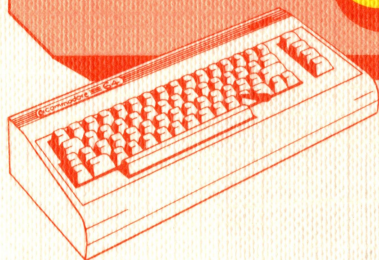


Vera F. Birkenbihl

**EINSTIEG IN  
SIMON'S BASIC  
FÜR DEN  
COMMODORE  
64**



Schwerpunkt  
Grafik

Programme auf Diskette erhältlich

**iWT**







Vera F. Birkenbihl

**EINSTIEG IN  
SIMON'S BASIC  
FÜR DEN  
COMMODORE  
64**



Schwerpunkt  
Grafik

**iWT**



---

CIP-Kurztitelaufnahme der Deutschen Bibliothek

**Birkenbihl, Vera F.:**

Einstieg in Simon's BASIC für den Commodore 64/

Vera F. Birkenbihl. – Vaterstetten: IWT

Schwerpunkt Grafik. – 1984.

ISBN 3-88322-056-6

ISBN 3-88322-056-6

1. Auflage 1984

Alle Rechte, auch die der Übersetzung, vorbehalten. Kein Teil des Werkes darf in irgendeiner Form (Druck, Fotokopie, Mikrofilm oder einem anderen Verfahren) ohne schriftliche Genehmigung des Verlages reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden. Der Verlag übernimmt keine Gewähr für die Funktion einzelner Programme oder von Teilen derselben. Insbesondere übernimmt er keinerlei Haftung für eventuelle, aus dem Gebrauch resultierende, Folgeschäden.

CBM ist ein Warenzeichen der Commodore Business Machine Inc. USA.

Printed in Western Germany

© Copyright 1984 by IWT Verlag GmbH

Vaterstetten bei München

Produktion und Umschlaggestaltung: Kaselow+Partner, München

Satz: Köhler Fotosatz, München

Reproduktionen: Reprodienst B. Steinger GmbH, München

Druck: Freiburger Graphische Betriebe, Freiburg i. Br.

---



---

## UEBERSICHT: AUFBAU DIESES BUCHES:

**KAP.1-2:** GRUNDLAGEN ZUR GRAFIK MIT SIMON'S BASIC (HIRES/MULTI).  
**KAP.3:** JEDER BEFEHL WIRD ERKLAERT.  
**KAP.4:** ANALOG ZU KAP.3: ZU JEDEM BEFEHL EIN LISTING + HARDCOPY, WOBEI JEDE MINI-DEMO EINE QUIZ-AUFGABE STELLT. (LOESUNGEN/ANHANG). PLUS EIN RIESEN-PROGRAMM MIT WEITEREN DEMONSTRATIONEN. DIESES PROGRAMM WIRKT WIE EIN TRICKFILM!  
**KAP.5:** SIMON'S BASIC-GRAFIKEN AUF PAPIER (GAR NICHT SO LEICHT!)  
**KAP.6:** TIPS+TRICKS ZU WEITEREN BEFEHLEN (RENUMBER, MERGE, USW).  
**PLUS: WICHTIGE INFO IM ANHANG!!!**



---

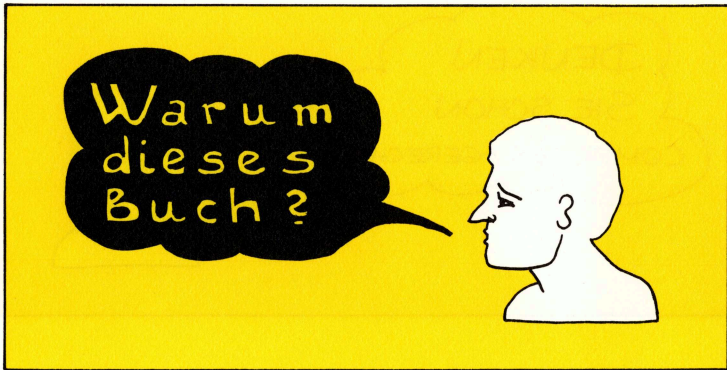
# Inhaltsverzeichnis

Seiten

Einführung . . . . .	3-11
<b>Kapitelchen 1:</b> Ein wenig Terminologie . . . . .	13-23
<b>Kapitelchen 2:</b> Farbe und Modus . . . . .	25-37
<b>Kapitelchen 3:</b> Grafik-Befehle . . . . .	39-125
1. REC . . . . .	46-53
2. BLOCK . . . . .	54-58
3. LINE . . . . .	59-62
4. PAINT . . . . .	63-65
5. TEXT . . . . .	66-76
6. CIRCLE . . . . .	77-90
7. ARC . . . . .	91-103
8. ANGL . . . . .	104-107
9. PLOT . . . . .	108-109
10. TEST . . . . .	110
11. CHAR . . . . .	111-115
12. DRAW / 13. ROT . . . . .	116-125
<b>Kapitelchen 4:</b> Beispiel-Programme . . . . .	127-158
Teil I: Mini-Listings plus Quiz zu jedem Befehl . . . . .	128-151
Teil II: Ein DEMO-Programm . . . . .	152-158
<b>Kapitelchen 5:</b> <i>sibas</i> auf Papier . . . . .	159-164
1. Commodore-Drucker . . . . .	160-163
2. EPSON . . . . .	163-164
<b>Kapitelchen 6:</b> <i>sibas</i> : Tips + Tricks . . . . .	165-178
1. Funktionstasten . . . . .	166-168
2. Renumber . . . . .	168-170
3. FIND . . . . .	170-173
4. PAUSE . . . . .	174-175
5. AUTO . . . . .	175
6. MERGE . . . . .	176
7. DIR . . . . .	177
8. DISK . . . . .	178
<b>Anhang</b> . . . . .	179-201
A: Lösungen (zu Kap. 4) . . . . .	180-190
B: Panda-Listing (zu Kap. 3) . . . . .	191
C: POKE-Werte im Block (Tabelle) . . . . .	192-197
D: Literatur- und Stichwortverzeichnis . . . . .	198-201
E: Karten (befinden sich am Ende des Buches)	
Karte 1: Die Farben des Commodore 64 (unabhängig von Simon's BASIC)	
Karte 2: Farb- und Grafik-Modus-Befehle des Simon's BASIC	
Karte 3: Alle HIRES/MULTI-Befehle: Formate	
Karte 4: Tabelle – Zeichentyp (HIRES + MULTI)	
Karte 5: DRAW + ROT – Übersichten	
Karte 6: Figuren per ARC-Befehl	

---

Dieses Buch soll *kein Ersatz* für die offizielle Simon's BASIC-Anleitung sein, sondern stellt eine *Ergänzung* dar.



Sie sind stolzer Besitzer des C 64. Sie wissen, daß dieses Gerät, von der Hardware her, ungeheure Möglichkeiten bietet. Aber Sie wissen auch, daß das CBM-BASIC diese Möglichkeiten überhaupt nicht ausschöpft. Also haben Sie sich SIMON's BASIC dazugekauft (oder stehen kurz vor der Anschaffung).

SIMON's BASIC stellt eine sogenannte Befehls-Erweiterung dar, d. h. es bietet Ihnen ZUSÄTZLICHE BASIC-Befehle. Dieses Programm wurde von einem (damals) 16 Jahre jungen Engländer geschrieben. Nun können Sie sich denken, daß so ein »Wunderkind«, das im zartesten Alter bereits fachmännisch Maschinensprache-Programme schreiben kann, wohl kaum viel Verständnis für Normal-Anwender hat, die froh sind, wenn Sie in BASIC ein bißchen programmieren können! Dementsprechend ist auch die Anleitung ausgefallen:



---

ERSTENS: Sie ist im Telegramm-Stil abgefaßt worden; für Einsteiger nicht gerade optimal.

ZWEITENS: Die Parameter-Angaben sind in der deutschen Ausgabe fast alle vom Englischen übernommen worden. Dies aber ist unnötig, denn sie stellen ja schließlich nur Symbole für Buchstaben oder Zahlenwerte dar. Warum also muß »Abstand« mit einem »i« (für INTERVALL) geschrieben werden, wenn an der Stelle genausogut »A« oder »AB« stehen könnte?

DRITTENS: Einige Beispiel-Programme sind in FOR...NEXT-Schleifen so kunstvoll versteckt worden, oder enthalten statt einfacher Parameter-Werte mathematische Formeln, so daß man sich fragt, wie der Befehl, um den es gerade geht, wirklich »funktioniert«.

VIERTENS und letztens: Manche Befehle sind unlogisch aufgebaut bzw. bei manchen ist sogar die Format-Angabe im Handbuch falsch. Dies verursacht insbesondere bei (Noch-)Nicht-Profis völlig unnötigen Frust!



Es gilt einmal über folgende Frage nachzudenken:

Wenn die Computer- und Software-Produzenten fordern, der Anwender müsse COMPUTER-GERECHT vorgehen, warum machen sie sich dann nicht mehr Gedanken über eine **GEHIRN-GERECHTE** Anleitung zu ihren Produkten?!

Was heißt das?

Wenn Sie einen Computer oder ein Programm anwenden wollen, müssen zuerst zwei Faktoren gewährleistet sein:

1. Sie müssen die Anweisung **verstehen**, und
2. Sie müssen sie **lernen** (wenn Sie nicht ewig im Handbuch nachschlagen wollen)!

---

Wenn also Software- und Handbuch-Autoren ein wenig mehr Rücksicht auf die ARBEITSWEISE DES GEHIRNS nehmen würden, dann könnte der Anwender

1. leicht und mühelos begreifen, und
2. fast nebenbei ins Gedächtnis einspeichern, was er nun wie machen muß!

(Falls Sie sich für Details über GEHIRN-GERECHTES ARBEITEN interessieren, so sei hier auf mein Buch *Stroh im Kopf? Oder: Gebrauchsanweisung für's Gehirn* verwiesen.)

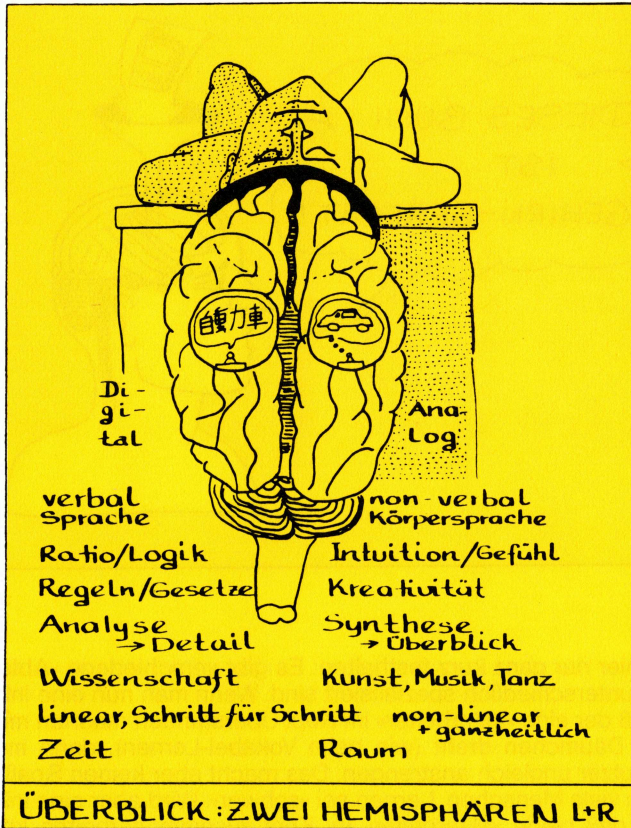


Wollen wir hier nur ganz kurz festhalten: Es gibt verschiedene »Abteilungen« im Gehirn, die unterschiedlich spezialisiert sind. Wenn man nun eine Information so anbietet, daß der eine »Mitarbeiter« im Kopf Überstunden machen muß, während der andere Däumchen dreht (wie beim Vokabel-Lernen), dann muß sich der Gehirn-Benutzer ungleich anstrengen. Das macht aber keinen Spaß mehr! Deshalb meinen viele Menschen, Lernen sei »schwer«. Wird aber eine Information so angeboten, daß beide »Mitarbeiter« im Kopf zusammenarbeiten können, dann kann man sich sofort »ein Bild« machen, man be-grieft (praktisch) worum es geht oder wie etwas funktioniert. Aber nicht nur das Verstehen ist viel leichter geworden: Was man einmal wirklich be-griffen hat, oder was einem mit einem »Aha!« plötzlich klar geworden ist, das hat man schon zu 80% im Gedächtnis drin. Der eigentliche Lernvorgang (im oben beschriebenen Sinn) ist UNNÖTIG geworden.

Welcher »Mitarbeiter« im Kopf macht nun was?

Herr LINKS (in der linken Großhirn-Hälfte) ist zuständig für SPRACHE. Also für das Denken, Sprechen, Lesen und Schreiben von WORTEN. Er denkt analytisch, kann (be-)rechnen, detailliert arbeiten und ist für Ihr Zeitgefühl zuständig.

Herr RECHTS (in der rechten Großhirn-Hälfte) hingegen ist STUMM. Er denkt auch, aber NICHT MIT WORTEN. Er denkt in BILDERN. Er ist für jedes klare Fallbeispiel dankbar. Er will praktisch ausprobieren und experimentieren, und zwar mit der Neugierde und Faszination eines Kindes. (Wenn Herr LINKS hingegen experimentiert, dann will er vorher schon wissen, was bei dem Versuch herauskommt!)



