>Y
25 FS?C 00
26 GTO 05
27 X=Y?
28 STOP
29 LBL 05
30 STO 03
31 GTO 00
32 END
```

Michael Schilli

GEW + POP

Gewichtsberechnung einfacher Körper und Unterdrücken des letzten RTN

von Günter Schapka

SIZE 009, HP-41CX, CCD, Rambox

Mit dem Programm GEW lassen sich kubische und zylindrische Körper berechnen, z.B. Bleche, Stäbe und Rohre.

Variablen sind: Spezifisches Gewicht, Breite und Höhe (bzw. Außen- und innendurchmesser), Länge und Gewicht. Gleichgültig, welche der vier Variablen eingegeben werden, die letzte, 5. wird berechnet.

Da das Programm nicht andauernd in Benutzung ist, wurde es über AVIEW und GETKEYX gesteuert (Menüanzeige) und nicht über z.B. lokale Alpha-Labels. Somit ist es selbsterklärend und bestehende Tastenzuordnungen bleiben erhalten.

Für das spezifische Gewicht (hier mit μ angezeigt) sind bereits drei Werte (für Stahl, Blei und Aluminium) im Programm eingebaut, die Auswahl erfolgt bei LBL 11 durch drücken der zugehörigen Buchstabentaste. Bei Taste M kann ein Wert manuell eingegeben werden, bei Taste R wird das spezifische Gewicht berechnet (Dimension in kg/Liter).

Das Hauptmenü bei LBL 16 fragt die Variablen ab. Die möglichen Werte stehen über den Tasten in der oberen Reihe. Drückt man zuerst μ , dann kann, wie bereits vorher beschrieben, ausgewählt werden (hier berechnen zu wollen ist natürlich Unsinn!). Der gewählte oder eingegebene Wert wird als μ mit Index S, A, B oder M angezeigt. Pfeil oder Doppelpunkt dahinter fordern bereits zur Eingabe des nächsten Zahlenwertes auf. Nach Drücken von Taste R/S kann der Wert bei Anzeige des Hauptmenüs seiner Variablen zugeordnet werden: Breite [mm] auf Taste B, Höhe [mm] auf D, oder Gewicht [kg] auf E. Wird als erste Taste eine andere Taste als μ gedrückt, erscheint in der Anzeige vorläufig die Variable mit dem Wert 0.

Nach Eingabe von B oder H ist Flag 1 gesetzt, bei D oder d gelöscht, damit wird jeweils die richtige Gleichung angesteuert.

Für die fest eingegebenen μ -Werte und die Konstante $0.25 \cdot \pi \cdot E \cdot 6$ ist in den Zeilen 42 und 43 das Alpha-Register komplett belegt (Reg. M, N, O und P). Wird zwischen den Zeilen 42 und 44 im SST-Modus durchgetastet, dann gehen die ersten 4 Bytes von P (25-28 des Alpha-Registers) verloren und die Konstante ist verkehrt.

LBL 05 (Zeile 60) enthält die Gleichungen zur Berechnung des spezifischen Gewichts:

$$\mu = \frac{4 \cdot G \cdot E \cdot 6}{\pi \cdot l \cdot (D^2 - d^2)} \quad (F01=0) \quad \text{oder} \quad \frac{4 \cdot G \cdot E \cdot 6 \cdot \pi}{\pi \cdot l \cdot B \cdot H \cdot 4} \quad (F01=1) \quad R01 = \mu$$

LBL 06 (Zeile 162) enthält die Gleichungen zur Berechnung von D oder B:

$$D = \sqrt{\frac{G}{K \cdot \mu \cdot L} + d^2} \quad (F01=0) \quad \text{oder} \quad B = \frac{G \cdot \pi}{K \cdot \mu \cdot L \cdot 4 \cdot H} \quad (F01=1) \quad \begin{matrix} R02 = D^2 \\ \text{oder} \\ R02 = B \end{matrix}$$

LBL 07 (Zeile 174) enthält die Gleichungen zur Berechnung von d oder H:

$$d = \sqrt{\frac{G}{K \cdot \mu \cdot L} + D^2} \quad (F01=0) \quad \text{oder} \quad H = \frac{G \cdot \pi}{K \cdot \mu \cdot L \cdot 4 \cdot B} \quad (F01=1) \quad \begin{matrix} R03 = d^2 \\ \text{oder} \\ R03 = H \end{matrix}$$

LBL 08 (Zeile 201) enthält die Gleichungen zur Berechnung von L:

$$L = \frac{G}{K \cdot \mu \cdot (D^2 - d^2)} \quad (F01=0) \quad \text{oder} \quad L = \frac{G}{K \cdot \mu \cdot B \cdot H} \quad (F01=1) \quad R04 = L$$

LBL 09 (Zeile 211) enthält die Gleichungen zur Berechnung von G:

$$G = K \cdot \mu \cdot L \cdot (D^2 - d^2) \quad (F01=0) \quad \text{oder} \quad G = \frac{K \cdot \mu \cdot L \cdot H \cdot B \cdot \pi}{4} \quad R05 = G$$

Weitere Register sind wie folgt belegt:

$$R06 = \frac{G}{K}; \quad R07 = \frac{4}{\pi}; \quad R08 = \frac{\pi}{4 \cdot E \cdot 6}$$

Nach Eingabe aller Werte und Berechnung des Ergebnisses können einzelne Variablen verändert und ihre Auswirkung auf das Ergebnis untersucht werden. Ein Zwischenergebnis kann vom Hauptmenü nach Drücken der STO-Taste (LBL 33) zur späteren Weiterverwendung in R00 gespeichert werden.

Erfolgt innerhalb 99 Sekunden keine Eingabe ins Hauptmenü, schaltet der Rechner mit gesetzter Flag 11 ab. Beim Einschalten erscheint wieder das Hauptmenü.

Programm-Ausstieg mit R/S aus dem Hauptmenü. Der Rechner schaltet ab, bleibt aber am Programmende stehen und kann von da aus über erneutes R/S wieder von vorn beginnen.

Noch einige Programmdetails:

Solange noch kein Gewichtswert bekannt ist (entweder durch Eingabe oder Rechnung) bleibt bei LBL 10 auch Register 06 Null. Wird aus dem Hauptmenü eine Taste der oberen Reihe ohne Dateneingabe gedrückt, dann veranlaßt dies normalerweise Berechnung des zugehörigen Wertes. Im Unterprogramm 19 wird in diesem Fall die RTN-Adresse (XEQ POP) gelöscht, direkt zum Aufruf der Berechnung zurückverzweigt und für diesen Wert

Null gespeichert (Soll z.B. für Rundstabmaterial anstelle von Rohr gerechnet werden, dann muß für d Null gespeichert werden!).

Die GETKEYX-Ansteuerung im Hauptmenü wurde bereits von Michael Kamp im Programm RAMED (PRISMA 87.7.24) veröffentlicht.

Das Unterprogramm mit POP wurde im PPC-Journal V12 N5 P47 von Robert Swanson veröffentlicht, siehe auch gesonderten Artikel mit einer MCODE-Version.

POP

Das Programm POP unterdrückt den letzten RTN-Befehl im Register b, alle anderen RTN werden im RTN-Stack um eine Ebene heruntergeschoben.

Die synthetische Version wurde bereits im PPC-Journal V12 N5 P47 von Robert Swanson veröffentlicht, sie benötigt eine Unterprogrammebene.

Als ersten MCODE-Gehversuch habe ich eine MCODE-Version geschrieben, die keine Unterprogrammebene belegt.

Ein beigefügtes Testprogramm TST arbeitet erst normal (Flag 01=1, F00=0),

dann unterdrückt es hintereinander 3 Rücksprünge mit der synthetischen Version (F01=0, F00=0), dann nocheinmal mit der MCODE-Version (F01=0, F00=1).

POP-Beschreibung (Zeilenweise)

- 515-517 Funktionsname
- 518 Aktiver Pointer auf Nybble 3 gesetzt
- 51A Statusregister a von CPU-Register C nach A kopiert
- 51B Einlesen Statusregister b
- 51C Programmzeiger aus C und RTN-Adresse 3/4 (jeweils halb) aus A werden miteinander vertauscht (Nybbles 0 bis 3)
- 51D CPU-Register C wird um 4 Nybbles nach rechts rotiert - die Adressen stehen in der richtigen Reihenfolge
- 51E Programmzeiger wird aus CPU-Register A zurückgeholt
- 51F Statusregister b wird zurückgeschrieben, enthält jetzt die Rücksprünge 2, 3 und 4 (zur Hälfte)
- 520 CPU-Register A nach C zurückholen
- 521 C wird bis inclusive Nybble 3 Null gesetzt (RTN Adresse 1)
- 522 CPU-Register C wird um 4 Nybbles nach rechts rotiert
- 523 Statusregister a wird aus CPU-Register C zurückgeschrieben, enthält jetzt RTN 4/2, 5, 6 und an letzter Stelle 0
- 524 Funktionsende

```

18:54 31.07.89 PRP "GEW"
E515 090 P 17:15 11.08
E516 00F 0
E517 010 P 01*LBL "GEW"
SIZE? 9 X?Y? PSIZE
18:55 31.07.89 CLRG
E518 R= 3 07*LBL 20
E519 READ 11(a) 1.008 CLRGX 4 PI /
E51A A=C ALL STO 07 CF 22 . X<F
E51B READ 12(b) "KG/LTR MM:KG" AVIEW
E51C A<>C R<- PSE
E51D RCR 4 20*LBL 16
E51E A<>C R<- CF 21 "p B:H D:d L G"
E51F WRIT 12(b) AVIEW 15 GETKEYX RDN
E520 A<>C ALL X<0? -5 X<0? + X=0?
E521 C=0 R<- CLA X=0? 99 RDN
E522 RCR 4 SF 25 GTO IND T
E523 WRIT 11(a) FC?C 25 SF 11 OFF
E524 RTN GTO 16