Diese Abbildung wurde mit freundlicher Genehmigung entnommen aus: *Stroh im Kopf. Oder: Gebrauchsanweisung für's Gehirn.*



---

Dieser »Arbeitsteilung« im Gehirn trägt dieses Buch Rechnung: Wenn Sie eine Erklärung LESEN, dann ist zunächst Herr LINKS dran. Wenn Sie sich aber ein BILD machen können, dann hat Herr RECHTS mitgearbeitet. Diese Tatsache wird wie folgt ausgenützt:

## 1

**Dieses Buch enthält** zahlreiche Abbildungen, damit Herr RECHTS immer gleich mit-informiert wird.

## 2

**Dieses Buch enthält** GEHIRN-GERECHTE Beispiel-Programme. So bewirken diese z. B. oft, daß ein Text am Bildschirm für Herrn LINKS erscheint, während Herr RECHTS gleich beobachten kann, was dieser Befehl be-WIRKT (praktisch nämlich). Übrigens gibt's zu dem Buch eine Diskette, so daß Sie sich das Eintippen der Programme sparen können.

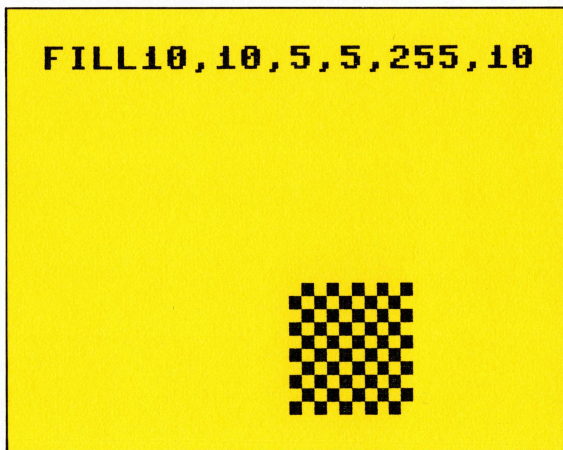
## 3

**Dieses Buch enthält** zahlreiche Bildschirmausdrucke, damit Herr RECHTS genau das richtige Bild zu einer Information für Herrn LINKS erhält.

## 4

**Dieses Buch enthält** einen Ausdruck sämtlicher POKE-Werte (von 0-255), jeweils in einem kleinen Block angeordnet, damit Sie eine BILDLICHE NACHSCHLAGE-Liste haben, wenn Sie einen bestimmten Effekt zu erreichen suchen. Denn: Die Liste in Ihrem C 64-Handbuch auf Seite 133f zeigt erstens nur je ein Symbol und zweitens die Invertierten gar nicht. Ein Symbol wirkt aber total anders, wenn es in einem Block erscheint.

Beispiel:



(Die gesamte Liste beginnt auf Seite 192)

---

## 5

**Dieses Buch enthält** ÜBERSICHTS-KARTEN. Diese befinden sich am Ende des Buches und sollten herausgetrennt werden. Vielleicht stecken Sie sie in eine Klarsicht-Hülle? Diese Karten sollen als Nachschlag-Karten dienen, wenn Sie am Computer sitzen. Sie enthalten:

KARTE 1: Die Farben des Commodore 64 (unabhängig von Simon's BASIC!)

KARTE 2: FARB- und GRAFIK-MODUS-Befehle des Simon's BASIC

KARTE 3: Alle HIRES/MULTI-Befehle:Formate

KARTE 4: TABELLE – ZEICHENTYP (HIRES + MULTI)

KARTE 5: DRAW + ROT – Übersichten

KARTE 6: Figuren per ARC-BEFEHL.

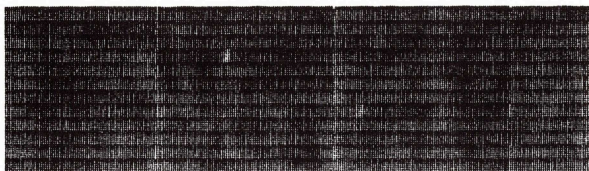
## 6

**Dieses Buch enthält** auch kleine EXPERIMENTE, anhand derer Sie sich selber eine Erfahrung vermitteln. Das spart lange Textseiten für Herrn LINKS. Außerdem merken Sie sich die Ergebnisse sofort, da Sie diese selbst erarbeitet haben.

---

Durch diese Vorgehensweise wird es Ihnen Spaß machen, sich in die neuen Befehle einzuarbeiten. Geistige Arbeit macht nämlich Spaß – wenn sie GEHIRN-GERECHT betrieben wird! Daher kann ich allen Ernstes sagen:

VIEL SPASS WUENSCH  
ICH IHNEN MIT...



Übrigens: Da es etwas umständlich ist, immer »Simon's BASIC« zu sagen, werden wir ab jetzt etwas verkürzen:

VIEL SPASS WUENSCH  
ICH IHNEN MIT...









Wie Sie wissen, spricht der eine Autor von der »Bildschirmfarbe« (im Gegensatz zur Rahmenfarbe) während der andere statt »Bildschirmfarbe« immer nur »HG« (für Hintergrund) sagt. Deshalb soll hier festgehalten werden, was wir in diesem Buch meinen, wenn wir bestimmte Begriffe verwenden. Außerdem möchte ich hier bereits auf einige Eigenarten hinweisen. Sie zu kennen heißt, gewisse Effekte verstehen, die durch die »Bauweise« des C 64 ausgelöst werden.

RA  
= Rahmen



niedrige Auflösung

**LORES**OLUTION

Der LORES-Bildschirm wird auch "normal"

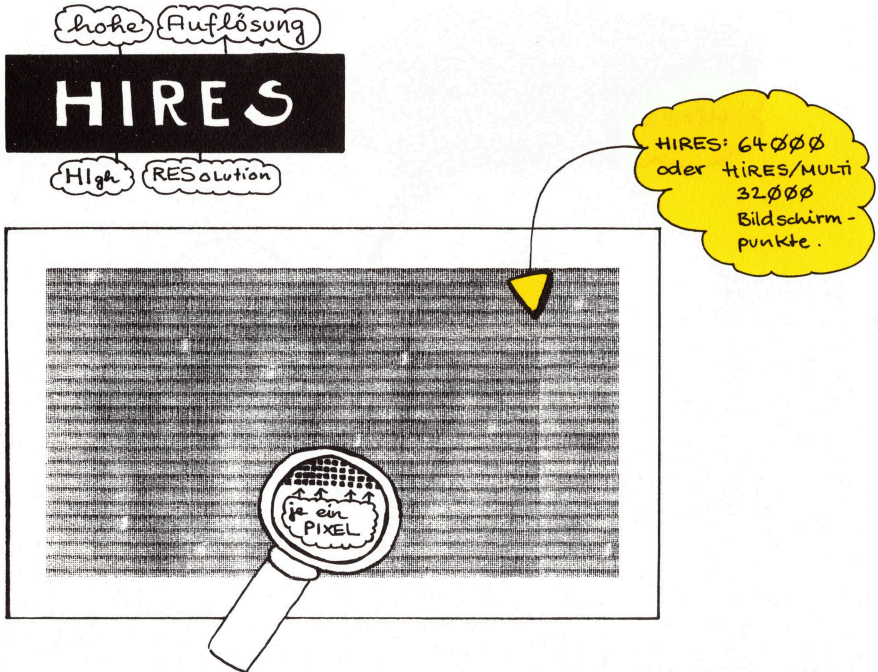
**NRM**

oder TEXT-Modus genannt

1000  
BLOCKS à  
8\*8 Pixels







Dieser Begriff ist etwas unklar, da er zwei verschiedene Bedeutungen hat, und da er mal dies mal das bedeutend, angewendet wird:

### 3.1 HIRES im Gegensatz zu LO RES


Hier bezeichnet man mit HIRES einfach den Grafik-Modus, bei dem man jeden Punkt einzeln ansteuern kann (während wir bei LO RES Zeilen und Spalten zählen, also den Bildschirm in 25\*40 kleine »Blocks« unterteilen.)

### 3.2 HIRES im Gegensatz zu MULTI (siehe Seite 17)

Dieses HIRES ist, genaugenommen, eine Untermenge des oben beschriebenen HIRES-Modus: Es beschreibt den spezifischen hochauflösenden Modus, in dem wir 320\*200 Punkte einzeln ansteuern können. Hier heißt es immer, wir hätten nur eine Zeichenfarbe (zf) und eine Hintergrundfarbe (hg) zur Verfügung. Aber das stimmt nicht ganz, denn auch in diesem HIRES-Modus kann via *sibas*-Befehle die Zeichenfarbe geändert werden.



---

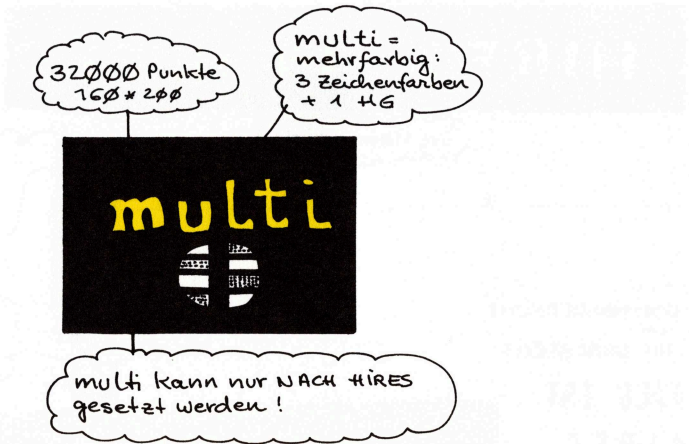


Der C64 hat einige  
komische Eigenarten im  
Hires-Modus



Wiewohl man ja im HIRES jeden Punkt einzeln ansteuern kann, bezieht sich diese Aussage nur auf das SETZEN des Punktes, nicht aber auf seine Farbgebung. Diese wird nämlich, um Speicherplatz zu sparen, genau wie im LO RES-Modus gehandhabt. Daraus ergeben sich drei Konsequenzen für den Programmierer:

1. Wenn Sie mehrfarbig arbeiten, müssen Sie davon ausgehen, daß Sie jeweils nur einen 8\*8-Pixel-Block farblich beherrschen. Kommen sich Punkte oder Linien zu nahe, dann werden sie sich farblich gegenseitig beeinflussen. Das *Trainingsbuch zum SIMON's BASIC* von Data Becker erklärt diese Zusammenhänge sehr detailliert. Im allgemeinen ist es nicht »schlimm«, man muß es nur wissen, um zu begreifen, daß dies »normal« ist.
2. Selbst wenn Sie nur mit einer Zeichenfarbe arbeiten, ist die Farbgebung im Detail »unsauber«. Viele Linien oder Punkte nahe beieinander werden immer »bunt« wirken, wiewohl Sie nur weiß oder rot gezeichnet haben. Dieser Effekt kann sogar ausgesprochen schön wirken, aber bei geometrischen Zeichnungen ist er eher etwas störend.
3. Bei diesem HIRES-Modus von 320\*200 Bildschirm-Punkten sind Linien unterschiedlicher Farbe auch unterschiedlich dick! Hellrot, zum Beispiel, ergibt weit dünnere Linien als weiß, grau3 (= silbergrau) wirkt zwar genauso hell wie weiß, ergibt jedoch weit dünnere Linien! Allerdings nur auf dem Bildschirm, bei einem Hardcopy (Bildschirmausdruck auf Papier) merken Sie nichts davon!



Mit MULTI ist jetzt der zweite HIRES-Modus gemeint: Auch hier steuern wir einzelne Bildschirm-Punkte, aber MULTI unterscheidet sich von HIRES (320\*200 Pixels) in folgender Weise: Jetzt können wir pro 8\*8-Pixel-Block je drei Zeichenfarben und eine Hintergrundfarbe bestimmen. Dafür aber verlieren wir einen Teil der hohen Auflösung. Dies hat natürlich wieder mit dem Speicherplatz zu tun. Nun hat Commodore sich entschlossen, uns auf der waagerechten x-Achse weniger ansteuerbare Punkte zu geben. Resultat: Jeder Bildschirmpunkt ist genau DOPPELT SO BREIT wie in HIRES, so daß wir jetzt auf 160\*200 Punkte kommen. Da jeder Punkt bei Multi 2 Pixel breit ist, schlage ich vor, bei HIRES von »Pixeln« zu sprechen, bei MULTI hingegen von »Punkten«.

# HIRES + MULTI

im Vergleich

320 WAAGERECHT  
200 SENKRECHT  
DIES IST  
HIRES  
64000  
PIXELS

320 Pixel waagerecht (X-Achse)  
160 Punkte waagerecht (X-Achse)

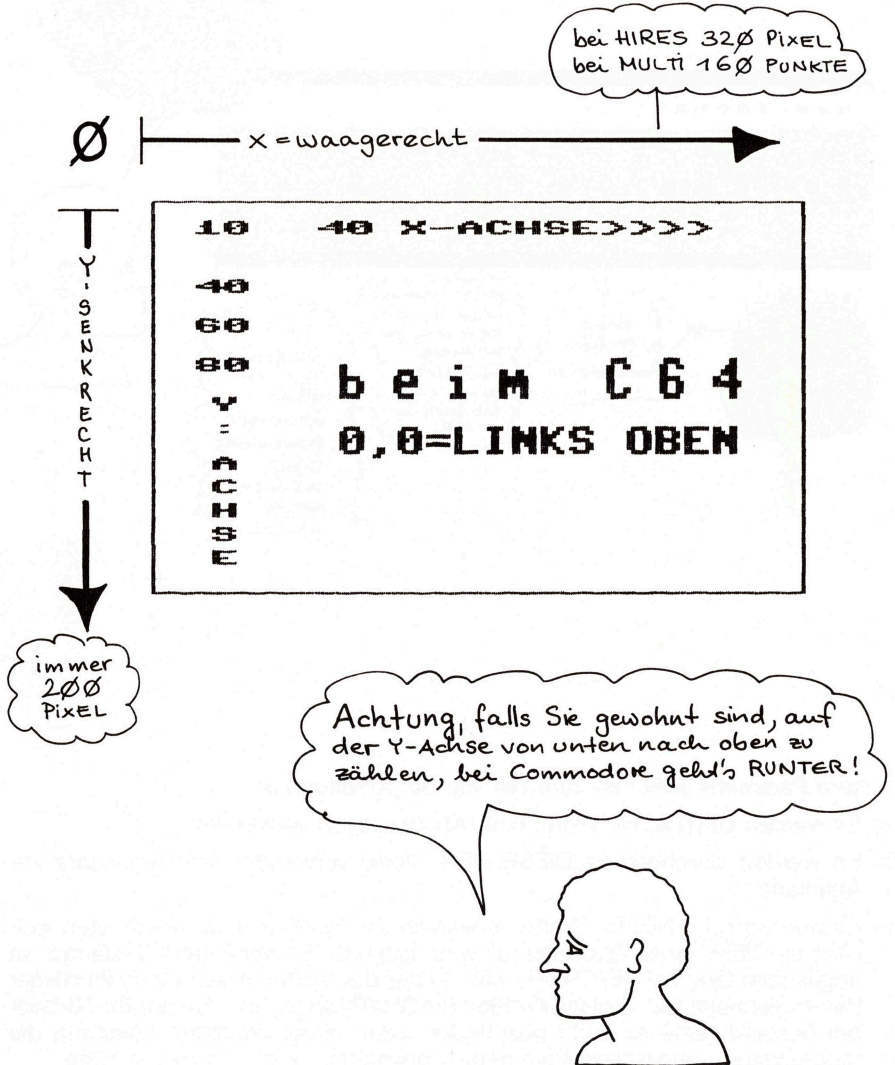
160 WAAGERECHT  
1 PUNKT = 2 PIXELS BREIT  
200 senkrecht  
dies ist jetzt  
hires / multi  
32000 PUNKTE

jeder der 64000 Bildschirm-  
punkte heisst:  
**PIXEL.**  
Da bei MULTI jeder Punkt  
doppelt so breit ist, also  
2 Pixel, nennen wir MULTI-  
punkte jeweils einen  
**PUNKT.**

MULTI ist eine  
Sonderform von  
HIRES

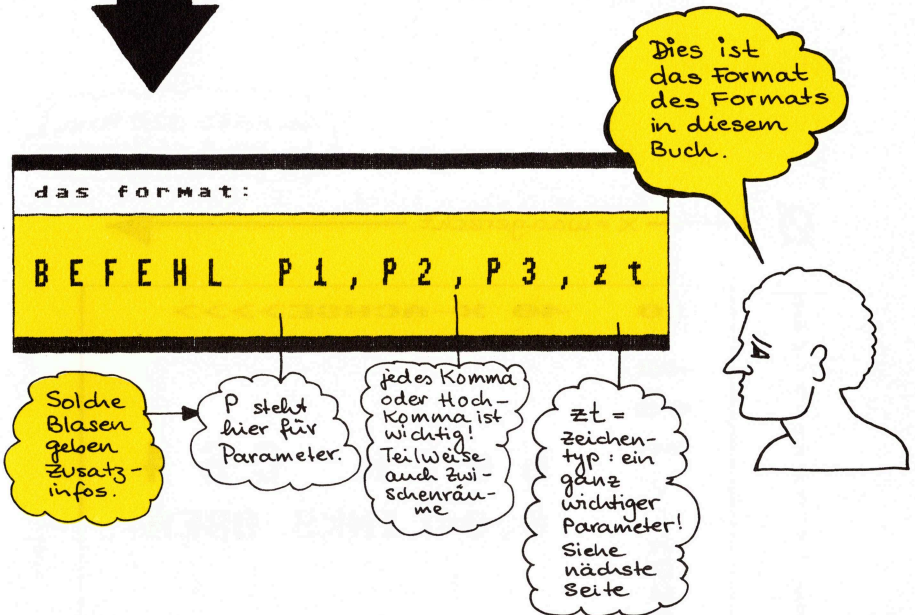


Übrigens: Falls Sie sich mit den Werten auf der x- und y-Achse beim C64 noch nicht näher befaßt haben sollten: Im Gegensatz zu manchen anderen Computern (und Mathematikern) heißt bei Commodore der OBERE LINKE Punkt/Pixel immer 0 (Null). Bei x wird nach RECHTS, bei y hingegen nach UNTEN weitergezählt:





# FORMAT



Unsere Parameter weichen zum Teil von der Anleitung ab:

- a) Es werden DEUTSCHE Worte bzw. Abkürzungen verwendet.
- b) Es werden durchgängig DIESELBEN Worte verwendet (im Gegensatz zur Anleitung)!
- c) Es werden z. T. ANDERE Worte verwendet. So lautet z. B. der Parameter, welcher die TIEFE eines Zeichens (es wird nach UNTEN verlängert) bestimmt, im englischen Original DEPTH (=Tiefe). In der deutschen Anleitung heißt dieser Parameter beim FILL-Befehl »Größe«, bei CHAR hingegen d (für depth). Nebenbei bemerkt hätte es nicht geschadet, wenn diese deutsche Anleitung die (abgekürzten) englischen Parameter wenigstens einmal übersetzt hätte.

---

Ein Zeichen kann auf verschiedene Weise gesetzt werden. Dabei sind 3 Parameter ausschlaggebend:

- a) ZEICHENTYP (zt)
- b) TIEFE (ti)
- c) ABSTAND von anderen Zeichen (ab).

Beginnen wir mit dem



Mit **0** erreichen Sie *sowohl* bei HIRES als auch bei MULTI, daß die Zeichnung bzw. die Zeichen in der Hintergrundfarbe ausgeführt wird/werden. Dies ist nützlich, wenn man in eine durch PAINT (= anmalen) gefüllte Fläche zeichnen/schreiben will. Die Tabelle (auf der nächsten Seite) demonstriert dies!

Mit **1** zeichnen/schreiben Sie immer mit **Zeichenfarbe 1**. Da HIRES nur 1 zf hat, ist es *die* Zeichenfarbe! Mit **2** bei HIRES (**4** bei MULTI) wird **invertiert**: Wo ein Punkt gesetzt ist wird gelöscht, und umgekehrt! Bei HIRES bedeutet dies, daß **0** (HG) und **1** (zf) gegeneinander ausgetauscht werden.

Mit **2** und **3** wählen Sie bei MULTI die Zeichenfarben 2 und 3. Bei MULTI *invertiert* Zeichentyp **4**: Er **vertauscht** die Farben **0** (HG) und **3** (= 3. Zeichenfarbe), da diese beiden im normalen Farbram gespeichert werden. Und er **vertauscht** die Zeichenfarben **1** und **2**, welche beide im Videoram gespeichert werden.

Dieser letztgenannten Tatsache ist folgendes Phänomen zu verdanken: Wenn sie eine HIRES/MULTI Grafik am Bildschirm *hatten* und schalten zurück auf LORES (NRM), dann hat sich die Farbe 3 von MULTI verändert. Denn: der Farbram enthält ja auch die LORES-Farben! Wundern Sie sich also nicht, wenn das passiert.

Die Tabelle auf der nächsten Seite vermittelt die Übersicht für Ihre rechte Hirnhälfte:

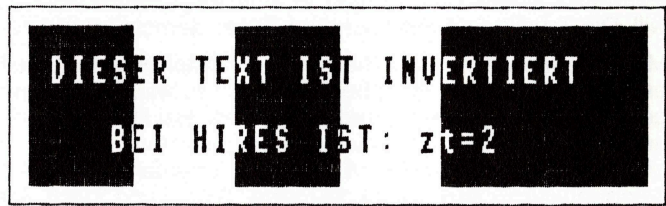
<b>zeichentyp: HIRES+MULTI</b>	
<b>0=IMMER HINTERGRUNDFARBE</b>	
<b>HIRES</b>	<b>MULTI</b>
<b>1: Zeichenfarbe</b>	<b>1: Zch. farbe NR.1</b>
<b>2: invertiert</b>	<b>2: Zch. farbe NR.2</b>
<b>NICHTS</b>	<b>3: Zch. farbe NR.3</b>
<b>NICHTS</b>	<b>4: invertiert: Farben 0 und 3 Farben 1 und 2</b>

Diese Schrift  
zt = 2  
(invertiert  
bei HIRES)

diese Linie:  
zt = 2  
(invertiert  
bei HIRES)

Durch Invertieren können höchst interessante Effekte erzielt werden.

Nochmal: Invertierter Text (zt=2) im HIRES-Modus



Als nächstes wenden wir uns den beiden Parametern TIEFE (ti) und ABSTAND (ab) zu.



# TIEFE + ABSTAND

Bei TIEFE (ti) geht es um die *Größe* von einzelnen Zeichen. In der Anleitung steht als Parameter mal »Größe«, mal d (für depth = Tiefe). Da die Zeichen nach UNTEN vergrößert werden, ist der Begriff TIEFE besser, daher verwenden wir ihn in diesem Buch. Abkürzung: ti.

Bei ABSTAND (ab) geht es um den Abstand zwischen einzelnen Zeichen: Es wird immer von der linken Kante des einen Zeichens zur linken Kante des nächsten Zeichens gerechnet.

**ti**

- 1 = NORMALE "GRÖSSE"
- 2 = doppelt so tief
- 3 = Faktor 3 tief
- etc.

DIESER TEXT: TI=1, AB=7

JETZT: TI=2, AB=14

JETZT: TI=3, AB=7

\* bei MULTI in Punkten

Der AB wird in Pixels\* gemessen. Minimum 7 ergibt, dass sich die Zeichen nicht überschneiden.

JETZT: TI=2, AB=14

JETZT: TI=3, AB=7

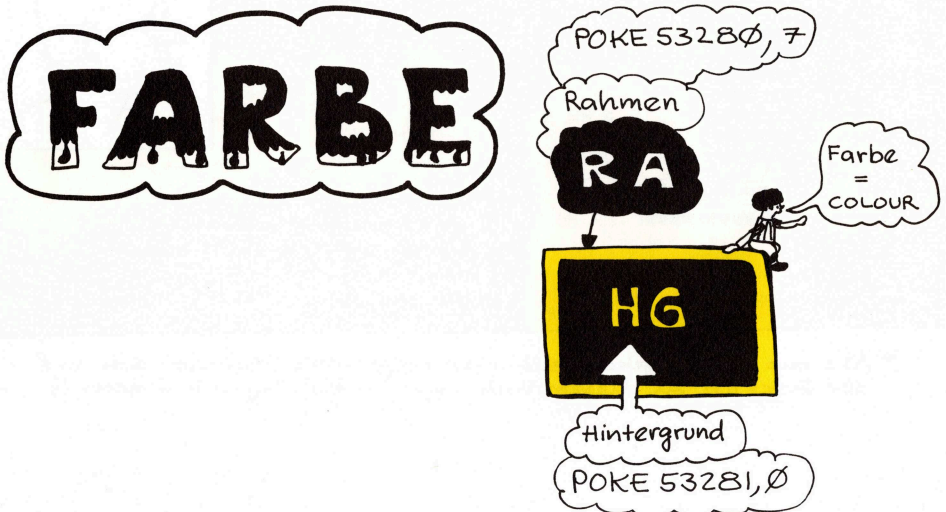
Gerade das Überschneiden kann höchst interessante Effekte bei Grafikzeichen bewirken!





Da manche Befehle die Farbgebung mit der Modus-Wahl eng miteinander verknüpfen, wollen wir die beiden Aspekte auch zusammen besprechen. Mit MODUS ist gemeint, ob Sie in LO RES, HIRES oder MULTI arbeiten wollen.

Beginnen wir mit:



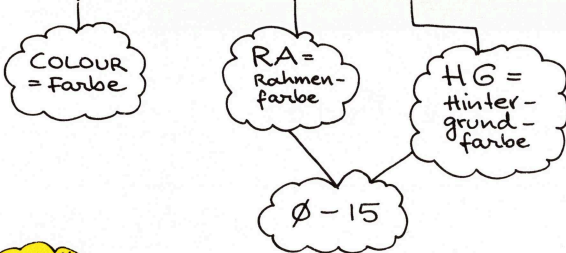
Falls Sie sich in der Vergangenheit geärgert haben, daß RA (Rahmenfarbe) und HG (Hintergrund) nur über mühselige POKES festzulegen waren, dann können Sie jetzt aufatmen: Es gibt nämlich einen *sibas*-Befehl (den Sie in der Anleitung allerdings vergeblich suchen werden), der diese POKES unnötig macht:



```

das format :
COLOUR RA, HG

```



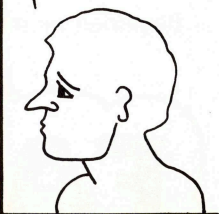
Dieses Experiment sollten Sie unbedingt probieren.

\* Disk

```

10 REM FARBGEBUNG MIT SIBAS : COLOUR
12 REM INT/BIRKENBIHL, C. 1984
14 :
16 PRINTCHR$(147); REM BILDSCHIRM LOESCHEN
18 PRINTAT(5,5) "###COLOUR RA,HG" REM "###"=CBM+8: SILBERGRAU
20 PRINT"#####SIE WERDEN JETZT DIE RAHMENFARBE"
22 PRINT"#####UND DIE HINTERGRUNDFARBE SELBST"
24 PRINT"#####BESTIMMEN, UM DEN COLOUR-BEFEHL"
26 PRINT"#####KENNZULERNEN...": PAUSE3
28 PRINT"#####BITTE RA, KOMMA UND HG"
30 PRINT"#####EINGEBEN! MIT 33,33 BEENDEN SIE"
32 PRINT"#####DIESES PROGRAMM."
34 :
36 : LOOP
38 : INPUT"#####": RA, HG
40 : COLOUR RA, HG: PAUSE1
42 : EXIT IF RA=33
44 : EXIT IF HG=33
46 : END LOOP

```

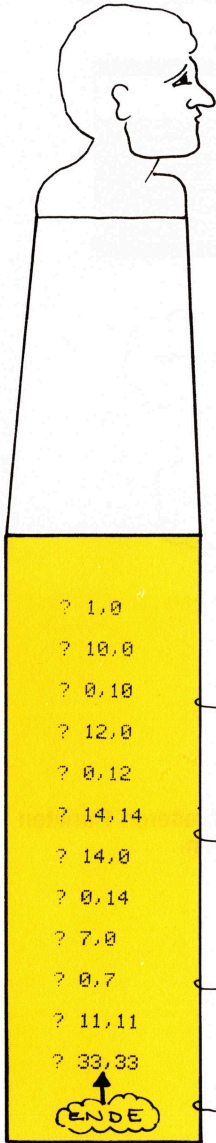


Hier sehen Sie ein Beispiel für strukturiertes Programmieren mit sibas. Siehe S. 58 im sibas-Handbuch.