42*LBL 11
"p SBAMR" PMTK
"↓9cY=ā++++"
"↑↓+++++zP++++" RCL ↑
STO 08 RDN GTO IND X

51*LBL 01
83 RCL \ GTO 10

55*LBL 02
66 RCL ] GTO 10

59*LBL 03
65 RCL [ GTO 10

63*LBL 05
RCL 05 RCL 08 /
RCL 04 / FS? 01
RCL 07 FS? 01 /
RCL 02 RCL 03 FS? 01
* FC? 01 - / STO 01
"pR" GTO 05

83*LBL 04
77 ENTER↑ SF 00
    
```

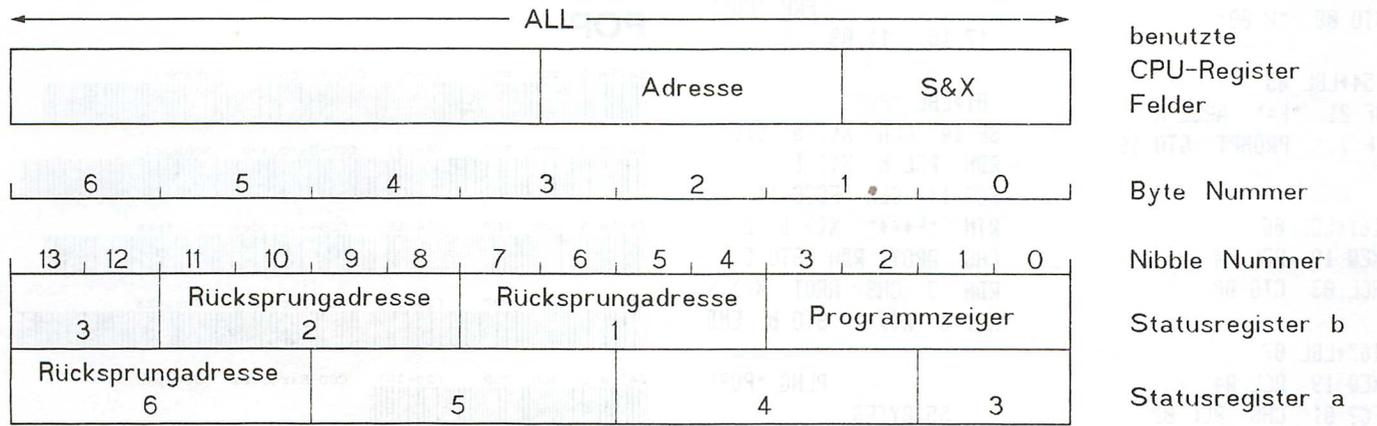
IN 1
IN 2 Normaler
IN 3 UP-Ablauf
IN 4
IN 5
ZUR. IN 4
ZUR. IN 3
ZUR. IN 2
ZUR. IN 1
NACH 1 I.HS

IN 1
IN 2
IN 3 synthet.
IN 4 POP
IN 5
ZUR. IN 1
NACH 1 I.HS

IN 1
IN 2 MCODE
IN 3 POP
IN 4
IN 5
ZUR. IN 1
NACH 1 I.HS

Um die Funktionsweise anschaulich zu machen, habe ich die benutzten Status-Register a und b mit Adressen, sowie ein CPU-Register mit den benutzten Feldern aufgezeichnet.

Setzt man im Testprogramm einen weiteren Unterprogramm-Aufruf ein, dann kommt das synthetische POP nicht mehr ins Hauptprogramm zurück, da es selbst eine Unterprogrammebene benötigt.



07*LBL 10
 "p?" FS?C 00 PROMPT
 CF 22 STO 01 RCL 05
 X<>Y / STO 06 X<> L
 ABSP RDH XTOA R†
 GTO 05

103*LBL 12
 SF 01 "B" GTO 00

107*LBL 13
 CF 01 X†2 "D"

111*LBL 00
 FC?C 22 XEQ 06 STO 02
 GTO 01

116*LBL 17
 SF 01 "H" GTO 00

120*LBL 18
 CF 01 X†2 "d"

124*LBL 00
 FC?C 22 XEQ 07 STO 03

128*LBL 01
 FC? 01 SORT GTO 05

132*LBL 14
 "L" FC?C 22 XEQ 08
 STO 04 GTO 05

138*LBL 15
 "G" FC?C 22 XEQ 09
 STO 05 STO Y RCL 08
 SF 25 / FS?C 25
 STO 06 X<>Y GTO 05

151*LBL 33
 STO 00 "R 00"

154*LBL 05
 SF 21 "†=" ARCL X
 "† >:" PROMPT GTO 16

161*LBL 06
 XEQ 19 RCL 04 /
 RCL 03 GTO 00

167*LBL 07
 XEQ 19 RCL 04 /
 FC? 01 CHS RCL 02

174*LBL 00
 FS? 01 / FC? 01 +
 RTN

180*LBL 19
 RCL 06 X=0? XEQ "POP"
 X=0? RTN RCL 01 /
 FC? 01 RTN RCL 07 /
 RTN

193*LBL 08
 XEQ 19 RCL 02 RCL 03
 FC? 01 - FS? 01 * /
 RTN

203*LBL 09
 RCL 02 RCL 03 FC? 01
 - FS? 01 * RCL 04 *
 RCL 01 * RCL 08 *
 FC? 01 RTN RCL 07 *
 RTN

221*LBL 04
 CLA CLD OFF GTO 20
 END
 PLNG "GEN"
 422 BYTES

GEN
 022:FD 0C 20 42 3A 48 20
 44 3A 64 20 4C 20 47
 043:F7 0C 20 53 42 41 4D
 52
 045:FE 07 85 39 81 63 59
 93 01 13 00 00 00 01

046:FF 7F 07 85 00 00 00
 00 00 02 70 00 00 00
 00
 081:F2 0C 52
 088:F2 0C 3F
 123:F1 64

17:15 11.08 PRP "POP"

01*LBL "POP"
 SF 14 CLA X<> a STO \
 RDH RCL b X<> [
 FC? 14 CLA FC?C 14
 RTN "†+++ X<> [2
 CHS ARDT RDH STO [
 RDH 3 CHS ARDT X<> \
 X<> a X<> [STO b END

PLNG "POP"
 55 BYTES

17:14 11.08
 61*LBL "TST"
 CF 00 SF 01

04*LBL 10
 XEQ 01 "NACH 1 I.HS"
 AVIEW PSE FS?C 00
 GTO 08 FS?C 01 GTO 10
 SF 00 GTO 10

15*LBL 01
 "IN 1" AVIEW PSE
 XEQ 02 "ZUR. IN 1"
 AVIEW PSE RTN

24*LBL 02
 "IN 2" AVIEW PSE
 XEQ 03 "ZUR. IN 2"
 AVIEW PSE RTN

33*LBL 03
 "IN 3" AVIEW PSE
 XEQ 04 "ZUR. IN 3"
 AVIEW PSE RTN

42*LBL 04
 "IN 4" AVIEW PSE
 XEQ 05 "ZUR. IN 4"
 AVIEW PSE RTN

51*LBL 05
 "IN 5" AVIEW PSE
 FC? 00 GTO 06
 XROM 17,17 XROM 17,17
 XROM 17,17 RTN

61*LBL 06
 FS? 01 RTN XEQ "POP"
 XEQ "POP" XEQ "POP"

67*LBL 07
 RTN

69*LBL 08
 STOP GTO 08 END
 PLNG "TST"
 184 BYTES

19:05 11.08.89
 E3C8 LD0R 2
 E3C9 JC +01 E3CA
 E3CA LD0R 0
 E3CB R= 3
 E3CC READ 11(a)
 E3CD A=C ALL
 E3CE READ 12(b)
 E3CF A<>C R<-
 E3D0 RCR 4
 E3D1 A<>C R<-
 E3D2 WRIT 12(b)
 E3D3 A<>C ALL
 E3D4 C=0 R<-
 E3D5 RCR 4
 E3D6 WRIT 11(a)
 E3D7 RTN

19:06 11.08.89
 E3C8 090 P
 E3C9 00F 0
 E3CA 010 P
 E3CB 01C \
 E3CC 2F8 8
 E3CD 10E N
 E3CE 338 8
 E3CF 0AA *
 E3D0 07C <
 E3D1 0AA *
 E3D2 328 <
 E3D3 0AE >
 E3D4 04A J
 E3D5 07C <
 E3D6 2E8 <
 E3D7 3E0

Günter Schapka
 Rebusgasse 11
 6100 Darmstadt

POP

Zeile 1 von POP (1-5)	CCD-Barcodes	Schapka
Zeile 2 von POP (6-13)	CCD-Barcodes	Schapka
Zeile 3 von POP (14-19)	CCD-Barcodes	Schapka
Zeile 4 von POP (20-27)	CCD-Barcodes	Schapka
Zeile 5 von POP (28-28)	CCD-Barcodes	Schapka

Editorerweiterung für den HP41CX

Trotz aller Trivialität, hier meine Editorerweiterung:

XEQ "EDFL" (oder ASN "EDFL" 15, LN), beim Prompt 'EDFL?' kann mit R/S die aktuelle Datei verwendet werden, bzw. die zu bearbeitende Datei eingegeben werden. Damit befindet man sich im Editor (wie mit ED). Nach Betätigen der ON-Taste erscheint dann ein Eingabemenü in der Form 'R:D,I,M,S?' oder 'C:D,I,M,S?'. Das erste Zeichen (R oder C) zeigt den momentanen Modus an (Record oder Character). Zu diesem Zeitpunkt sind folgende Tasten gültig:

- M schaltet zwischen R(ecord) und C(haracter) Modus hin und her
- D löscht eine Zeile oder X-Zeichen (je nach Modus)
- I fügt Zeilen oder Zeichen ein
- S Sucht Zeichenkette (ab der momentanen Position)
- E schaltet den Rechner aus (Exit)

Falls keine Taste gedrückt wird, kehrt das Programm wieder in den Editor zurück. Durch dieses Programm kann das Antwortverhalten bei längeren Dateien verbessert werden.