\* Alle mit Disk gekennzeichneten Programme befinden sich auf der Diskette zu diesem Buch (plus, natürlich, noch anderes !)

Falls Sie jetzt in LO RES weitermachen wollen, brauchen Sie die jeweilige Zeichenfarbe nur in PRINT-Befehle zu »verpacken«, entweder in Form von Farbzeichen innerhalb der Hochkommas (siehe Zeile 18), oder als PRINTCHR\$(...)-Befehle. Damit Ihr »Herr RECHTS« im Gehirn (siehe Intro) glasklar sieht, wie Sie welche Farben erhalten, legen Sie bitte KARTE NR. 1 (am Ende des Buches) griffbereit!

Na, haben Sie das COLOUR-Experiment gleich ausprobiert?  
Hier folgt ein Bildschirmausdruck und ein Probelauf:



zeile 18:  
Reverse Schrift:  
COLOUR RA, HG

SIE WERDEN JETZT DIE RAHMENFARBE  
UND DIE HINTERGRUNDFARBE SELBST  
BESTIMMEN, UM DEN COLOUR-BEFEHL  
KENNENZULERNEN...

BITTE ████, KOMMA UND ████  
EINGEBEN! MIT 33,33 BEENDEN SIE  
DIESES PROGRAMM.

Gerade hier sieht der "Herr RECHTS" (s. INTRO) sofort, wie der Befehl aufgebaut ist.

Ausserdem TRAINIERT dieses Programm, wenn Sie's öfter durchlaufen, Ihr Gedächtnis für die Farb-Werte von 0 (schwarz) bis 15 (grau 3 = silbergrau, fast wie weiss). Dieses TRAINING ist besonders gehirn-gerecht, wenn Sie vor dem Drücken von Return erst vor Ihrem geistigen Auge zu sehen versuchen, was gleich anschliessend zu sehen sein wird. (Auch bei Grafik-Befehlen später ein gehirn-gerechtes Vorgehen. Man lernt viel schneller!)

---

Wenn Sie hingegen in den Grafik-Modus wechseln möchten, müssen Sie nach COLOUR RA, HG jetzt weiterdenken:

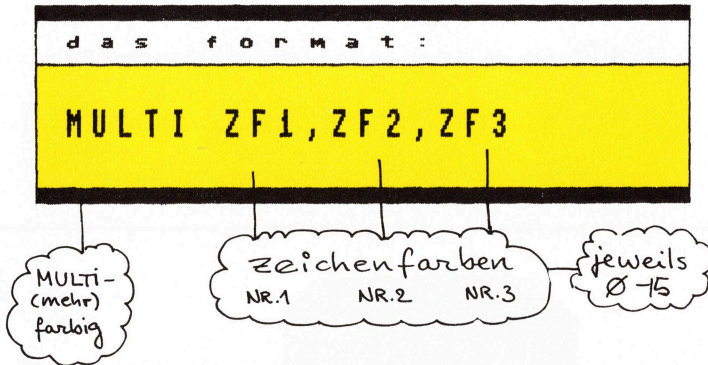
1. Zunächst müssen Sie auf HIRES umschalten. Dieser Befehl ist mit Farbwahl-Angaben verknüpft: Sie müssen nämlich die ZF (Zeichenfarbe) und HG (Hintergrund) bestimmen. Wenn Sie hier einen anderen HG angeben als im vorangegangenen COLOUR-Befehl, so gilt natürlich dieser zuletzt angegebene:



2. Vielleicht wollen Sie jedoch mit drei Zeichenfarben und breiteren Punkten arbeiten? Dann müssen Sie den MULTI-Befehl nachschieben.



Wichtig: HIRES schaltet auf hochauflösende Grafik um. Hier wird auch die Hintergrundfarbe bestimmt. HIRES muß daher immer gesetzt werden, auch wenn Sie anschließend VON HIRES AUF MULTI wechseln wollen. Denn der MULTI-Befehl definiert lediglich die drei Zeichenfarben. Weder kann er die hochauflösende Grafik einschalten, noch enthält er eine Angabe zum HG:



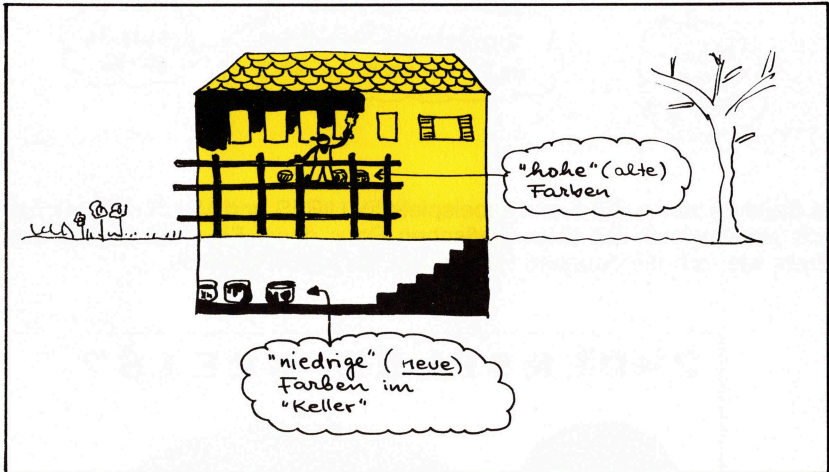
Auf Seite 23 sehen Sie zwei Fallbeispiele für HIRES und MULTI mit Text. Es folgt noch ein Vergleich mit einer grafischen Figur, damit Sie noch einmal deutlich sehen, wie sich die doppelte Punktbreite bei MULTI auswirkt.





---

Im folgenden wird die Möglichkeit zur FARB-ÄNDERUNG besprochen. Sie gilt auch für HIREs, so daß man nicht unbedingt auf MULTI umschalten muß, nur weil man mehr als eine Zeichenfarbe haben will! Denn MULTI bedeutet immer auch eine gröbere Zeichnung, weil ja jeder Bildschirmpunkt 2 Pixels breit ist!



### 3. NEUE FARBEN?

Der Befehl, mit dem man die Zeichenfarbe(n) ändern kann, hat einen sehr eigenartigen Namen. NEW COL(our) wäre »einfach« gewesen, da aber NEW ein BASIC-Wort ist, geht das nicht so einfach. Also mußte man sich etwas anderes einfallen lassen, aber das Resultat ist schon höchst erstaunlich: »Niedrige Farbe« heißt er jetzt. Im Gegensatz zu »hoher Farbe«, welche die ursprünglichen Farben wieder zurückbringt. Damit Sie sich später nicht des öfteren fragen, ob jetzt die neuen Farben »hoch« oder »niedrig« genannt werden, folgende Eselsbrücke: Stellen Sie sich einen Maler vor, der eine Hauswand bemalt. Die Farbtöpfe stehen im Keller: Wenn er also *neue* Farben will, muß er die Leiter hinunter, um die »niedrigen« Farben zu holen; während die bereits benutzten (*alten*) Farben natürlich oben (auf seinem Gerüst) herumstehen!

Jetzt zu den beiden Befehlen selbst:



# ACHTUNG:



# LOW COL + HIRES

Die Befehlskombination LOW COL + HIRES ist so eigenartig, daß wir kurz einen Blick in zwei Texte werfen wollen:

- a) in die offizielle *sibas*-Anleitung (Seiten 27/28) und
- b) in *Das Trainingsbuch zum Simon's BASIC* von Data Becker (Seite 234 ff):



Beginnen wir mit der *sibas*-Anleitung von Commodore: Hier heißt es zunächst nur, daß drei weitere Farben bereitgestellt werden. Aber bei HI COL (Seite 27) steht explizit:

»**Ziel: Zurückholen der unter MULTI festgelegten Farben.**«

Blättern wir noch einmal zurück zu Seite 27, so finden wir trotzdem einen Mini-Hinweis auf die Möglichkeit, LOW COL mit HIRES anzuwenden:

»Achtung: Es müssen immer alle drei Parameter angegeben werden, **auch in HIRES-Grafik**.«

Prima. Also könnte man es ja ausprobieren. . . Allerdings sind die Resultate ganz anders, als man erwartet. Deshalb war ich höchst gespannt, was die Data Becker-Leute wohl zu dieser Frage zu sagen haben: Dieses Buch enthält mehrere Seiten für LOW COL, aber auch hier findet man Verwirrendes:

Auf Seite 214 *oben* heißt es, die **zweite Farbe des LOW COL-Befehls habe »KEINE FUNKTION.**«

Aber schon auf derselben Seite im *unteren* Drittel lesen wir:

»Im HGR (= HIRES) werden von den drei erforderlichen Werten (f=Farbe) f1, f2 und F3 lediglich . . . **die beiden ersten** benötigt, da wir hier ja nur eine Punktfarbe pro 8\*8-Block festlegen können.«

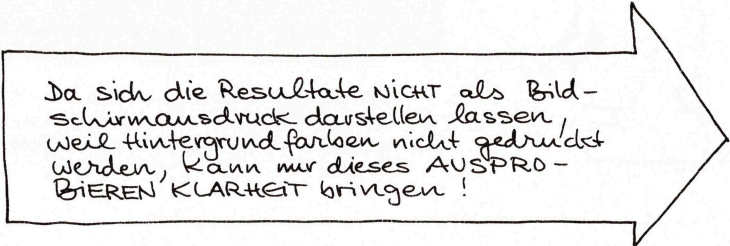
**Hat nun die zweite Farbe eine Funktion oder nicht??** Gleich im Anschluß an dieses Rätsel folgt:

»**Der zweite Wert dient zur Festlegung der Hintergrundfarbe des 8\*8 Feldes.**...«

---

Aha. Also doch! Sie erinnern sich vielleicht an den ersten Hinweis (Seite 16 dieses Buches) auf die eigenartige Tatsache, daß Farbe im hochauflösenden Modus wie bei LO RES gehandhabt wird, um Speicherplatz zu sparen. Also gut, wenn jetzt, wie die Data Becker-Autoren sagen, mit dem zweiten Farb-Parameter die Hintergrundfarbe für ein 8\*8-Pixel-Feld geändert werden kann, dann könnte dies höchst interessante Effekte erzeugen. Allerdings zeigt das praktische Ausprobieren, daß auch dies noch nicht ganz stimmt. Denn:

**ZWAR WIRD DIE HG BEI JEDEM EINZELNEN »DANACH« GESCHRIEBENEN ZEICHEN VERÄNDERT, ABER EBEN NICHT IMMER IN EINEM SAUBEREN 8\*8-BLOCK; INSBESONDERE, WENN DIE BUCHSTABEN (DURCH  $t_i = 2$  ODER 3) VERLÄNGERT WERDEN. OFT IST ES EIN UNREGELMÄSSIGER RAND UM DAS ZEICHEN HERUM.**



Da sich die Resultate NICHT als Bildschirmausdruck darstellen lassen weil Hintergrundfarben nicht gedruckt werden, kann mir dieses AUSPROBIEREN KLARHEIT bringen!





```

1 REM HIRES+LOW COL:IWT/BIRKENBIHL,1984
2 :
3 HIRES15,14:REC0,0,319,199,1 ← REC*
4 TEXT62,25,"ACHTUNG:",1,2,19 ← TEXT*
5 LOW COL 0,14,0
6 TEXT62,45,"DIES IST",1,2,15
7 LOW COL15,14,0 ← CTRL+B
8 TEXT62,70,"HIRES+LOW COL",1,2,15
9 LOW COL 6,14,0
10 TEXT62,185,"TIEFE=1,ABSTAND=28",1,1,12:REM ÄNDERN SIE DEN AB.
11 LINE 0,105,319,105,1:LINE 0,170,319,170,1
12 FORX=1TO11 STEP .3 ← SCHLEIFENBEGINN
13 F=F+1:LOW COL F,0,0:IF F=5 THEN F=0
14 TEXT 10,120,"FARBWECHSEL",1,1,28: REM ÄNDERN SIE DEN AB.
15 NEXTX:WAIT 653,2

```

ABSTAND

AB.

AB.

CBM - BEFEHL: WARTEN AUF DIE C<-Taste

Experimentieren Sie auch mit TIEFE

ÄNDERN!

\*Achtung: Zur Illustration des LOW COL greifen wir etwas vor: Sie finden zwei Befehle, die im nächsten Kapitel erläutert werden: REC (Rechteck, es zieht den Rahmen) und TEXT (es erlaubt Text im HIRES oder HIRES/MULTI-Modus).



FALLS SIE ABER NUR DIE ZEICHEN-FARBE VERÄNDERN WOLLEN - DANN (siehe Zeilen 5, 7 und 9):

... GEBEN SIE EINFACH BEIM ZWEITEN FARB-PARAMETER D I E S E L B E HINTERGRUNDFARBE EIN, WIE SIE DIESE BEREITS UNTER HIRES FESTGELEGT HATTEN!

Im Gegensatz zu Zeile 13: Hier ist der zweite Farb-Parameter von LOW COL anders! (Das ist ja der Effekt!)

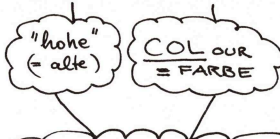
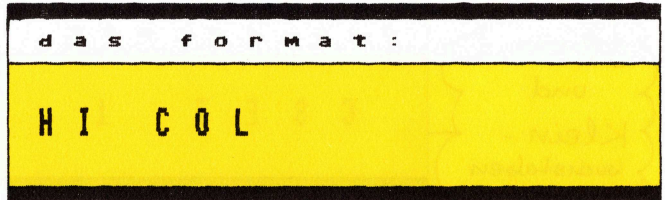
So, das war diese eigentümliche Kombination von HIRES + LOW COL.