```

01LBL "EDFL"
02 CF 00
03 "EDFL?"
04 XEQ 03
05 FC?C 23
06 CLA
07LBL 00
08 ED
09LBL 01
10 "R"
11 FS? 00
12 "C"
13 ": D,I,M,S ?"
14 AVIEW
15 GETKEY
16 CLD
17 CLA
18 XEQ IND X
19 CLA
20 GTO 00
21LBL 03
22 CF 23
23 AON
24 STOP
25 AOFF
26 RTN
27LBL 14
28 FS? 00
29 GTO 16
30 DELREC
31 RTN
32LBL 16
33 "DC CNT?"
34 PROMPT
35 DELCHR
36 RTN
37LBL 24
38 FS? 00
39 GTO 26
40 "IR TXT?"
41 XEQ 03
42 FC?C 23
43 RTN
44 INSREC
45 GETREC
46 GTO 24
47LBL 26
48 "IC CHRS?"
49 XEQ 03
50 INSCHR
51 RTN
52LBL 34
53 FC?C 00
54 SF 00
55 GTO 01
56LBL 53
57 RCLPT
58 "SEARCH TXT?"
59 XEQ 03
60 POSFL
61 RTN
62LBL 15
63 OFF
64 END
    
```

```

Betriebsmodus (CF 00 Zeile, SF 00 Zeichen)
Prompt für Dateiname
Abfrage Dateiname

falls R/S aktuelle Datei
Hauptschleife:
  Editoraufruf

  R(ecord) mode

  bzw. C(haracter) mode
  "┆:..." append!
  Auswahl 'Delete, Insert, Mode, Search, Exit'
  Abfrage, 0 falls kein Tastendruck

  Abarbeiten der Routine ('C' -> LBL 13)

  -> Hauptschleife
  Zeicheneingabe:
  Alpha-Flag

  Eingabe in Alpha Mode

Taste 'D'
  falls Zeichenmode
  weiter mit 16
  lösche Record

Zeichen löschen:
  Abfrage Zeichenanzahl

  lösche X Zeichen

Taste 'I'
  falls Zeichenmode
  weiter mit 26

  Einlesen einer Zeile
  falls nur 'R/S'
  Ende Zeileneingabe
  Einfügen Zeile
  überspringe neue Zeile
  nächste Zeile
  Zeicheneingabe:

  lese Zeichen
  Einfügen

Taste 'M'
  invertiere Flag 00

Taste 'S'
  momentane Position

  Eingabe Suchstring
  Positionieren

Taste 'E'
  Rechner aus
    
```

Andreas Huemer
Postfach 115
A-5600 St.Johann / Pg

UPLE . . .

was war das denn gleich ???

In letzter Zeit tauchten viele Fragen zu einer sogenannten HP41-Programmbibliothek auf, womit oft Fragen zu der UPLE gemeint waren, aber kurz einmal die Historie dieser 4 Buchstaben:

UPLE ist die Abkürzung für User's Program Library Europe, was nichts anderes heißt als Benutzer Programmbibliothek Europa.

Diese Programmbibliothek, zur Unterstützung der programmierbaren Taschenrechner HP67/97 und HP34 Ende der 70-er Jahre in Genf gegründet, wurde von Mitarbeitern von Hewlett Packard organisiert und gepflegt. Ab 1979 kam dann ja der damals wie heute noch revolutionäre Taschencomputer, oder von seiner Größe her betrachtet, Taschenrechner HP41 auf den Markt und stellte alles bis dahin dagewesene schnell in den Schatten. Es entstanden binnen kürzester Zeit unzählige Lösungen für alle Fachgebiete.

Um das Jahr 1984 herum kam bei Hewlett Packard wohl die Erkenntnis zum tragen, daß es billiger sei, wenn sich lokale Clubs um die Fragen und Sorgen der Anwender kümmern, ebenso um die Veröffentlichung der entstehenden anwendungsspezifischen Lösungen auf allen Gebieten der Wissenschaft. Man beschloß die Auflösung dieser Programmbibliothek und hätte wahrscheinlich alle Programme mitsamt der Dokumentation in den Genfer See gekippt, wenn sich der CCD damals nicht in einer Eilaktion dazu entschlossen hätte das gesamte Material nach Frankfurt zu karren.

Zuerst nur zwischengelagert wurde dann der gesamte Bestand an HP41 Programmen komplett durchforstet und in eine Datenbank eingetragen, da wir festgestellt hatten, daß man den zur Zeit der Auflösung dieser Bibliothek herausgegebenen Katalog eigent-

Fortsetzung auf Seite 48

28-Mini-Pack

von Ralf Pfeifer

SORT sortiert Listen, die entweder reelle Zahlen oder Strings enthalten. Das kleinste Element kommt dabei ganz nach links, das größte ganz nach rechts. Wer es genau andersherum braucht, dreht in Zeile 5 die "Größer"-Funktion (>) in ein "Kleiner" (<) um. Auf 28C Modellen kann das Programm versagen, denn die haben einen Bug. Die Tests arbeiten bei Strings mit gleichen Anfangsbuchstaben falsch. Beispiel: "AA" "AB" < ergibt unsinnigerweise 0. Um eine Liste mit N Objekten zu sortieren, muß SORT $N(N-1)/2$ Tests ausführen, die Rechenzeit steigt also quadratisch mit N. Dieses Programm ist erheblich schneller als das im HP-28S-Handbuch vorgestellte (beim dortigen Beispiel 5 mal schneller).

Manchmal ist es sehr praktisch, wenn man alle 64 Flags auf einmal setzen kann. Ich benutze dazu ein Programm mit dem Namen **TOP** im Home-Directory und **CUSTOM**-Menue, welches die Effekte von Experimenten rückgängig macht. Wenn ich in irgendeinem Subdirectory bin, bringt mich **SHIFT CUSTOM TOP** wieder zurück, beendet alle unterbrochenen Programme und wählt das **USER-Menue**, außerdem ordnet eine Binärzahl und **STOF** den Status aller 64 Flags.

Um diese Binärzahl einfach zu ermitteln, habe ich **FLAG** geschrieben. Nach dem Start mit **FLAG** fragt der Rechner nach dem gewünschten Zustand von Flag 1. Jetzt können die Tasten **S** oder **1** zum Setzen und die Tasten **C** oder **0** zum Löschen benutzt werden. Andere Tasten ignoriert das Programm. Nachdem Flag 64 abgefragt wurde, endet das Programm automatisch, und in Ebene 1 bleibt eine Binärzahl zurück. Diese Zahl ist am besten im **HEX-Modus** lesbar und kann (gefolgt von einem **STWS**-Befehl) in das eigene Programm eingebaut werden. Doch Vorsicht, diese Speicherung klappt nur dann wirklich sicher, wenn das Programm vorher noch 64 **STWS** ausführt. Ein besonderer Kunstgriff ist die Abfrage der erlaubten Tasten: Diesen ordnet die Funktion **KEY** genau einen Buchstaben zu. Nun fragt **POS**, ob der String den **KEY** geliefert hat, auch im String "SC10" (Zeile 10) vorkommt. Falls nicht, liefert **POS** eine 0 und das Programm ignoriert die Eingabe, falls doch, gibt **POS** eine ungerade Zahl für Setzen und eine gerade Zahl für Löschen zurück, die dann mit **2 MOD** entschlüsselt werden kann.

LNG ermittelt die Länge von Objekten, die in Ebene 1 stehen. Über die Schwierigkeiten die damit verbunden sind, habe ich bereits in **PRISMA 88.5.43** berichtet,

denn holt man ein Programm mit **RCL** in den Stack, so braucht diese Kopie nur 2,5 Bytes, solange sie nicht verändert wird. Inzwischen ist mir aufgefallen, daß "1 ->LIST" vollständige Kopien erstellt, und **MEM SWAP DROP MEM** den Speicherbedarf mißt. Doch auch hier steckt der Teufel im Detail: Objekte die im Stack stehen, bekommen den Stackzuschlag von 5,5 Bytes in jeder Ebene. In 88.5.44 habe ich also 23 Bytes für ein Programm im Stack ermittelt, minus 5,5 Bytes heißt, daß es 17,5 Bytes benötigt. In seinem Buch "HP-28 Insights" gibt der Entwickler Wickes nur 12,5 Bytes an. Daher habe ich erneut gemessen und fand Wickes nur dann bestätigt, wenn ein Programm in einem bestehenden eingeschlossen ist. Zu allem Überfluß brauchen Programme in Listen offenbar nur 10 Bytes, während alle anderen Objekte ihre Länge nicht verändern.

Aus diesem Grund habe ich in **LNG** definiert, daß falls in Ebene 1 ein Programm steht, das erste und letzte Programmbegrenzungszeichen zusammen 17,5 Bytes brauchen (wer Wickes glaubt kann die 10.5 in Zeile 2 in eine 5.5 ändern). Wer weiter mit **LNG** sucht findet dann auch den Null-String (""), die 5 Konstanten (e, i, pi, MAXR, MINR auch wenn sie der Stack als Zahlen zeigt). Bei den 28C Modellen benötigen die globalen Namen A...Z nur 2,5 Bytes. Außerdem spart man jeweils 5 Bytes, wenn zwischen **IF/IFERR...THEN, THEN...ELSE, THEN...END, ELSE...END** und **REPEAT...END** genau ein Objekt steht, was bei **IF...THEN** immer möglich ist! **LNG** verlangt in Ebene 1 das zu messende Objekt und ersetzt es dann durch die Zahl seiner Bytes. Dabei schaltet **LNG** den **LAST**-Modus vorübergehend ab. **COMMAND** und **UNDO** müssen dagegen nur bei der Längenmessung über die Tastatur abgeschaltet werden, denn ein laufendes Programm hat keinen Einfluß auf diese Funktion. Es ist ja so möglich, mit **UNDO** nach dem Ende eines Programmes den Stackinhalt vor Programmstart und mit **COMMAND** die entsprechende Befehlszeile wiederherzustellen.

PRP (aus 89.4.36) wurde nicht nur kürzer, sondern mißt jetzt ebenfalls die Programmlänge und druckt sie aus, sodaß auch 28C-Benutzer jetzt vorher wissen warum der Speicher zu klein ist. Dazu wie bisher das Programm oder seinen 'Namen' in Ebene 1 eingeben und **PRP** drücken. Der 10 **CHR** Befehl in Zeile 18 wurde nur zur besseren Darstellung verwendet, es ist das **NEWLINE** auf der Tastatur.

```

SORT
001 * LIST → DUP + n
    *
003   FOR a 2 a
       START DUP2
005   IF >
       THEN SWAP
007   END a ROLLD
       NEXT LAST
009 ROLLD -1
    STEP n →LIST
011 *
    * 95 Bytes

1: ( "GTI" "CCD" "KGB"
    "CIA" "007" "0815"
    "ADAC" "PP" "P" )

SORT
1: ( "007" "0815"
    "ADAC" "CCD" "CIA"
    "GTI" "KGB" "P" "PP"
    )

TOP
001 * HOME 8 SO STWS
    # F089FF8400000000h
003 STOF 23 MENU KILL
    * 58,5 Bytes

FLAG
001 * # 0h CLLCD
    "1 oder S = SET
003 0 oder C = CLEAR"
005 2 DISP 1 64 DUP STWS
    FOR f "FLAG" f
    →STR + " 2" + 1 DISP
007 DO KEY
    UNTIL
009 IF DUP
    THEN "SC10"
011 ROT
    IF POS
    THEN LAST 2
013 MOD ROT + RR SWAP
015 440 ,03 BEEP
    ELSE NOT
017 END
    END
019 NEXT CLMF
021 * 236,5 Bytes

LNG
001 * RCLF 31 CF OVER
    TYPE 3 SAME 10,5 3
003 IFTE ROT 1 →LIST MEM
    SWAP DROP - MEM +
005 SWAP STOF
    * 86 Bytes