Was LOW COL bei MULTI angeht, so ist das problemlos: Genau wie der MULTI-Befehl selbst:

Die 3 Farbparameter setzen 3 neue Zeichenfarben, basta!

Jetzt fehlt noch HI COL. Sie erinnern sich: Die "alten" Farben sind ja schon "oben", also High (= hoch)!

Das Format ist extrem einfach:



Erinnern Sie sich an unsere Eselsbrücke mit den neuen Farben "unten" (im Keller)?

Wir brauchen keine weiteren Parameter, denn HI COL = Zurückholen der »alten« Farben; und die sind ja bereits weiter oben festgelegt worden!

MIT 3 BEFEHLEN KANN MAN EINZELNE  
MODI "EINSCHALTEN" bzw. "UMSCHALTEN":

GROSS-  
buchstaben  
+ GRAFIK-  
zeichen  
GG

```
das format :  
C S E T 0
```

GROSS-  
und  
Klein-  
buchstaben  
GK

```
das format :  
C S E T 1
```

```
das format :
```

```
C S E T 2
```

ACHTUNG: CSET 2 holt die letzte  
Grafik "zurück", wenn man in den  
Text-Modus umgeschaltet hatte!

HIRES EIN  
ohne zu löschen  
im Gegensatz  
zu HIRES, wel-  
ches den Bild-  
schirm auto-  
matisch löscht

---

Mein Tip: Wenn Sie die Möglichkeit die F-Tasten zu programmieren nutzen, dann genügt sogar nur noch ein einziger Tastendruck, z. B.:

f1 = cset 0

f3 = cset 1

f3 = cset 2:pause50

Durch die angehängte PAUSE von ca. 50 Sekunden, welche Sie jederzeit durch RETURN unterbrechen können, können Sie die »zurückgeholte« letzte Grafik noch einmal in Ruhe betrachten.

Oder: Sie schreiben ein Textprogramm in LO RES, das Sie illustrieren wollen. Wenn Sie die Grafik-Befehle ins Programm »schmuggeln« ohne auf Grafik umzuschalten, dann werden Sie bereits auf dem Grafikbildschirm präpariert, ohne daß der Anwender Ihres Programmes dies sieht. Wann immer Sie wollen, lassen Sie diesen Anwender durch CSET 2 die bis jetzt entstandene Zeichnung sehen. Danach wird mit CSET 0 in den Großbuchstaben oder mit CSET 1 in den Groß- und Kleinbuchstaben-Modus zurückgeschaltet, und Ihr Textprogramm läuft weiter. Später fügen Sie neue Details hinzu und lassen wieder einmal die Grafik erscheinen usw.





---

# Kapitelchen

## 3

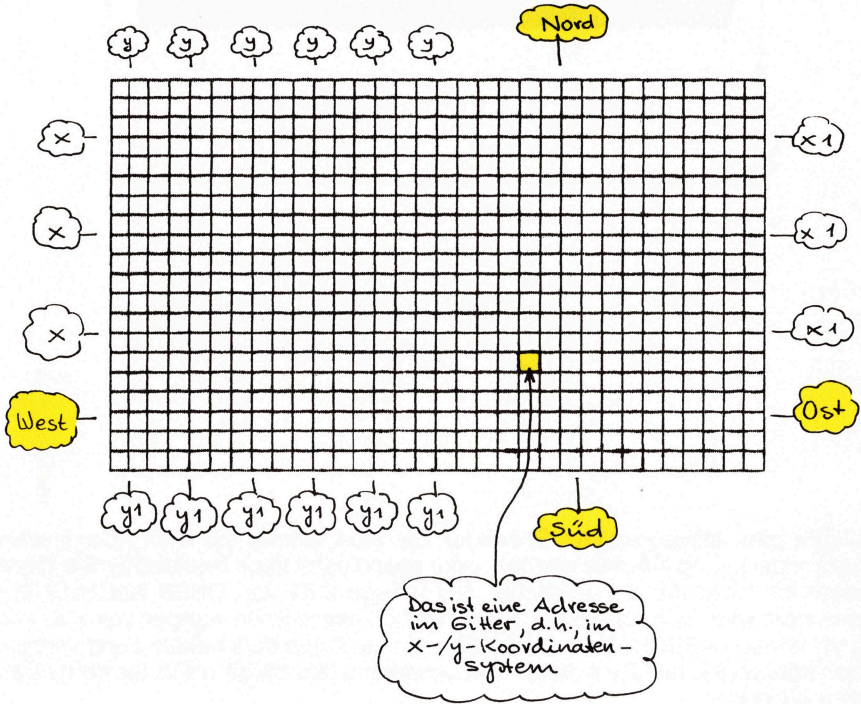
### GRAFIK - Befehle



Es gibt zwei Möglichkeiten: Entweder Sie sind bereits mit dem Koordinaten-System der x- und y-Achse vertraut, oder aber dies ist noch Neuland für Sie. Damit meine ich nicht nur die Tatsache, daß y beim C 64 von OBEN NACH UNTEN berechnet wird (siehe Seite 19), sondern noch etwas: Entfernungen von x zu x1 (y zu y1) können ABSOLUT oder RELATIV sein. Ist Ihnen dies bekannt und vertraut, dann können Sie den Rest dieser Vorbemerkung überblättern. Für Sie geht es auf Seite 43 weiter.

Wenn Sie aber damit noch nicht vertraut sind, dann möchte ich Sie auf etwas aufmerksam machen:

Wenn wir auf einem Koordinaten-System planen, dann ist das ähnlich wie beim Kartenlesen im Straßenverkehr. Die waagerechte x-Achse entspricht der West-Ost-Richtung, die senkrechte y-Achse entspricht (beim C 64) der NORD-SÜD-Richtung. Auf einer Karte könnten wir zum Beispiel feststellen, daß wir 2 Straßen nach Westen und 5 Straßen nach Süden fahren wollen. So ist es auch auf dem x- und y-Gitter:

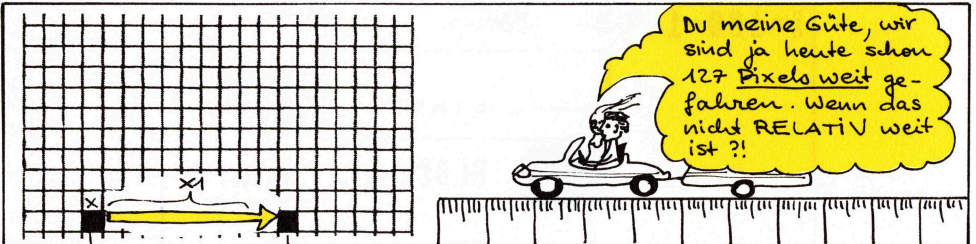
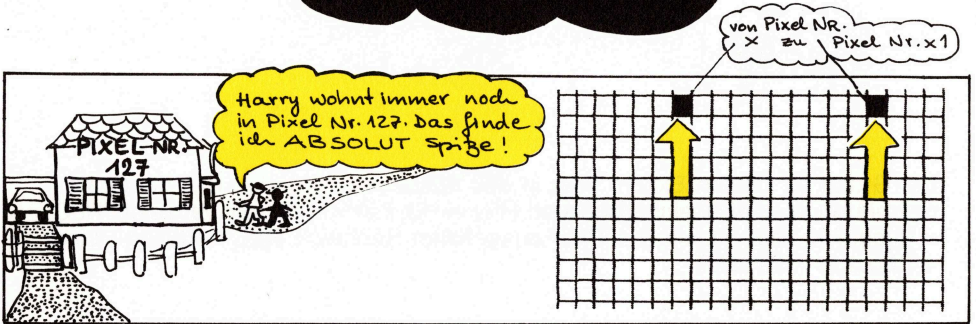


Aber bei *sibas* ist nichts so einfach, wie es sein könnte: Manche Adressen sind relativ, andere absolut. Was heißt das?

Eine ABSOLUTE Adresse ist – auf dem Gitter – ein FESTER Platz, wie die Adresse eines Freundes, den Sie besuchen wollen. Gottseidank müssen Sie nicht immer befürchten, daß sein Haus bis zu Ihrem nächsten Besuch einige Straßen weitergewandert sein könnte...

Anders ist eine RELATIVE »Adresse«: Sie ist RELATIV zum Ausgangspunkt (bzw. zum letzten Punkt von dem aus wir kalkulieren). So sagen wir z.B., wir fahren 300 km bis zum ersten Stop, dann 200 weiter bis zum nächsten Rastplatz usw. Die folgende Abbildung verdeutlicht dies.

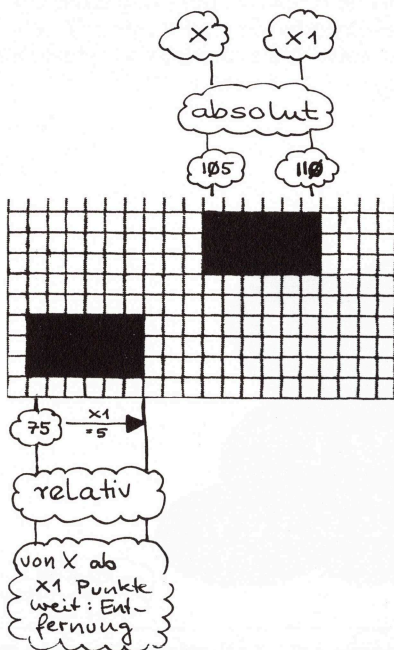
# ABSOLUTE oder RELATIVE Adressen



Von Pixel Nr. x rechnen wir x1 Pixel WEIT (Entfernung).

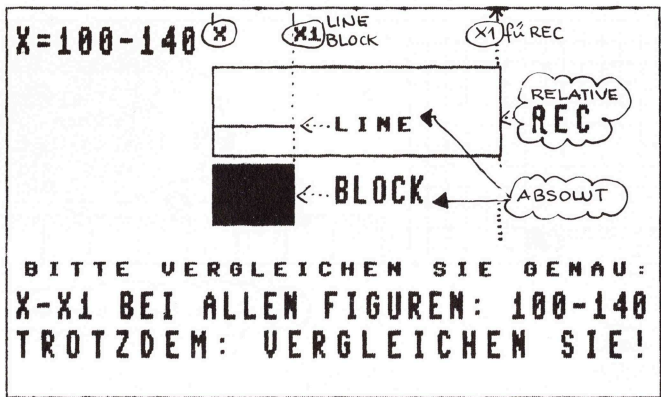
Pixel als Mass-Einheit





Wenn wir X (Y) und X1 (Y1) für ein Rechteck festlegen wollen, haben wir zwei Möglichkeiten: ABSOLUT oder RELATIV

Der Grund für diesen Exkurs liegt in der Tatsache begründet, daß die Grafik-Befehle bei *sibas* nicht einheitlich sind. Hier ein kleiner Vorgriff zum Vergleich: REC = Rechteck, LINE = Linie, BLOCK = ausgefülltes Rechteck: REC = RELATIV, Line + BLOCK = ABSOLUT:



Wie Sie wissen, heißt das ERSTE beim C 64 immer 0:

Zeile 1 heißt Zeile 0

Spalte 1 heißt Spalte 0

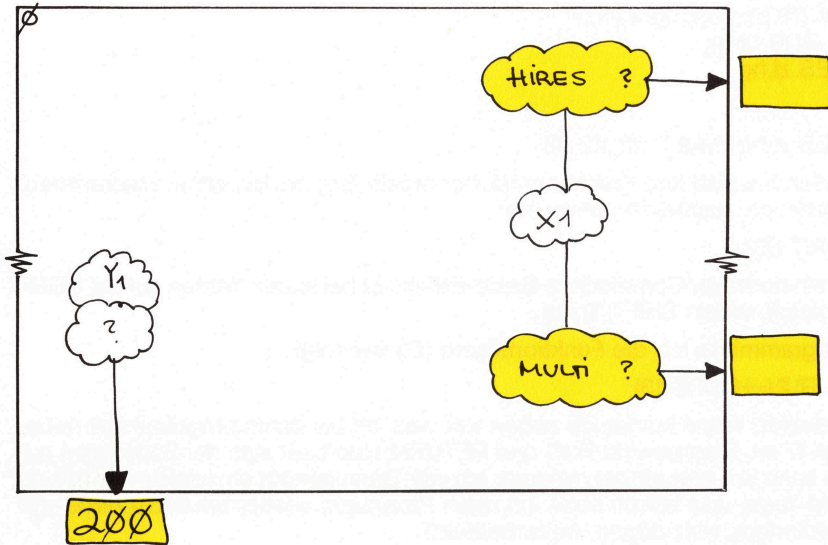
Pixel 1 heißt Pixel 0

usw.

Quizfrage: Wie heißt der LETZTE PUNKT auf der x-Achse –

a) bei HIRES (320\*200)

b) bei HIRES/MULTI (160\*200)?



Wenn Sie die Antwort (319 bei HIRES und 159 bei MULTI) selbst eintragen, dann werden Sie sie nicht mehr so leicht vergessen.

So, und jetzt zu den Befehlen im einzelnen:

---

## DIE BEFEHLE IM EINZELNEN:

Nehmen wir an, Sie wollen etwas entwerfen. Das BASIS-Programm benötigt folgende Informationen:

```
10 REM (TITEL,copyright,etc.)
20 COLOUR ra,hg
30 HIRES zf,hg
```

oder:

```
30 HIRES zf,hg: MULTI zf1,zf2,zf3
```

Nun wollen Sie sich Ihre Kreationen sicher regelmäßig am Bildschirm betrachten. Dazu gebe ich zusätzlich immer ein:

```
1000 WAIT 653,2
```

Das ist ein normaler Commodore-Basic-Befehl. Er bedeutet: Warten auf die CBM-Taste (gleich neben SHIFT, links).