'LNG' RCL LNG
1: 86

PRP
001 * CR RCLF SWAP STD
    DUP TYPE 6
003 IF ==
    THEN 27 CHR 253
005 CHR + " " + OVER
    →STR 2 OVER SIZE 1 -
007 SUB + 27 CHR + 252
    CHR + PR1 DROP RCL
009 END →STR " " LAST
    LNG →STR + " Bytes"
011 + + 1
    DO DUP 2
    TE MOD
    THEN DUP 10001
015 % →STR 1 4 SUB
    ELSE " "
017 END " " + ROT +
    2 OVER 10 CHR POS
019 DUP 4 PICK SIZE 2 +
    IFTE 1 - SUB LAST +
021 OVER SIZE SUB SWAP
    PR1 SIZE 24 / CEIL
023 ROT + OVER SIZE
    UNTIL NOT
025 END DROP2 STOF
    * 366,5 Bytes

```

IPD Daten zu IP

0.387099000 Nr. 0 CCD Barcodes von Dr.G.Heilmann	75.780000000 Nr. 20 CCD Barcodes von Dr.G.Heilmann	359.0498000 Nr. 40 CCD Barcodes von Dr.G.Heilmann	360.0000000 Nr. 60 CCD Barcodes von Dr.G.Heilmann
0.205614000 Nr. 1 CCD Barcodes von Dr.G.Heilmann	0.900000000 Nr. 21 CCD Barcodes von Dr.G.Heilmann	35640.00000 Nr. 41 CCD Barcodes von Dr.G.Heilmann	15.000000000 Nr. 61 CCD Barcodes von Dr.G.Heilmann
0.000020000 Nr. 2 CCD Barcodes von Dr.G.Heilmann	54.383800000 Nr. 22 CCD Barcodes von Dr.G.Heilmann	101.2208000 Nr. 42 CCD Barcodes von Dr.G.Heilmann	0.640600000 Nr. 62 CCD Barcodes von Dr.G.Heilmann
102.2794000 Nr. 3 CCD Barcodes von Dr.G.Heilmann	0.508000000 Nr. 23 CCD Barcodes von Dr.G.Heilmann	1.719200000 Nr. 43 CCD Barcodes von Dr.G.Heilmann	0.556800000 Nr. 63 CCD Barcodes von Dr.G.Heilmann
72.515300000 Nr. 4 CCD Barcodes von Dr.G.Heilmann	1.523692000 Nr. 24 CCD Barcodes von Dr.G.Heilmann	16.027000000 Nr. 44 CCD Barcodes von Dr.G.Heilmann	180.0000000 Nr. 64 CCD Barcodes von Dr.G.Heilmann
149400.00000 Nr. 5 CCD Barcodes von Dr.G.Heilmann	0.093313000 Nr. 25 CCD Barcodes von Dr.G.Heilmann	1.011900000 Nr. 45 CCD Barcodes von Dr.G.Heilmann	6.000000000 Nr. 65 CCD Barcodes von Dr.G.Heilmann
7.002900000 Nr. 6 CCD Barcodes von Dr.G.Heilmann	0.000092000 Nr. 26 CCD Barcodes von Dr.G.Heilmann	0.500000000 Nr. 46 CCD Barcodes von Dr.G.Heilmann	50.060000000 Nr. 66 CCD Barcodes von Dr.G.Heilmann
0.001900000 Nr. 7 CCD Barcodes von Dr.G.Heilmann	319.5294000 Nr. 27 CCD Barcodes von Dr.G.Heilmann	36525.00000 Nr. 47 CCD Barcodes von Dr.G.Heilmann	-8.420000000 Nr. 67 CCD Barcodes von Dr.G.Heilmann
47.146000000 Nr. 8 CCD Barcodes von Dr.G.Heilmann	59.858500000 Nr. 28 CCD Barcodes von Dr.G.Heilmann	24.000000000 Nr. 48 CCD Barcodes von Dr.G.Heilmann	22640.00000 Nr. 68 CCD Barcodes von Dr.G.Heilmann
1.185000000 Nr. 9 CCD Barcodes von Dr.G.Heilmann	19080.00000 Nr. 29 CCD Barcodes von Dr.G.Heilmann	1.011950000 Nr. 49 CCD Barcodes von Dr.G.Heilmann	769.0000000 Nr. 69 CCD Barcodes von Dr.G.Heilmann
28.753700000 Nr. 10 CCD Barcodes von Dr.G.Heilmann	1.850300000 Nr. 30 CCD Barcodes von Dr.G.Heilmann	0.092900000 Nr. 50 CCD Barcodes von Dr.G.Heilmann	36.000000000 Nr. 70 CCD Barcodes von Dr.G.Heilmann
0.370500000 Nr. 11 CCD Barcodes von Dr.G.Heilmann	-0.000700000 Nr. 31 CCD Barcodes von Dr.G.Heilmann	8640184.500 Nr. 51 CCD Barcodes von Dr.G.Heilmann	-125.0000000 Nr. 71 CCD Barcodes von Dr.G.Heilmann
0.723332000 Nr. 12 CCD Barcodes von Dr.G.Heilmann	48.786000000 Nr. 32 CCD Barcodes von Dr.G.Heilmann	23925.83600 Nr. 52 CCD Barcodes von Dr.G.Heilmann	2370.0000000 Nr. 72 CCD Barcodes von Dr.G.Heilmann
0.006821000 Nr. 13 CCD Barcodes von Dr.G.Heilmann	0.771000000 Nr. 33 CCD Barcodes von Dr.G.Heilmann	3600.0000000 Nr. 53 CCD Barcodes von Dr.G.Heilmann	-668.0000000 Nr. 73 CCD Barcodes von Dr.G.Heilmann
-0.000048000 Nr. 14 CCD Barcodes von Dr.G.Heilmann	285.4322000 Nr. 34 CCD Barcodes von Dr.G.Heilmann	1.002740000 Nr. 54 CCD Barcodes von Dr.G.Heilmann	412.0000000 Nr. 74 CCD Barcodes von Dr.G.Heilmann
212.6032000 Nr. 15 CCD Barcodes von Dr.G.Heilmann	1.069800000 Nr. 35 CCD Barcodes von Dr.G.Heilmann	Text: 50 48 41 53 45 Nr. 55 CCD Barcodes von Dr.G.Heilmann	212.0000000 Nr. 75 CCD Barcodes von Dr.G.Heilmann
197.8039000 Nr. 16 CCD Barcodes von Dr.G.Heilmann	1.000000000 Nr. 36 CCD Barcodes von Dr.G.Heilmann	16.000006000 Nr. 56 CCD Barcodes von Dr.G.Heilmann	4586.0000000 Nr. 76 CCD Barcodes von Dr.G.Heilmann
58320.00000 Nr. 17 CCD Barcodes von Dr.G.Heilmann	0.016751000 Nr. 37 CCD Barcodes von Dr.G.Heilmann	12.000000000 Nr. 57 CCD Barcodes von Dr.G.Heilmann	192.0000000 Nr. 77 CCD Barcodes von Dr.G.Heilmann
3.393600000 Nr. 18 CCD Barcodes von Dr.G.Heilmann	-0.000042000 Nr. 38 CCD Barcodes von Dr.G.Heilmann	23.439300000 Nr. 58 CCD Barcodes von Dr.G.Heilmann	165.0000000 Nr. 78 CCD Barcodes von Dr.G.Heilmann
0.001000000 Nr. 19 CCD Barcodes von Dr.G.Heilmann	358.4758000 Nr. 39 CCD Barcodes von Dr.G.Heilmann	0.013000000 Nr. 59 CCD Barcodes von Dr.G.Heilmann	206.0000000 Nr. 79 CCD Barcodes von Dr.G.Heilmann

BARCODES

-110.0000000 Nr. 80
CCD Barcodes von Dr.G.Heilmann



148.0000000 Nr. 81
CCD Barcodes von Dr.G.Heilmann



541.0000000 Nr. 82
CCD Barcodes von Dr.G.Heilmann



18520.00000 Nr. 83
CCD Barcodes von Dr.G.Heilmann



21.00000000 Nr. 84
CCD Barcodes von Dr.G.Heilmann



25.00000000 Nr. 85
CCD Barcodes von Dr.G.Heilmann



-526.0000000 Nr. 86
CCD Barcodes von Dr.G.Heilmann



44.00000000 Nr. 87
CCD Barcodes von Dr.G.Heilmann



-31.00000000 Nr. 88
CCD Barcodes von Dr.G.Heilmann



-23.00000000 Nr. 89
CCD Barcodes von Dr.G.Heilmann



11.00000000 Nr. 90
CCD Barcodes von Dr.G.Heilmann



10.00000000 Nr. 91
CCD Barcodes von Dr.G.Heilmann



3423.000000 Nr. 92
CCD Barcodes von Dr.G.Heilmann



187.0000000 Nr. 93
CCD Barcodes von Dr.G.Heilmann



34.00000000 Nr. 94
CCD Barcodes von Dr.G.Heilmann



28.00000000 Nr. 95
CCD Barcodes von Dr.G.Heilmann



149597892.0 Nr. 96
CCD Barcodes von Dr.G.Heilmann



6378.140000 Nr. 97
CCD Barcodes von Dr.G.Heilmann



92.11100000 Nr. 98
CCD Barcodes von Dr.G.Heilmann



1000.000000 Nr. 99
CCD Barcodes von Dr.G.Heilmann



481267.0000 Nr. 100
CCD Barcodes von Dr.G.Heilmann



270.4341640 Nr. 101
CCD Barcodes von Dr.G.Heilmann



-0.001133000 Nr. 102
CCD Barcodes von Dr.G.Heilmann



0.883142000 Nr. 103
CCD Barcodes von Dr.G.Heilmann



477198.0000 Nr. 104
CCD Barcodes von Dr.G.Heilmann



296.1046080 Nr. 105
CCD Barcodes von Dr.G.Heilmann



0.009192000 Nr. 106
CCD Barcodes von Dr.G.Heilmann



0.849108000 Nr. 107
CCD Barcodes von Dr.G.Heilmann



-1934.000000 Nr. 108
CCD Barcodes von Dr.G.Heilmann



259.1832750 Nr. 109
CCD Barcodes von Dr.G.Heilmann



0.002078000 Nr. 110
CCD Barcodes von Dr.G.Heilmann



-0.142008000 Nr. 111
CCD Barcodes von Dr.G.Heilmann



36000.00000 Nr. 112
CCD Barcodes von Dr.G.Heilmann



279.6966780 Nr. 113
CCD Barcodes von Dr.G.Heilmann



0.000303000 Nr. 114
CCD Barcodes von Dr.G.Heilmann



0.768925000 Nr. 115
CCD Barcodes von Dr.G.Heilmann



35999.00000 Nr. 116
CCD Barcodes von Dr.G.Heilmann



358.4758330 Nr. 117
CCD Barcodes von Dr.G.Heilmann



-0.000150000 Nr. 118
CCD Barcodes von Dr.G.Heilmann



0.049750000 Nr. 119
CCD Barcodes von Dr.G.Heilmann



IP

Zeile 1 von IP (1-5) CCD-Barcodes Dr.G.Heilmann



Zeile 2 von IP (6-10) CCD-Barcodes Dr.G.Heilmann



Zeile 3 von IP (11-15) CCD-Barcodes Dr.G.Heilmann



Zeile 4 von IP (16-20) CCD-Barcodes Dr.G.Heilmann



Zeile 5 von IP (21-28) CCD-Barcodes Dr.G.Heilmann



Zeile 6 von IP (29-37) CCD-Barcodes Dr.G.Heilmann



Zeile 7 von IP (38-38) CCD-Barcodes Dr.G.Heilmann



Zeile 8 von IP (39-43) CCD-Barcodes Dr.G.Heilmann



Zeile 9 von IP (44-46) CCD-Barcodes Dr.G.Heilmann



Zeile 10 von IP (47-51) CCD-Barcodes Dr.G.Heilmann



Zeile 11 von IP (52-53) CCD-Barcodes Dr.G.Heilmann



Zeile 12 von IP (54-55) CCD-Barcodes Dr.G.Heilmann



Zeile 13 von IP (56-58) CCD-Barcodes Dr.G.Heilmann



Zeile 14 von IP (59-65) CCD-Barcodes Dr.G.Heilmann



Zeile 15 von IP (66-70) CCD-Barcodes Dr.G.Heilmann



Zeile 16 von IP (71-76) CCD-Barcodes Dr.G.Heilmann



Zeile 17 von IP (77-82) CCD-Barcodes Dr.G.Heilmann



Zeile 18 von IP (83-91) CCD-Barcodes Dr.G.Heilmann



Zeile 19 von IP (92-102) CCD-Barcodes Dr.G.Heilmann



Zeile 20 von IP (103-107) CCD-Barcodes Dr.G.Heilmann



Zeile 21 von IP (108-117) CCD-Barcodes Dr.G.Heilmann



Zeile 22 von IP (118-123) CCD-Barcodes Dr.G.Heilmann



Zeile 23 von IP (124-130) CCD-Barcodes Dr.G.Heilmann



Zeile 24 von IP (131-140) CCD-Barcodes Dr.G.Heilmann



Zeile 25 von IP (141-150) CCD-Barcodes Dr.G.Heilmann



Zeile 26 von IP (151-156) CCD-Barcodes Dr.G.Heilmann
 Zeile 27 von IP (157-162) CCD-Barcodes Dr.G.Heilmann
 Zeile 28 von IP (163-169) CCD-Barcodes Dr.G.Heilmann
 Zeile 29 von IP (170-176) CCD-Barcodes Dr.G.Heilmann
 Zeile 30 von IP (177-182) CCD-Barcodes Dr.G.Heilmann
 Zeile 31 von IP (183-190) CCD-Barcodes Dr.G.Heilmann
 Zeile 32 von IP (191-200) CCD-Barcodes Dr.G.Heilmann
 Zeile 33 von IP (201-209) CCD-Barcodes Dr.G.Heilmann
 Zeile 34 von IP (210-220) CCD-Barcodes Dr.G.Heilmann
 Zeile 35 von IP (221-226) CCD-Barcodes Dr.G.Heilmann
 Zeile 36 von IP (227-235) CCD-Barcodes Dr.G.Heilmann
 Zeile 37 von IP (236-241) CCD-Barcodes Dr.G.Heilmann
 Zeile 38 von IP (242-249) CCD-Barcodes Dr.G.Heilmann
 Zeile 39 von IP (250-257) CCD-Barcodes Dr.G.Heilmann
 Zeile 40 von IP (258-265) CCD-Barcodes Dr.G.Heilmann
 Zeile 41 von IP (266-272) CCD-Barcodes Dr.G.Heilmann
 Zeile 42 von IP (273-280) CCD-Barcodes Dr.G.Heilmann
 Zeile 43 von IP (281-287) CCD-Barcodes Dr.G.Heilmann
 Zeile 44 von IP (288-293) CCD-Barcodes Dr.G.Heilmann
 Zeile 45 von IP (294-305) CCD-Barcodes Dr.G.Heilmann
 Zeile 46 von IP (306-312) CCD-Barcodes Dr.G.Heilmann
 Zeile 47 von IP (313-322) CCD-Barcodes Dr.G.Heilmann
 Zeile 48 von IP (323-331) CCD-Barcodes Dr.G.Heilmann
 Zeile 49 von IP (332-342) CCD-Barcodes Dr.G.Heilmann
 Zeile 50 von IP (343-350) CCD-Barcodes Dr.G.Heilmann
 Zeile 51 von IP (351-358) CCD-Barcodes Dr.G.Heilmann

Zeile 52 von IP (359-367) CCD-Barcodes Dr.G.Heilmann
 Zeile 53 von IP (368-378) CCD-Barcodes Dr.G.Heilmann
 Zeile 54 von IP (379-388) CCD-Barcodes Dr.G.Heilmann
 Zeile 55 von IP (389-397) CCD-Barcodes Dr.G.Heilmann
 Zeile 56 von IP (398-406) CCD-Barcodes Dr.G.Heilmann
 Zeile 57 von IP (407-417) CCD-Barcodes Dr.G.Heilmann
 Zeile 58 von IP (418-428) CCD-Barcodes Dr.G.Heilmann
 Zeile 59 von IP (429-440) CCD-Barcodes Dr.G.Heilmann
 Zeile 60 von IP (441-448) CCD-Barcodes Dr.G.Heilmann
 Zeile 61 von IP (449-459) CCD-Barcodes Dr.G.Heilmann
 Zeile 62 von IP (460-470) CCD-Barcodes Dr.G.Heilmann
 Zeile 63 von IP (471-478) CCD-Barcodes Dr.G.Heilmann
 Zeile 64 von IP (479-488) CCD-Barcodes Dr.G.Heilmann
 Zeile 65 von IP (489-499) CCD-Barcodes Dr.G.Heilmann
 Zeile 66 von IP (500-510) CCD-Barcodes Dr.G.Heilmann
 Zeile 67 von IP (511-517) CCD-Barcodes Dr.G.Heilmann
 Zeile 68 von IP (518-527) CCD-Barcodes Dr.G.Heilmann
 Zeile 69 von IP (528-535) CCD-Barcodes Dr.G.Heilmann
 Zeile 70 von IP (536-545) CCD-Barcodes Dr.G.Heilmann
 Zeile 71 von IP (546-554) CCD-Barcodes Dr.G.Heilmann
 Zeile 72 von IP (555-558) CCD-Barcodes Dr.G.Heilmann
 Zeile 73 von IP (559-564) CCD-Barcodes Dr.G.Heilmann
 Zeile 74 von IP (565-573) CCD-Barcodes Dr.G.Heilmann
 Zeile 75 von IP (574-584) CCD-Barcodes Dr.G.Heilmann
 Zeile 76 von IP (585-587) CCD-Barcodes Dr.G.Heilmann
 Zeile 77 von IP (588-593) CCD-Barcodes Dr.G.Heilmann

BARCODES

Zeile 78 von IP (594-601) CCD-Barcodes Dr.G.Heilmann



Zeile 79 von IP (602-610) CCD-Barcodes Dr.G.Heilmann



Zeile 80 von IP (611-621) CCD-Barcodes Dr.G.Heilmann



Zeile 81 von IP (622-630) CCD-Barcodes Dr.G.Heilmann



Zeile 82 von IP (631-636) CCD-Barcodes Dr.G.Heilmann



Zeile 83 von IP (637-643) CCD-Barcodes Dr.G.Heilmann



Zeile 84 von IP (644-653) CCD-Barcodes Dr.G.Heilmann



Zeile 85 von IP (654-662) CCD-Barcodes Dr.G.Heilmann



Zeile 86 von IP (663-670) CCD-Barcodes Dr.G.Heilmann



Zeile 87 von IP (671-679) CCD-Barcodes Dr.G.Heilmann



Zeile 88 von IP (680-687) CCD-Barcodes Dr.G.Heilmann



Zeile 89 von IP (688-696) CCD-Barcodes Dr.G.Heilmann



Zeile 90 von IP (697-705) CCD-Barcodes Dr.G.Heilmann



Zeile 91 von IP (706-716) CCD-Barcodes Dr.G.Heilmann



Zeile 92 von IP (717-727) CCD-Barcodes Dr.G.Heilmann



Zeile 93 von IP (728-738) CCD-Barcodes Dr.G.Heilmann



Zeile 94 von IP (739-747) CCD-Barcodes Dr.G.Heilmann



Zeile 95 von IP (748-758) CCD-Barcodes Dr.G.Heilmann



Zeile 96 von IP (759-770) CCD-Barcodes Dr.G.Heilmann



Zeile 97 von IP (771-781) CCD-Barcodes Dr.G.Heilmann



Zeile 98 von IP (782-793) CCD-Barcodes Dr.G.Heilmann



Zeile 99 von IP (794-803) CCD-Barcodes Dr.G.Heilmann



Zeile 100 von IP (804-814) CCD-Barcodes Dr.G.Heilmann



Zeile 101 von IP (815-825) CCD-Barcodes Dr.G.Heilmann



Zeile 102 von IP (826-836) CCD-Barcodes Dr.G.Heilmann



Zeile 103 von IP (837-847) CCD-Barcodes Dr.G.Heilmann



Zeile 104 von IP (848-857) CCD-Barcodes Dr.G.Heilmann



Zeile 105 von IP (858-868) CCD-Barcodes Dr.G.Heilmann



Zeile 106 von IP (869-878) CCD-Barcodes Dr.G.Heilmann



Zeile 107 von IP (879-889) CCD-Barcodes Dr.G.Heilmann



Zeile 108 von IP (890-900) CCD-Barcodes Dr.G.Heilmann



Zeile 109 von IP (901-908) CCD-Barcodes Dr.G.Heilmann



Zeile 110 von IP (909-918) CCD-Barcodes Dr.G.Heilmann



Zeile 111 von IP (919-923) CCD-Barcodes Dr.G.Heilmann



Zeile 112 von IP (924-923) CCD-Barcodes Dr.G.Heilmann



GEO

Zeile 1 von GEO (1-5) CCD-Barcodes Michael Schilli



Zeile 2 von GEO (6-11) CCD-Barcodes Michael Schilli



Zeile 3 von GEO (12-21) CCD-Barcodes Michael Schilli



Zeile 4 von GEO (22-30) CCD-Barcodes Michael Schilli



Zeile 5 von GEO (31-40) CCD-Barcodes Michael Schilli



Zeile 6 von GEO (41-50) CCD-Barcodes Michael Schilli



Zeile 7 von GEO (51-59) CCD-Barcodes Michael Schilli



Zeile 8 von GEO (60-66) CCD-Barcodes Michael Schilli



Zeile 9 von GEO (67-73) CCD-Barcodes Michael Schilli



Zeile 10 von GEO (74-78) CCD-Barcodes Michael Schilli



Zeile 11 von GEO (79-88) CCD-Barcodes Michael Schilli



Zeile 12 von GEO (89-95) CCD-Barcodes Michael Schilli



Zeile 13 von GEO (96-98) CCD-Barcodes Michael Schilli



Zeile 14 von GEO (99-109) CCD-Barcodes Michael Schilli



Zeile 15 von GEO (110-111) CCD-Barcodes Michael Schilli



Zeile 16 von GEO (112-119) CCD-Barcodes Michael Schilli



BARCODES

Zeile 17 von GEO (120-125) CCD-Barcodes Michael Schilli



Zeile 18 von GEO (126-132) CCD-Barcodes Michael Schilli



Zeile 19 von GEO (133-138) CCD-Barcodes Michael Schilli



Zeile 20 von GEO (139-147) CCD-Barcodes Michael Schilli



Zeile 21 von GEO (148-156) CCD-Barcodes Michael Schilli



Zeile 22 von GEO (157-166) CCD-Barcodes Michael Schilli



Zeile 23 von GEO (167-174) CCD-Barcodes Michael Schilli



Zeile 24 von GEO (175-182) CCD-Barcodes Michael Schilli



Zeile 25 von GEO (183-190) CCD-Barcodes Michael Schilli



Zeile 26 von GEO (191-198) CCD-Barcodes Michael Schilli



Zeile 27 von GEO (199-203) CCD-Barcodes Michael Schilli



Zeile 28 von GEO (204-209) CCD-Barcodes Michael Schilli



Zeile 29 von GEO (210-219) CCD-Barcodes Michael Schilli



Zeile 30 von GEO (220-230) CCD-Barcodes Michael Schilli



Zeile 31 von GEO (231-239) CCD-Barcodes Michael Schilli



Zeile 32 von GEO (240-250) CCD-Barcodes Michael Schilli



Zeile 33 von GEO (251-261) CCD-Barcodes Michael Schilli



Zeile 34 von GEO (262-267) CCD-Barcodes Michael Schilli



Zeile 35 von GEO (268-272) CCD-Barcodes Michael Schilli



Zeile 36 von GEO (273-277) CCD-Barcodes Michael Schilli



Zeile 37 von GEO (278-282) CCD-Barcodes Michael Schilli



Zeile 38 von GEO (283-287) CCD-Barcodes Michael Schilli



Zeile 39 von GEO (288-297) CCD-Barcodes Michael Schilli



Zeile 40 von GEO (298-305) CCD-Barcodes Michael Schilli



Zeile 41 von GEO (306-310) CCD-Barcodes Michael Schilli



Zeile 42 von GEO (311-319) CCD-Barcodes Michael Schilli



Zeile 43 von GEO (320-327) CCD-Barcodes Michael Schilli



Zeile 44 von GEO (328-336) CCD-Barcodes Michael Schilli



Zeile 45 von GEO (337-342) CCD-Barcodes Michael Schilli



Zeile 46 von GEO (343-348) CCD-Barcodes Michael Schilli



Zeile 47 von GEO (349-353) CCD-Barcodes Michael Schilli



Zeile 48 von GEO (354-359) CCD-Barcodes Michael Schilli



Zeile 49 von GEO (360-363) CCD-Barcodes Michael Schilli



Zeile 50 von GEO (364-369) CCD-Barcodes Michael Schilli



Zeile 51 von GEO (370-376) CCD-Barcodes Michael Schilli



Zeile 52 von GEO (377-385) CCD-Barcodes Michael Schilli



Zeile 53 von GEO (386-392) CCD-Barcodes Michael Schilli



Zeile 54 von GEO (393-399) CCD-Barcodes Michael Schilli



Zeile 55 von GEO (400-403) CCD-Barcodes Michael Schilli



Zeile 56 von GEO (404-411) CCD-Barcodes Michael Schilli



Zeile 57 von GEO (412-422) CCD-Barcodes Michael Schilli



Zeile 58 von GEO (423-433) CCD-Barcodes Michael Schilli



Zeile 59 von GEO (434-443) CCD-Barcodes Michael Schilli



Zeile 60 von GEO (444-453) CCD-Barcodes Michael Schilli



Zeile 61 von GEO (454-463) CCD-Barcodes Michael Schilli



Zeile 62 von GEO (464-469) CCD-Barcodes Michael Schilli



Zeile 63 von GEO (470-477) CCD-Barcodes Michael Schilli



Zeile 64 von GEO (478-486) CCD-Barcodes Michael Schilli



Zeile 65 von GEO (487-496) CCD-Barcodes Michael Schilli



Zeile 66 von GEO (497-506) CCD-Barcodes Michael Schilli



Zeile 67 von GEO (507-515) CCD-Barcodes Michael Schilli



Zeile 68 von GEO (516-525) CCD-Barcodes Michael Schilli



BARCODES