Nun programmiere ich die Funktionstaste (f7) wie folgt:

```
KEY 7, »RUN«+CHR$(13)
```

Das bedeutet: Wann immer ich sehen will, was ich bis dahin programmiert habe, tippe ich f7 an. Dies bewirkt RUN und RETURN. Nun baut sich der Bildschirm auf und ich kann ihn betrachten, solange ich will. Dann genügt ein leichtes Antippen der CBM-Taste und schon sehe ich mein Programm wieder am Bildschirm und kann verändern, hinzufügen, weiterbasteln!

Wollen wir nun ein BASIS-PROGRAMM schreiben, welches uns folgendes bringt:

1. Hintergrundfarbe hellblau (14)
2. Zeichenfarbe schwarz (0)
3. HIRES-Modus plus
4. die WART-Funktion.

Wenn Sie die f7 als RUN+RETURN nutzen wollen, so programmieren Sie dies jetzt, wie oben beschrieben im Direkt-Modus, dann folgt:

```
10 REM BASISPROGRAMM FÜR HIRES
20 COLOUR 14,14: HIRES 0,14
30 :
1000 WAIT 653,2
```

Als nächstes folgt bei mir immer ein Rahmen, da dieser bei Ausdruck auf Papier immer besser wirkt:

```
40 REC 0,0,319,199,1
```

---

Womit wir gleich zum RECHTECK-Befehl kommen werden. Dieses Basis-Programm können Sie abspeichern, damit Sie es nicht jedesmal wieder neu schreiben müssen. Es kann später jeweils der Programm-Anfang sein, könnte aber auch als Unterprogramm konzipiert werden. Dies wäre dann sinnvoll, wenn Sie mehrere *sibas*-Bildschirme aneinanderreihen wollen. Durch RENUMBER ist es kein Problem:

```
RENUMBER 5000
```

Wenn Sie dies probieren, ist Ihr Unterprogramm fertig. Richtig? Falsch! Denn: vor dem ersten Unterprogramm muß ein END stehen, außerdem sollte ein REM erklären, was dieses Unterprogramm soll. Gute Programmier-Praxis wäre daher noch:

```
4997 END
```

```
4998 REM UNTERPROGRAMM HIRES HELLBLAU/SCHWARZ
```

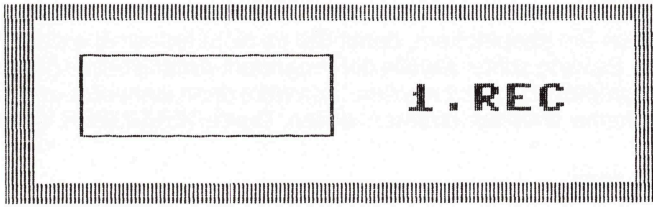
Ich persönlich grenze bestimmte Programmabschnitte oft noch durch Leerzeilen ab, z. B.:

```
4999 :
```

```
5001 :
```

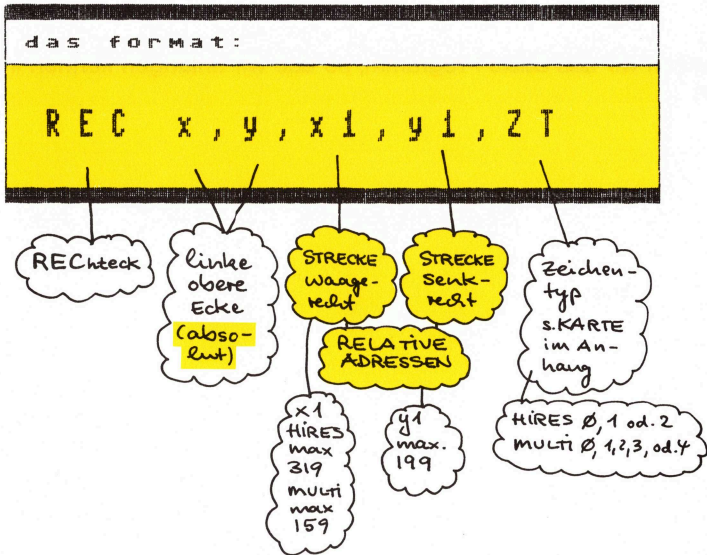
Nun haben wir das Basis-Programm, so daß wir anfangen können ins Detail zu gehen!





Dieser Befehl ist sehr einfach:

$x, y$  bestimmt die linke, obere Ecke des Rechtecks, ist also eine ABSOLUTE Adresse.  $x1, y1$  sind STRECKEN, nämlich die Länge der Seiten.  $x1$  ist die waagerechte Strecke (obere und untere Linie des Rechtecks),  $y1$  ist die senkrechte Strecke (also die beiden senkrechten Seitenlängen):



---

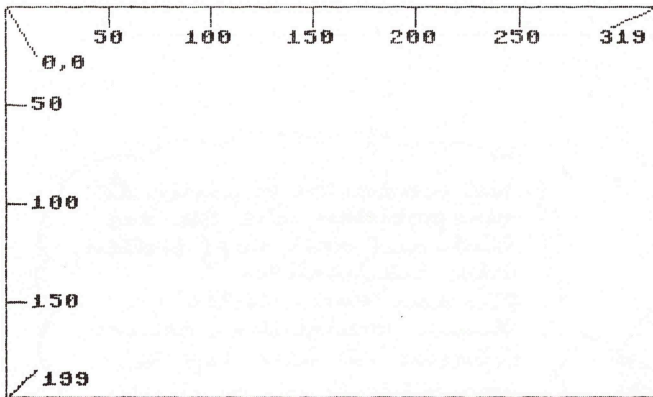
Wir müssen uns nur daran erinnern, daß die Adressen  $x1$  und  $y1$  RELATIV sind.  
Eselsbrücke:

**RE**chteck = **RE**lativ

Also sind  $x1$  und  $y1$  Streckenabschnitte oder Seitenlängen des Rechtecks. Beim Rahmen ist dies noch egal, aber bei Rechtecken innerhalb des Bildschirms ist dies sehr wichtig.

Bitte überlegen Sie mit: Sie wollen drei Rechtecke plazieren, und zwar in etwa da, wo sie in der Abbildung auf der nächsten Seite zu sehen sind.

Probieren Sie ruhig ein wenig herum. Hier die wichtigsten Koordinatenpunkte für  $x$  und  $y$  (welche die linke, obere Ecke bezeichnen):



Achtung: Diese Zahlen beziehen sich nur auf  $x$  und  $y$ , denn  $x1$  und  $y1$  sind ja RELATIVE, Seitenlängen, und müssen dementsprechend gesetzt werden!



# REC - QUIZ

NA, WAS GLAUBEN SIE:

WO LIEGEN DIESE RECS  
AUF  
DER  
X-Y-  
ACHSE  
???

VIEL  
SPASS  
BEIM  
SU-  
CHEN



70 :  
74 REC 50,50,50,50,1  
78 :  
82 REC 150,100,50,50,1  
86 :  
90 REC 25,165,180,20,1  
94 :

---

Sie haben es sicher herausgefunden, zumindestens annähernd! Außerdem haben Sie gesehen, wie leicht ein Quadrat entsteht:

HIRES Quadrat:  $x1$  und  $y1$  identisch.

Da jedoch bei MULTI alle Punkte je 2 Pixel breit sind, müssen Sie beachten:

MULTI Quadrat:  $x1 = 1/2 y1$

z. B.  $x1 = 30, y1 = 60$

$x1 = 15, y1 = 30$

$x1 = 50, y1 = 100$

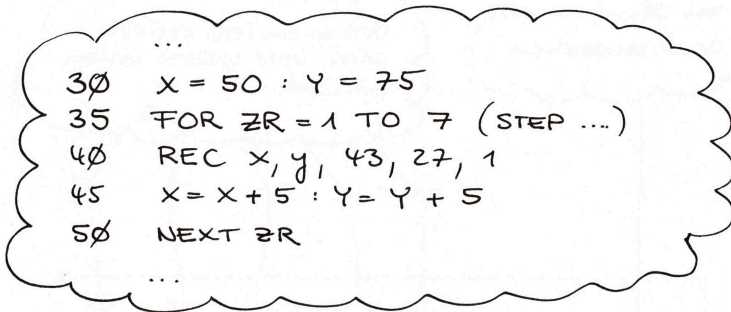
usw.

Wenn Sie im *sibas*-Handbuch geblättert haben, dann ist Ihnen sicher aufgefallen, daß dort viele Grafik-Befehle nur in Beispielen demonstriert werden, welche

1. FOR...NEXT-Schleifen, sowie
2. komplizierte mathematische Formeln

benutzen. Falls Sie sich gewundert haben:

1. Die FOR...NEXT-Schleifen führen zu sehr schönen Effekten, weil eine Form mit leichten Variationen zigmal am Bildschirm erstellt wird, aber –
2. die mathematischen Formeln sind meist nicht notwendig. Es genügt, wenn Sie vorerst einfache Zahlen verwenden, z. B.



...

```
30  X = 50 : Y = 75
35  FOR ZR = 1 TO 7 (STEP ...)
40  REC X, Y, 43, 27, 1
45  X = X + 5 : Y = Y + 5
50  NEXT ZR
```

...



Disk

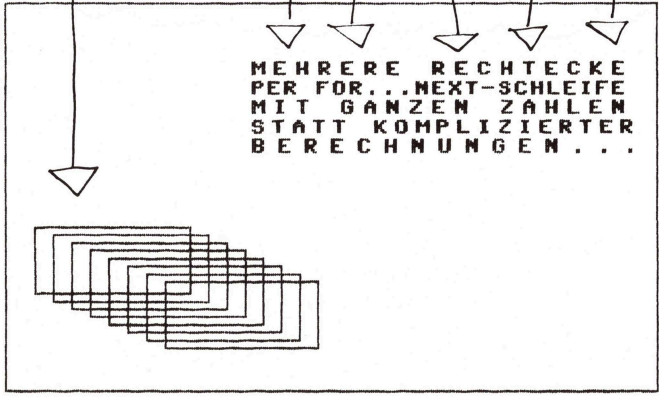
Wollen wir dies testen :

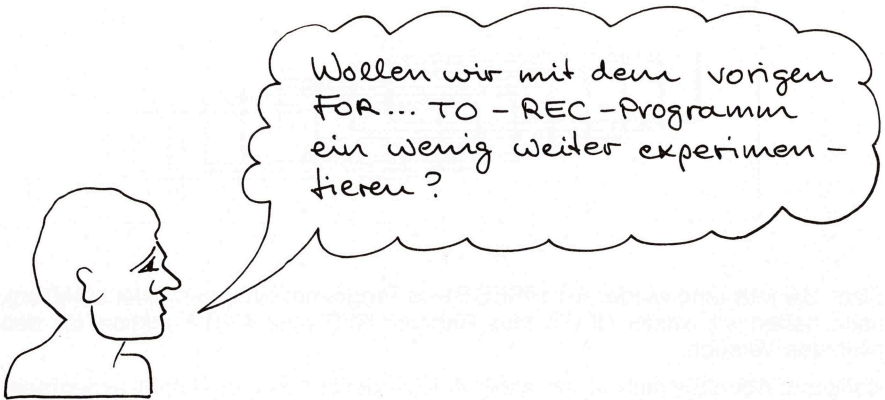
```
2 REM RECHTECKE PER FOR... NEXT-SCHLEIFE MIT GANZEN ZAHLEN ALS PARAMETER
8 REM INT/BIRKENBIHL.1984
14 COLOUR 14,14:HIRES 0,14
20 REC 0,0,319,199,1:REM: RAHMEN-REC
21 :
26 X=15:Y=115
32 :
38 :::FOR ZR=1 TO 8
44 :::REC X,Y,75,34,1
50 :::X=X+9:Y=Y+4
56 :::NEXT ZR
62 :
68 WAIT 653,2:REM COPY
```

BITTE PROBIEREN  
SIE DIESES  
PROGRAMM !

So müssten  
die REC's  
bei Ihnen  
auch aussehen!

Wie so ein Text erstellt  
wird, wird weiter unten  
erklärt.





1. Ändern Sie Programmzeile 14 so, daß auf MULTI umgeschaltet wird:  
14 COLOUR 14,14: HIRES 0,14: **MULTI 6, 15, 0**

Das bedeutet:

Zeichenfarbe 1 = dunkelblau = (6)  
Zeichenfarbe 2 = silbergrau = (15)  
Zeichenfarbe 3 = schwarz = (0)

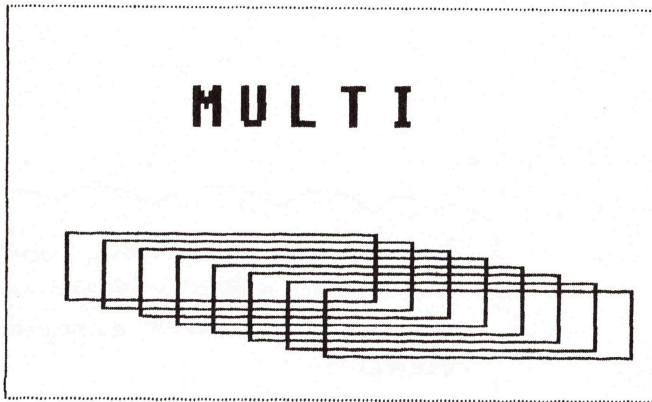
2. Testen Sie das Programm jetzt mit RUN. Was ist passiert mit dem Rahmenrechteck? Warum? Natürlich, bei MULTI ist ja jeder Punkt zwei Pixel breit. Wir haben in der Breite nur noch 160 Punkte. Da aber der erste immer Null heißt, heißt der letzte jetzt 159. Wenn Sie also bitte den Parameter x1 von 319 in 159 ändern, dann haben Sie wieder einen Rahmen! (Zeile 20)

Der Rest des Programms bleibt, aber die Rechtecke sehen anders aus:

```

2 REM RECHTECKE PER FOR... (diesmal MULTI)
8 REM IWT/BIRKENBIHL, 1984
14 COLOUR 14,14:HIRES 0,14:MULTI 6,15,0
20 REC 0,0,159,199,1:REM: RAHMEN-REC
21 :
26 X=15:Y=115
DER REST wie eben : FOR...NEXT Teil

```



Wenn Sie jetzt bitte wieder das HIRES Basis-Programm eintippen (oder LOADen), dann haben wir wieder HIRES plus Rahmen-REC plus WAIT-Funktion, für den nächsten Versuch.