Zeile 69 von GEO (526-535) CCD-Barcodes Michael Schilli



Zeile 70 von GEO (536-545) CCD-Barcodes Michael Schilli



Zeile 71 von GEO (546-555) CCD-Barcodes Michael Schilli



Zeile 72 von GEO (556-565) CCD-Barcodes Michael Schilli



Zeile 73 von GEO (566-574) CCD-Barcodes Michael Schilli



Zeile 74 von GEO (575-580) CCD-Barcodes Michael Schilli



Zeile 75 von GEO (581-588) CCD-Barcodes Michael Schilli



Zeile 76 von GEO (589-595) CCD-Barcodes Michael Schilli



Zeile 77 von GEO (596-600) CCD-Barcodes Michael Schilli



Zeile 78 von GEO (601-608) CCD-Barcodes Michael Schilli



Zeile 79 von GEO (609-613) CCD-Barcodes Michael Schilli



Zeile 80 von GEO (614-618) CCD-Barcodes Michael Schilli



Zeile 81 von GEO (619-622) CCD-Barcodes Michael Schilli



Zeile 82 von GEO (623-630) CCD-Barcodes Michael Schilli



Zeile 83 von GEO (631-640) CCD-Barcodes Michael Schilli



Zeile 84 von GEO (641-651) CCD-Barcodes Michael Schilli



Zeile 85 von GEO (652-662) CCD-Barcodes Michael Schilli



Zeile 86 von GEO (663-671) CCD-Barcodes Michael Schilli



Zeile 87 von GEO (672-682) CCD-Barcodes Michael Schilli



Zeile 88 von GEO (683-691) CCD-Barcodes Michael Schilli



Zeile 89 von GEO (692-701) CCD-Barcodes Michael Schilli



Zeile 90 von GEO (702-712) CCD-Barcodes Michael Schilli



Zeile 91 von GEO (713-721) CCD-Barcodes Michael Schilli



Zeile 92 von GEO (722-729) CCD-Barcodes Michael Schilli



Zeile 93 von GEO (730-733) CCD-Barcodes Michael Schilli



Zeile 94 von GEO (734-736) CCD-Barcodes Michael Schilli



Zeile 95 von GEO (737-740) CCD-Barcodes Michael Schilli



Zeile 96 von GEO (741-743) CCD-Barcodes Michael Schilli



Zeile 97 von GEO (744-749) CCD-Barcodes Michael Schilli



Zeile 98 von GEO (750-757) CCD-Barcodes Michael Schilli



Zeile 99 von GEO (758-765) CCD-Barcodes Michael Schilli



Zeile 100 von GEO (766-772) CCD-Barcodes Michael Schilli



Zeile 101 von GEO (773-778) CCD-Barcodes Michael Schilli



Zeile 102 von GEO (779-788) CCD-Barcodes Michael Schilli



Zeile 103 von GEO (789-797) CCD-Barcodes Michael Schilli



Zeile 104 von GEO (798-806) CCD-Barcodes Michael Schilli



Zeile 105 von GEO (807-815) CCD-Barcodes Michael Schilli



Zeile 106 von GEO (816-825) CCD-Barcodes Michael Schilli



Zeile 107 von GEO (826-832) CCD-Barcodes Michael Schilli



Zeile 108 von GEO (833-840) CCD-Barcodes Michael Schilli



Zeile 109 von GEO (841-848) CCD-Barcodes Michael Schilli



Zeile 110 von GEO (849-857) CCD-Barcodes Michael Schilli



Zeile 111 von GEO (858-866) CCD-Barcodes Michael Schilli



Zeile 112 von GEO (867-874) CCD-Barcodes Michael Schilli



Zeile 113 von GEO (875-883) CCD-Barcodes Michael Schilli



Zeile 114 von GEO (884-891) CCD-Barcodes Michael Schilli



Zeile 115 von GEO (892-902) CCD-Barcodes Michael Schilli



Zeile 116 von GEO (903-910) CCD-Barcodes Michael Schilli



Zeile 117 von GEO (911-915) CCD-Barcodes Michael Schilli



Zeile 118 von GEO (916-921) CCD-Barcodes Michael Schilli



Zeile 119 von GEO (922-930) CCD-Barcodes Michael Schilli



Zeile 120 von GEO (931-935) CCD-Barcodes Michael Schilli



Zelle 121 von GEO (936-942) CCD-Barcodes Michael Schilli



Zelle 122 von GEO (943-950) CCD-Barcodes Michael Schilli



Zelle 123 von GEO (951-955) CCD-Barcodes Michael Schilli



Zelle 124 von GEO (956-965) CCD-Barcodes Michael Schilli



Zelle 125 von GEO (966-973) CCD-Barcodes Michael Schilli



Zelle 126 von GEO (974-981) CCD-Barcodes Michael Schilli



Zelle 127 von GEO (982-991) CCD-Barcodes Michael Schilli



Zelle 128 von GEO (992-1002) CCD-Barcodes Michael Schilli



Zelle 129 von GEO (1003-1013) CCD-Barcodes Michael Schilli



Zelle 130 von GEO (1014-1021) CCD-Barcodes Michael Schilli



Zelle 131 von GEO (1022-1032) CCD-Barcodes Michael Schilli



Zelle 132 von GEO (1033-1040) CCD-Barcodes Michael Schilli



Zelle 133 von GEO (1041-1047) CCD-Barcodes Michael Schilli



Zelle 134 von GEO (1048-1054) CCD-Barcodes Michael Schilli



Zelle 135 von GEO (1055-1063) CCD-Barcodes Michael Schilli



Zelle 136 von GEO (1064-1072) CCD-Barcodes Michael Schilli



Zelle 137 von GEO (1073-1080) CCD-Barcodes Michael Schilli



Zelle 138 von GEO (1081-1088) CCD-Barcodes Michael Schilli



Zelle 139 von GEO (1089-1094) CCD-Barcodes Michael Schilli



Zelle 7 von GEW (12-39) CCD-Barcodes Schapka



Zelle 8 von GEW (40-43) CCD-Barcodes Schapka



Zelle 9 von GEW (44-45) CCD-Barcodes Schapka



Zelle 10 von GEW (46-46) CCD-Barcodes Schapka



Zelle 11 von GEW (47-50) CCD-Barcodes Schapka



Zelle 12 von GEW (51-58) CCD-Barcodes Schapka



Zelle 13 von GEW (59-67) CCD-Barcodes Schapka



Zelle 14 von GEW (68-77) CCD-Barcodes Schapka



Zelle 15 von GEW (78-84) CCD-Barcodes Schapka



Zelle 16 von GEW (85-92) CCD-Barcodes Schapka



Zelle 17 von GEW (93-102) CCD-Barcodes Schapka



Zelle 18 von GEW (103-110) CCD-Barcodes Schapka



Zelle 19 von GEW (111-117) CCD-Barcodes Schapka



Zelle 20 von GEW (118-124) CCD-Barcodes Schapka



Zelle 21 von GEW (125-132) CCD-Barcodes Schapka



Zelle 22 von GEW (133-139) CCD-Barcodes Schapka



Zelle 23 von GEW (140-146) CCD-Barcodes Schapka



Zelle 24 von GEW (147-153) CCD-Barcodes Schapka



Zelle 25 von GEW (154-158) CCD-Barcodes Schapka



Zelle 26 von GEW (159-166) CCD-Barcodes Schapka



Zelle 27 von GEW (167-175) CCD-Barcodes Schapka



Zelle 28 von GEW (176-183) CCD-Barcodes Schapka



Zelle 29 von GEW (184-193) CCD-Barcodes Schapka



Zelle 30 von GEW (194-202) CCD-Barcodes Schapka



Zelle 31 von GEW (203-213) CCD-Barcodes Schapka



Zelle 32 von GEW (214-224) CCD-Barcodes Schapka



GEW

Zelle 1 von GEW (1-5) CCD-Barcodes Schapka



Zelle 2 von GEW (6-12) CCD-Barcodes Schapka



Zelle 3 von GEW (13-17) CCD-Barcodes Schapka



Zelle 4 von GEW (18-21) CCD-Barcodes Schapka



Zelle 5 von GEW (22-22) CCD-Barcodes Schapka



Zelle 6 von GEW (23-31) CCD-Barcodes Schapka



BARCODES

Zeile 33 von GEW (225-226) CCD-Barcodes Schapka



TST

Zeile 1 von TST (1-5) CCD-Barcodes Schapka



Zeile 2 von TST (6-6) CCD-Barcodes Schapka



Zeile 3 von TST (7-13) CCD-Barcodes Schapka



Zeile 4 von TST (14-19) CCD-Barcodes Schapka



Zeile 5 von TST (20-23) CCD-Barcodes Schapka



Zeile 6 von TST (24-29) CCD-Barcodes Schapka



Zeile 7 von TST (30-34) CCD-Barcodes Schapka



Zeile 8 von TST (35-38) CCD-Barcodes Schapka



Zeile 9 von TST (39-43) CCD-Barcodes Schapka



Zeile 10 von TST (44-47) CCD-Barcodes Schapka



Zeile 11 von TST (48-52) CCD-Barcodes Schapka



Zeile 12 von TST (53-60) CCD-Barcodes Schapka



Zeile 13 von TST (61-65) CCD-Barcodes Schapka



Zeile 14 von TST (66-72) CCD-Barcodes Schapka



Zeile 15 von TST (73-72) CCD-Barcodes Schapka



EDFL

Zeile 1 von EDFL (1-3) CCD-Barcodes Andreas Huemer



Zeile 2 von EDFL (4-8) CCD-Barcodes Andreas Huemer



Zeile 3 von EDFL (9-13) CCD-Barcodes Andreas Huemer



Zeile 4 von EDFL (14-17) CCD-Barcodes Andreas Huemer



Zeile 5 von EDFL (18-27) CCD-Barcodes Andreas Huemer



Zeile 6 von EDFL (28-33) CCD-Barcodes Andreas Huemer



Zeile 7 von EDFL (34-38) CCD-Barcodes Andreas Huemer



Zeile 8 von EDFL (39-41) CCD-Barcodes Andreas Huemer



Zeile 9 von EDFL (42-47) CCD-Barcodes Andreas Huemer



Zeile 10 von EDFL (48-50) CCD-Barcodes Andreas Huemer



Zeile 11 von EDFL (51-57) CCD-Barcodes Andreas Huemer



Zeile 12 von EDFL (58-58) CCD-Barcodes Andreas Huemer



Zeile 13 von EDFL (59-64) CCD-Barcodes Andreas Huemer



LIM

Zeile 1 von LIM (1-4) CCD-Barcodes Michael Schilli



Zeile 2 von LIM (5-11) CCD-Barcodes Michael Schilli



Zeile 3 von LIM (12-22) CCD-Barcodes Michael Schilli



Zeile 4 von LIM (23-30) CCD-Barcodes Michael Schilli



Zeile 5 von LIM (31-30) CCD-Barcodes Michael Schilli



Welthunger. Erntedank.



Die Deutsche Welthungerhilfe unterstützt Selbsthilfe-Projekte von Bauern der Dritten Welt, damit für sie Ernährung aus eigener Kraft möglich wird. Und sie hilft den Bauern, Natur und Umwelt als Lebensgrundlage zu erhalten, damit Entwicklung auch Zukunft hat.



DEUTSCHE WELTHUNGERHILFE
Spendenkonto Sparkasse Bonn: 111

Deutsche Welthungerhilfe · Adenauerallee 134 · 5300 Bonn 1 · Tel.: 02 28 / 22 88 0

SERVICELEISTUNGEN

BEST OF PRISMA

Schutzgebühr: 30,- DM

Nachsendedienst PRISMA

Schutzgebühr: 5,- DM pro Heft für Jahrgänge 1982-1986
10,- DM pro Heft für Jahrgänge ab 1987

Inhaltsverzeichnis PRISMA

Schutzgebühr: 3,- DM in Briefmarken

Programmbibliothek HP71

Die bislang in PRISMA erschienenen Programme können durch Einsenden eines geeigneten Datenträgers (3,5" Diskette, Digitalkassette oder Magnetkarte) und eines SAFU angefordert werden.