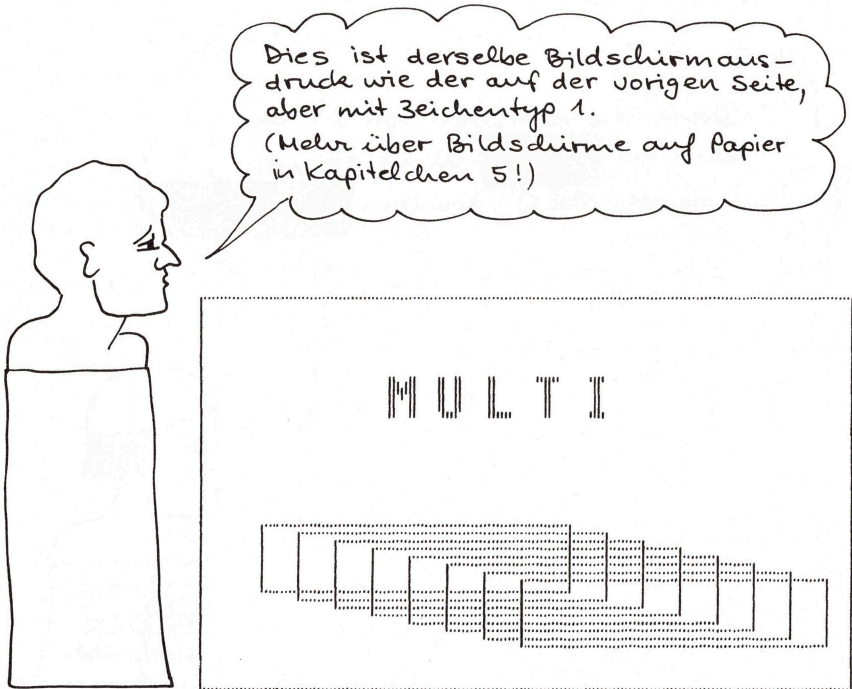
Übrigens: Wenn Sie mal mit, mal ohne MULTI (oder sonst einen Befehl) experimentieren wollen, legen Sie diesen Befehl doch einfach »auf Eis«: Ein :REM\*\*\*\*\*davor schaltet ihn zeitweilig aus, die Sternchen helfen, ihn schnell wieder zu finden. Wenn ich dies mit mehreren Befehlen mache, dann benütze ich statt der Sternchen auch »----« oder »/////« hinter REM.

Auf diese Weise sind sie später schneller wieder zu finden, wenn sie wieder aktiviert werden sollen.

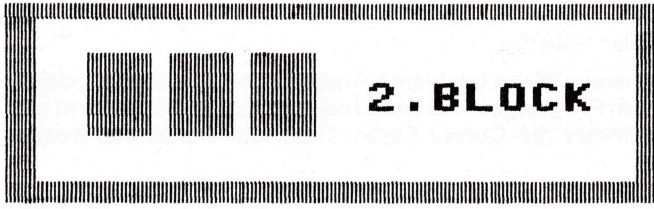
**Noch ein Hinweis: Wenn ich kleine Änderungen vornehme, oder Fehler korrigiere, oder ein Programm vom Directory-Listing am Bildschirm aus lade, dann ändere ich immer die Cursor-Farbe. Dann sieht man viel besser, was man gerade tut!**

Falls Sie dies testen und ein HIRES/MULTI Programm laufen lassen, dann werden Sie allerdings feststellen, daß Ihr Listing eine andere Farbe angenommen hat. Dies hängt damit zusammen, daß die Zeichenfarbe 3 (wie auch die Hintergrundfarbe) vom normalen Farbspeicher aus verwaltet wird (während die Zeichenfarben 1 und 2 aus dem Video-Ram kommen). Dieser Umstand ist dann wichtig, wenn Sie auf Papier drucken wollen, denn:

Zeichenfarben 1 und 2 bei MULTI werden senkrecht schraffiert ausgedruckt, während Zeichenfarbe 3 eine satte, schwarze Farbe ergibt:





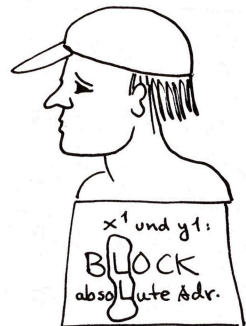


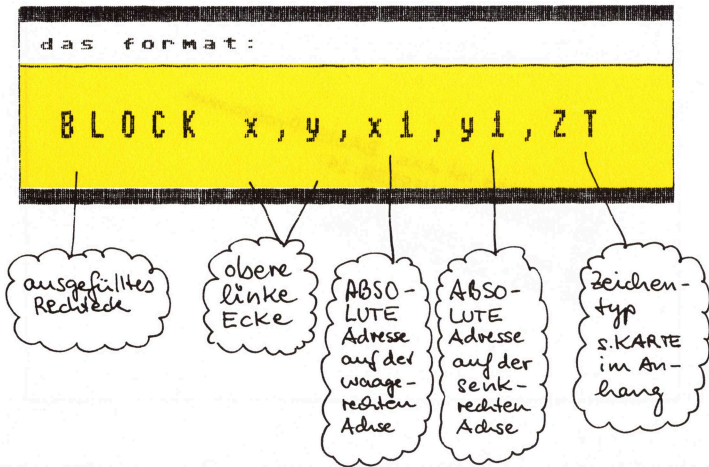
Mit BLOCK erhalten Sie ein ausgefülltes Rechteck. Der Befehl sieht zunächst genau so aus wie REC:

BLOCK x, y, x1, y1, zt.

Aber das stimmt nicht.

Denn die Adressen von  $x1$  und  $y1$  sind bei **BLOCK** nicht RELATIV (wie bei REC), sondern **ABSOLUT!**





Sollten Sie einen ganzen Bildschirm mit einem BLOCK füllen wollen, also bei Pixel Nr. 0 beginnen und bis zum letzten gehen, dann merken Sie noch keinen Unterschied. Probieren Sie einmal, die Zeile NR. 40 aus unserem Basis-Programm zu ändern in:

```
40 BLOCK 0,0,319,199,1
```

Klappt wunderbar. Wenn Sie jetzt einen kleineren Block hineinsetzen, dann verstehen Sie diesen Befehl sehr gut: Da die Adressen absolut sind, ist es leicht, mehrere Blocks miteinander zu verplanen. (Wobei dies auch bei REC sehr angenehm gewesen wäre!): Nehmen wir an, Sie möchten ein Stück aus dem Block heraus schneiden, indem Sie den kleineren Block in Hintergrundfarbe setzen. Nehmen wir desweiteren an, Sie wollen den äußeren Rand, der dann bleibt, 10 Pixel breit haben. Probieren Sie es einmal: Wie muß jetzt der Befehl lauten?

```
45 BLOCK , , , , 0
```

Die Null ist für den Zeichentyp, welcher uns die Hintergrundfarbe beschert.

```
10 REM Dies ist das BASIS-Programm
20 COLOUR 14,0:HIRES0,14:
30 :
40 REC 0,0,319,199,1
90 WAIT 653,2:REM COPY
```

Sicher haben Sie es beim Experimentieren gemerkt: Bei x und y brauchen Sie die 10 Pixels nur zu addieren, bei x1 und y1 jedoch müssen Sie diese abziehen:

Also ist x und y jetzt 10, x1 muß 309 lauten und y1 ist dann 189:

```
10 REM Dies ist der 10-Pixel Rahmen dank
20 COLOUR 14,0:HIRES0,14:
30 :
40 BLOCK 0,0,319,199,1
45 BLOCK 10,10,309,189,0
90 WAIT 653,2:REM COPY
```

*2 blocks !!*

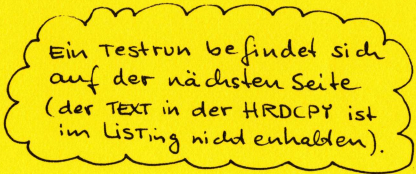
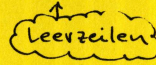
Wenn Sie in der *sibas*-Anleitung beim BLOCK-Befehl nachsehen, dann finden Sie dort die Möglichkeit, eine BLOCK-Grafik zu erstellen. Allerdings ist das Beispiel-Programm typisch: Es arbeitet mit der RND-(Zufalls-)Funktion. Würden Sie nicht lieber eine »echte« BLOCK-Grafik machen?

Die meisten BLOCK-Grafik-Programme in Zeitschriften und Büchern haben einen weiteren Nachteil: Entweder die Blocks hängen von oben nach unten (weil auf y von 0 bis zur gewünschten Zahl der Block entsteht) oder sie laufen von links nach rechts (auf der x-Achse von 0 bis zur gewünschten Zahl). Mich hat dies nicht befriedigt. Ich wollte eine »richtige« Block-Grafik: Eine, bei der die BLOCKS von unten nach oben angeordnet sind. Dazu muß man statt y einfach den Computer den x-Wert errechnen lassen.

Sie können die Werte in DATA-Zeilen packen oder aber per INPUT jedesmal neu eingeben. Zum Experimentieren ist dies die interessantere Version. Hier ist sie:



```
10 REM BLOCK-GRAFIK MIT INPUT
20 REM IWT/BIRKENBIHL, 1984
22 COLOUR 14,14
24 PRINTCHR$(147):REM CLR SCREEN = Bildschirm löschen
30 PRINT"#####":REM "☐"=CTRL+7:DUNKELBLAU;"␣"=CRSR RUNTER:JE 1 LEERZEILE
32 :
34 PRINT"SIE KOENNEN DIE WERTE FUER DIE BLOCK"
36 PRINT"☐GRAFIK SELBST EINGEBEN. WENN SIE"
38 PRINT"☐BITTE BEACHTEN: ZAHLEN UNTER 10 SIND"
40 PRINT"☐ZU KLEIN, MAXIMUM IST 100. ALLE"
42 PRINT"☐WERTE MUESSEN DURCH KOMMATA GETRENNT"
44 PRINT"☐SEIN. BEISPIEL:"
46 PRINT"☐23,45,59,67,80,112,130,145,45,56":PRINT"#####"
48 INPUT A,B,C,D,E,F,G,H,I,J
50 HIRES 0,14
52 REC 0,0,319,199,1
54 :
56 BLOCK 30,(198-A), 40,198,2
57 BLOCK 45,(198-B), 55,198,2
58 BLOCK 60,(198-C), 70,198,2
59 BLOCK 75,(198-D), 85,198,2
60 BLOCK 90,(198-E),100,198,2
61 BLOCK 105,(198-F),115,198,2
62 BLOCK 120,(198-G),130,198,2
63 BLOCK 135,(198-H),145,198,2
64 BLOCK 150,(198-I),160,198,2
65 BLOCK 165,(198-J),175,198,2
90 WAIT 653,2:REM ***COPY
```





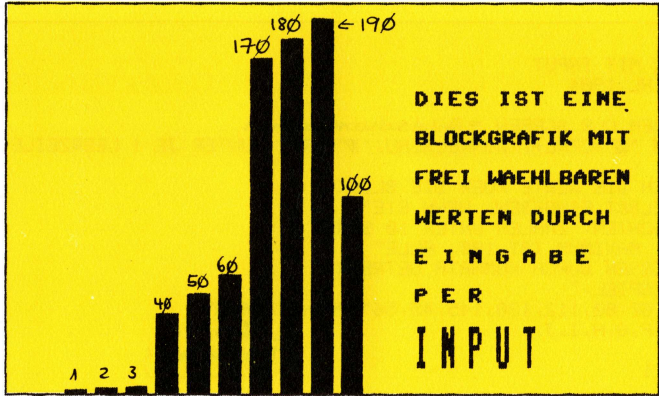
SIE KOENNEN DIE WERTE FUER DIE BLOCK  
GRAFIK SELBST EINGEBEN. WENN SIE  
BITTE BEACHTEN: MINIMUM = 1; UND  
MAX.=197. WIE BEKANNT, MUESSEN  
ALLE WERTE DURCH KOMMATA GETRENNT  
EINGEGEBEN WERDEN. ZUM BEISPIEL:

~~1,2,3,40,50,60,170,180,190,100~~

← REVERSE  
SCHRIFT, im  
HARD COPY immer  
schlecht zu  
lesen

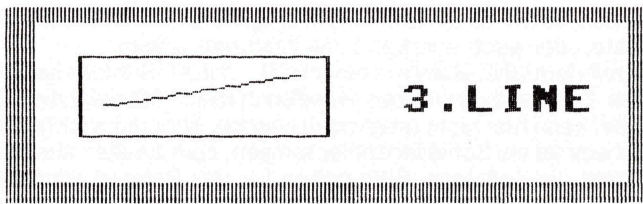
Das war mein INPUT

? 1,2,3,40,50,60,170,180,190,100



← Dieser TEXT  
fehlt im  
Listing

Ehe Sie zum nächsten Befehl übergehen, wollen Sie vielleicht noch ein wenig mit  
MULTI experimentieren – so wie nach dem REC-Befehl vorhin?