MS-DOS Inhaltsverzeichnis

Kann durch das Einsenden einer formatierten 360 kB oder 1,2 MB 5,25"-Diskette oder einer formatierten 720 kB oder 1,44 MB 3,5"-Diskette angefordert werden.

ATARI-Inhaltsverzeichnis

Kann durch das Einsenden einer 3,5"-Diskette + SAFU bei Werner Müller angefordert werden.

UPLE

Das UPLE-Verzeichnis mit der Kurzbeschreibung der einzelnen Programme sowie den Bezugsbedingungen kann gegen Einsendung von DM 10,- in Briefmarken angefordert werden.

Programme aus BEST OF PRISMA

- a) Eine Kopie der Programme von BEST OF PRISMA auf Kassette erfordert das Beilegen einer Leerkassette und eines SAFU.
- b) Für Barcodes von BEST OF PRISMA-Programmen gibt es folgendes Verfahren:
Schickt eine Liste mit den Namen und der Seitenangabe (der Barcodeseiten) an die Clubadresse, pro Barcode-Seite legt bitte 40 Pf., plus 2,40 DM für das Verschicken, in Briefmarken bei.
Die Liste der verfügbaren Programme ist in Heft 3/88 auf der Seite 35 abgedruckt, sie kann gegen einen SAFU angefordert werden.

Der Bezug sämtlicher Clubleistungen erfolgt über die Clubadresse, soweit dies nicht anders angegeben ist, oder telefonisch bei Dieter Wolf:

(069) 76 59 12

Die eventuell anfallenden Unkostenbeiträge können als Verrechnungsscheck beigelegt werden, Bargeld ist aus Sicherheitsgründen nicht zu empfehlen; ist dies nicht der Fall, so wird Rechnung gestellt und ein Überweisungsvordruck mitgesandt, dies macht die Sache natürlich nicht unbedingt einfacher, bzw. schneller.

Formvorschriften für Schreiben an die Clubadresse gibt es keine; das Schreiben kann durchaus handschriftlich verfasst sein, ein normaler Sterblicher sollte es noch lesen können. Vor allem den Absender und die Mitgliedsnummer deutlich schreiben !

(SAFU = Selbst Adressierter Freiumschlag)

CLUBADRESSEN

1. Vorsitzender

Prof. Dr. Wolfgang Fritz (125)
Kronenstraße 34, 7500 Karlsruhe, GEO1:W.FRITZ

2. Vorsitzender

Erich Klee (1170)
Ruhrallee 8, 4300 Essen 1, GEO1:E.KLEE

Schatzmeister / Mitgliederverwaltung

Dieter Wolf (1734)
Pützerstraße 29, 6000 Frankfurt 90, (069) 76 59 12,
GEO1:D.WOLF

1. Beisitzer

Werner Dworak (607)
Allewind 51, 7900 Ulm, (07304) 32 74, GEO1:W.DWORAK

2. Beisitzer / Geowissenschaften

Alf-Norman Tietze (1909)
Thudichumstraße 14, 6000 Frankfurt 90, (069) 78 93 99 5,
GEO1:A.N.TIETZE

MS-DOS Service / Beirat

Alexander Wolf (3303)
Pützerstraße 29, 6000 Frankfurt 90, (069) 76 59 12

ATARI Service / Beirat

Dr. Werner Müller (1865)
Schallstraße 6, 5000 Köln 41, (0221) 40 23 55,
MBK1:W.MUELLER

CP/M-80

Peter-C. Spaeth, Michaeliburgstraße 4, 8000 München 80

Regionalgruppe Berlin

Jörg Warmuth (79), Wartburgstraße 17, 1000 Berlin 62

Regionalgruppe Hamburg

Alfred Czaya (2225)
An der Bahn 1, 2061 Sülfeld, (040) 43 36 68 (Mo.-Do. abends)
Horst Ziegler (1361)
Schüslerweg 18b, 2100 Hamburg 90, (040) 79 05 67 2

Regionalgruppe Karlsruhe / Beirat

Stefan Schwall (1695)
Rappenwörthstraße 42, 7500 Karlsruhe 21, (0721) 57 67 56,
GEO1:S.SCHWALL

Regionalgruppe Köln

Frank Ortmann (1089), Okerstraße 24, 5090 Leverkusen 1

Regionalgruppe München / Beirat

Victor Lecoq (2246)
Seumestraße 8, 8000 München 70, (089) 78 93 79

Regionalgruppe Rhein-Main

Andreas Eschmann (2289)
Lahnstraße 2, 6906 Raunheim, (06142) 46 64 2

Beirat

Manfred Hammer (2742), Oranienstraße 42, 6200 Wiesbaden

Beirat

Peter Kemmerling (2466), Danziger Straße 17, 4030 Ratingen

Beirat

Martin Meyer (1000), Robert-Stolz-Straße 5, 6232 Bad Soden 1

E-Technik

Werner Meschede (2670), Sorpestraße 4, 5788 Siedlingshausen

Grabau GR7 Interface

Holger von Stillfried (2641)
Am Langdick 13, 2000 Hamburg 61

Hardware 41

Winfried Maschke (413)
Ursulakloster 4, 5000 Köln 1, (0221) 13 12 97

Mathematik

Andreas Wolpers (349), Steinstraße 15, 7500 Karlsruhe

Naturwissenschaften

Thor Gehrmann (3423)
Hobeuken 18, 4322 Spockhövel 2, (02339) 39 63

Serie 80

Klaus Kaiser (1661)
Mainzer Landstr. 561, 6230 Frankfurt 80, (069) 39 78 52

Vermessungswesen

Ulrich Kulle (2719), Memeler Straße 26, 3000 Hannover 51,
(0511) 604 27 28

Programmbibliothek HP71

Henry Schimmer (786), Homburger Landstr. 561,
6000 Frankfurt 50

"Clubadresse"

CCD e.V., Postf. 11 04 11, 6000 Frankfurt 1, (069) 76 59 12

Anwendungsbereich Naturwissenschaften und Technische Wissenschaften

Hallo CCD'ler, einige von Euch haben vielleicht schon bemerkt, daß es im CCD eine neue Fachgruppe gibt. Da die Resonanz bis jetzt nicht gerade überwältigend war (eine Zuschrift!), möchte ich mich an dieser Stelle kurz vorstellen und beschreiben, was ich mir für die Zukunft vorstelle.

Dieses neue "Referat" soll sich zu einem breiten Forum für technisch-wissenschaftliche EDV-Anwendungen (HP-Taschenrechner, MS-DOS) entwickeln. Ich stelle mir vor, daß folgende Punkte interessant sind:

1. rechnergestützte Meßwertaufnahme und Meßwertverarbeitung (Hard- und Software).
2. Besprechung und Tips zu Software aus dem wissenschaftlichen Bereich z.B. Statistik, graphische Präsentation von Meßwerten, technisch-wissenschaftliche Textverarbeitung, Literaturdatenbank und Recherche, Datenbanken, CAD, Compiler, Expertensysteme und was Euch noch einfällt.

3. Aufbau einer Programmbibliothek mit Programmen für Naturwissenschaftler und Techniker. Die Programme brauchen keine Programmierwunderwerke zu sein, sie sollten lediglich funktionieren. Was mir wichtig erscheint, ist aber eine vollständige Dokumentation mit genauer Angabe verwendeter Gleichungen, theoretischer Grundlagen, Leistungsgrenzen und Werten für einen Probelauf. Für den Vertrieb erwarte ich Anregungen von Euch z.B. eigene INFO-Disketten, Weitergabe an den MS-DOS Service.

4. Schaffung persönlicher Kontakte zwischen Mitgliedern. Ich gehe optimistisch davon aus, daß zu den oben aufgeführten Punkten ein Potential von Erfahrung im CCD besteht. Daher fände ich es schön, wenn sich auch Kontakte innerhalb des CCD entwickeln. Ich denke an eine Mitgliedsliste mit Angaben von Adresse, (Telefon), Arbeitsgebiet und verwendeter Hardware/Meßgeräten/Software, die an geeigneter Stelle veröffentlicht wird. Hierdurch ließe sich ein breiter Erfah-

rungsaustausch erzielen, der verhindert, daß Rad mehrmals zu erfinden. Auch hierzu sind Meinungsäußerungen erwünscht.

Jetzt kurz zu mir: Ich studiere an der Ruhr-Universität Bochum Biologie und stehe kurz vor dem Diplom. Erreichen könnt Ihr mich unter folgender Anschrift:

Thor Gehrman
Hobeuken 18
4322 Sprockhövel 2

Tel.: 02339/3963
(wochentags 19.00-22.00 Uhr)

Diese Adresse findet sich übrigens auch in jeder PRISMA. Ich hoffe, daß diese Aktion regen Zuspruch findet und warte gespannt auf Zuschriften. Ihr könnt auch 5.25" Disketten mit 360 KB oder 1.2 MB schicken. Bitte vergeßt das Rückporto nicht, da ich als Student knapp bei Kasse bin.

Thor Gehrman

Fortsetzung von Seite 37

lich nur verbrennen konnte, er war mehr unvollständig als brauchbar. Daß dies in Feierabendarbeit eine Weile dauern kann wird man feststellen, wenn man sich das Inhaltsverzeichnis der Bibliothek schicken läßt, das Verfahren findet Ihr unter den **SERVICELLEISTUNGEN** hinten bei den **CLUBADRESSEN** unter dem Stichpunkt **UPLE**.

Kurz und bündig, diese Liste enthält auf knapp 20 Seiten mehr als 1000, in Worten **tausend** Programmlösungen mit Dokumentation, die von allen Mitgliedern gegen einen geringen Unkostenbeitrag jederzeit abrufbar sind. Eine Liste der in der **UPLE** enthaltenen **Kategorien** von Lösungen findet

Ihr im **PRISMA 3/89** auf der **Seite 26**. Inzwischen haben wir allerdings festgestellt, daß bei ein oder zwei Kategorien keine Programme vorhanden sind, z.B. Kategorie 921 Schiffstabilität.

Ansonsten kann man aber reichlich wühlen, zu allererst sollte man sich allerdings das Verzeichnis der **UPLE** schicken lassen, es ist bei der Mitgliederverwaltung gegen 10.-DM in Briefmarken für die Kopiererei und den Versand zu bekommen.

mm



PostScript für HP-LaserJet II

Für den HP-LaserJet II gibt es jetzt eine Einsteckkassette, die dem Drucker Verständnis für die genormte Seitenbeschreibungssprache PostScript verleiht.

Voraussetzung ist lediglich ein RAM-Ausbau auf mindestens 2 Megabyte.

Zur Verfügung stehen dann 35 Schriften, die auch beliebig in der Größe skalierbar sind, sowie beliebig gedreht werden können.

Die Einsteckkarte nennt sich PacificPage und wurde von der Firma AMS entwickelt.

Die Einsteckkarte paßt nicht in den LaserJet IIP, eine extra Karte für diesen ist aber bereits in Vorbereitung.

mm