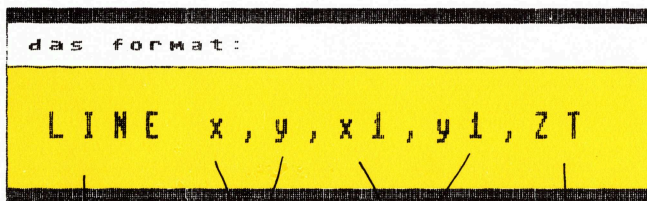
Der nächste Befehl zieht eine Linie:

Wenn Sie den Block-Befehl im Griff haben, ist LINE ein Kinderspiel. Eigentlich können Sie mit BLOCK eine Linie ziehen: Wenn die beiden Parameter x und x1 bzw. y und y1 identisch sind, dann bekommen Sie einen »Block« von 1 Pixel Breite oder Höhe:

BLOCK 25,50,130,50,ztyp (waagrecht)

BLOCK 25,50,25,150,ztyp (senkrecht)

Da bei **BL**ock und **LI**ne die Adressen **absolu**t sind, ist die Parallele klar. Wenn die Linie allerdings schräg verlaufen soll, dann setzen Sie unterschiedliche Parameter bei x und x1 bzw. y und y1 ein. Das Format ist sehr einfach:



Linie  
ziehen

ERSTER  
Endpunkt  
der Linie  
ANFANG

ZWEITER  
Endpunkt  
der Linie  
(ABSOLUT)  
ENDE

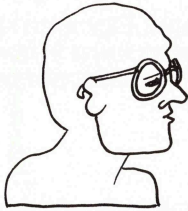
Zeichen-  
typ  
s. KARTE  
im  
Anhang

LINE ist  
wie der  
BLOCK-Befehl:  
ABSOLUTE  
Adressen  
bei x1 und  
y1

---

Sicher können Sie sofort mit diesem Befehl umgehen, wenn Sie *eine* waagerechte oder senkrechte oder auch schräge Linie zeichnen wollen.

Aber: Gerade mit dem LINE-Befehl in einer FOR...NEXT-Schleife lassen sich tolle Effekte erzielen. Das Beispiel im *sibas*-Handbuch (Seite 30) zeigt dies. Die mathematische Formel kann hier nicht umgangen werden, aber der wichtigste Faktor ist m.E. der, daß x einmal als Schleifenzähler fungiert, zum zweiten aber als Variable. Dadurch entsteht die Spitzform. Bitte geben Sie das Beispiel einmal ein, wie es hier folgt. Testen Sie dann noch die beiden folgenden Variationen durch; dabei werden Sie ein gutes Gefühl für diese Art von Befehlsaufbau erhalten. Falls Sie selbst solche Grafiken *entwerfen* wollen, dann denken Sie bitte auch daran, daß der Commodore 64 verschiedene Funktionen (SIN, COS, usw.) beherrscht. Diese können natürlich bei solchen Figuren voll eingesetzt werden (im Data Becker-Buch finden Sie 3 Beispiele mit SIN).



TOLLE GRAFIKMÖGLICH-  
KEITEN MIT LINE!

Achtung: Auch wenn Sie nur 1 Zeichen-  
farbe (bei schwarzem HG) benutzen:  
Ihre Grafik wird ganz phantas-  
tisch in allen Regenbogenfarben  
schillern. Testen Sie dies, indem  
Sie beim folgenden Listing auch  
mal HIRES 10,0 oder 1,0  
eingeben...

Disk

Das sieht ja  
so richtig  
professionell  
aus !!



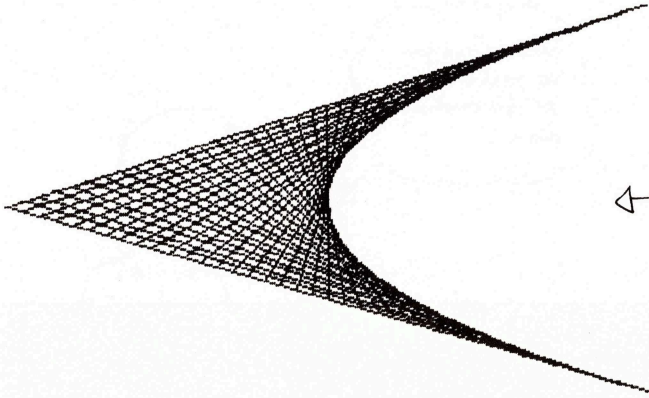
```
2 REM SIBAS ANLEITUNG, SEITE 30
4 REM BEISPIEL PROGRAMM ZU LINE
6 :
8 :
10 HIRES 1,6
12 ::: FOR X=0 TO 320 STEP 8
14 ::: M=100/320*X:REM ACHTUNG: X = GLEICHZEITIG SCHLEIFENZAehler!
16 ::: LINE 320-X,M,X,100+M,1
18 ::: NEXT X
20 REM STATT ZEILE 200 GOTO 200 FOLGT UNSER BEWAHRTES WAIT 653,2
22 WAIT 653,2:REM COPY
24 :
26 REM ES FOLGT DIE ERSTE VARIATION
28 :
30 HIRES 1,6
32 ::: FOR X=0 TO 320 STEP 16
34 ::: M=100/320*X:REM ACHTUNG: X = GLEICHZEITIG SCHLEIFENZAehler!
36 ::: LINE 320-X,M,X,100+M,1
38 ::: NEXT X
40 WAIT 653,2:REM COPY
42 :
44 REM ES FOLGT DIE ZWEITE VARIATION
46 :
48 HIRES 1,6
50 ::: FOR X=0 TO 320 STEP 32
52 ::: M=100/320*X:REM ACHTUNG: X = GLEICHZEITIG SCHLEIFENZAehler!
54 ::: LINE 320-X,M,X,100+M,1
56 ::: NEXT X
58 WAIT 653,2:REM COPY
```



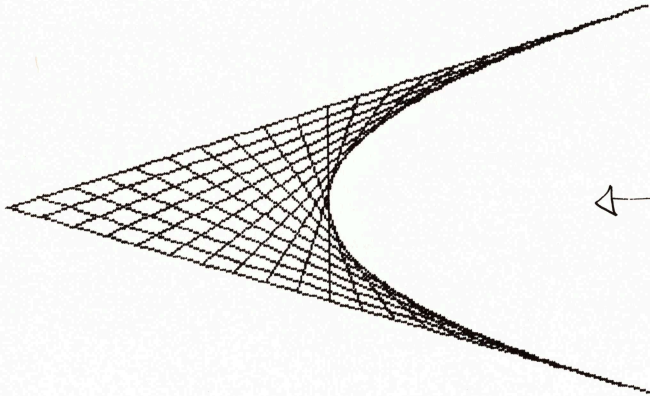
Natürlich  
benutzen  
Sie ab  
Zeile 30  
den Editier-  
Modus,  
oder ?!

Auf der nächsten  
Seite sehen Sie  
alle 3 Möglichkeiten  
als HRDPY aus-  
gedruckt.

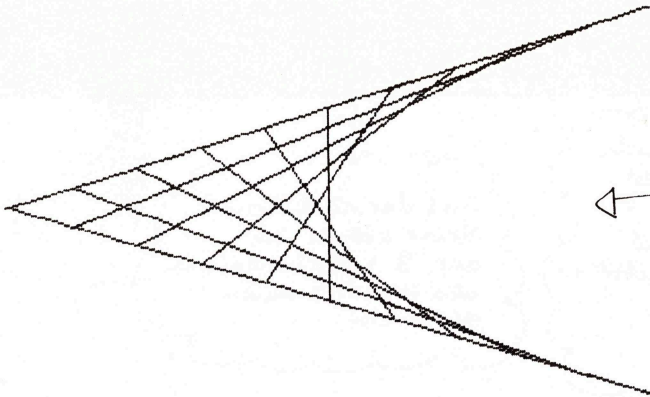




Zeilen  
10 - 22



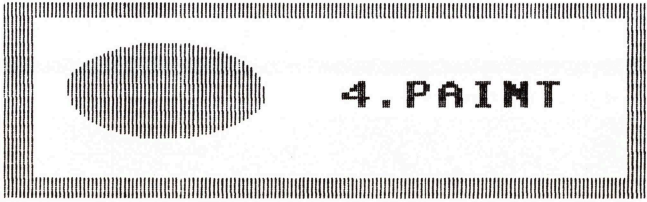
Zeilen  
30 - 40



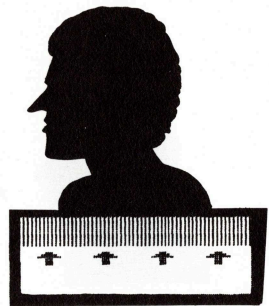
Zeilen  
48 - 58

---

Ein kinderleichter Befehl ist der nächste:



Nur BLOCK erlaubt das sofortige Erstellen einer gefüllten Figur (Rectes). Wenn wir aber eine andere Fläche, egal welcher Form, bemalen wollen, dann brauchen wir den PAINT - Befehl. Dieser ist sehr angenehm: Alle Rahmen bei den Titeln (wie oben) sind mit PAINT entstanden...



PAINT braucht nur drei Parameter: x und y müssen irgendwo innerhalb der Fläche sein, die Sie PAINTen (ausmalen) wollen, dann folgt der Zeichentyp:

```
das format:  
  
P A I N T   x , y , Z T
```

beMALEN  
= mit einer  
Farbe  
ausfüllen

irgendein  
Punkt  
innerhalb  
der zu  
füllenden  
Fläche

Zeichentyp,  
s. KARTE  
im Anhang

ACHTUNG: Sollte  
die Linie, welche  
die zu PAINTende  
Fläche umschließt,  
an irgendeiner  
Stelle "undicht" sein  
(ein Pixel genügt),  
dann läuft die  
füllende Farbe  
"hinaus" aus dem  
Feld !!

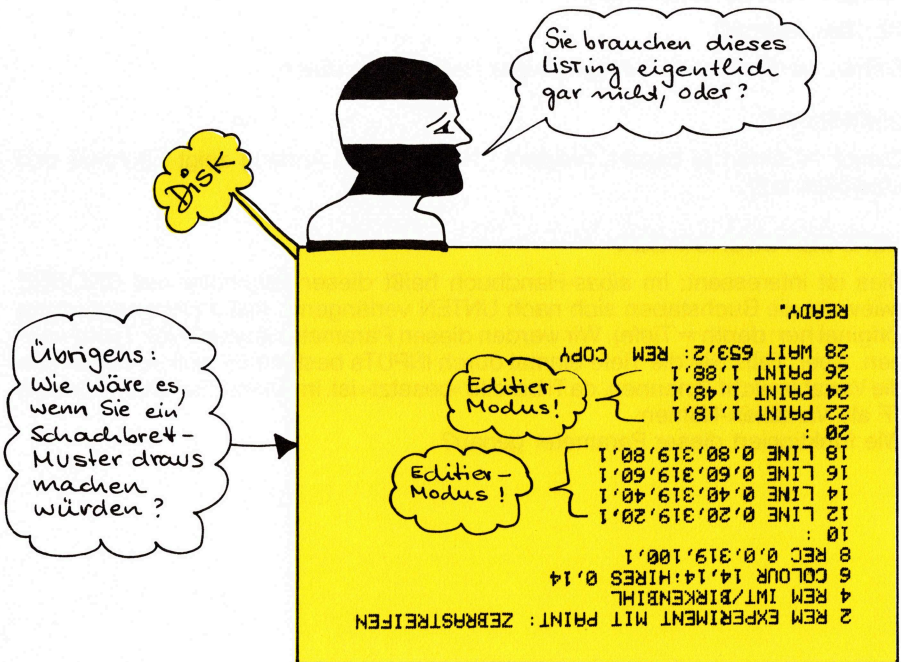
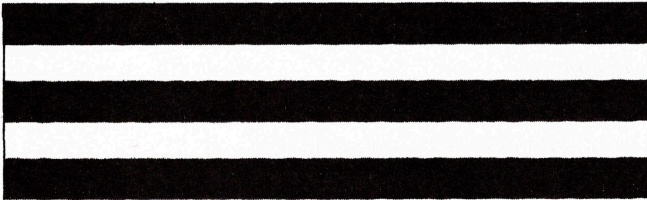


Wollen Sie wieder experimentieren?

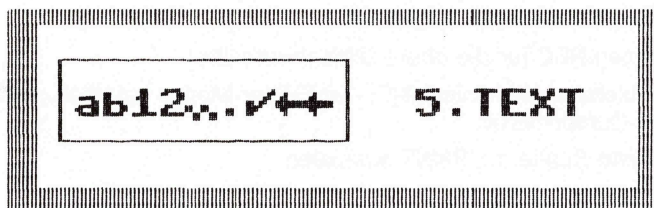
Dann probieren Sie einmal folgendes:

1. Ein Rahmen-REC für die obere Bildschirmhälfte!
2. Alle 20 Pixel eine Querlinie (LINE) – im Editier-Modus natürlich, optimal mit veränderter Cursor-Farbe.
3. Jede zweite Spalte mit PAINT ausmalen.

Das sieht dann so aus:







Wenn Sie bis hierher immer mit-probiert haben (damit Ihre BEIDEN Hirnhälften informiert wurden), dann sind Sie jetzt gut vorbereitet für einen Befehl, der zunächst relativ problematisch wirkt, weil er viele Parameter hat. Aber, dieser Befehl ist ungeheuer wichtig, wenn Sie TEXT im HIRES-Modus setzen wollen.

Hierzu müssen wir festlegen:

- 1: Die linke, obere Ecke des \$string (x, y)
- 2a: Innerhalb von Hochkommas den \$string selbst («...»)
- 3b: Innerhalb von den Hochkommas: Ob Groß- oder Kleinbuchstaben; falls Sie NICHTS festlegen, werden die Buchstaben GROSS
- 4a: Der Zeichentyp
- 4b: Die Tiefe der Buchstaben
- 4c: Der Abstand

Gehen wir die letzten 3 Angaben der Reihe nach durch:

#### ZEICHENTYP

Dieser ist Ihnen ja bereits bekannt. (KARTE 4 im Anhang zeigt nochmal den Überblick auf)

#### TIEFE DER BUCHSTABEN

Dies ist interessant: Im *sibas*-Handbuch heißt dieser Parameter mal GRÖSSE (wiewohl die Buchstaben sich nach UNTEN verlängern), mal d (vom englischen Original her: depth = Tiefe). Wir werden diesen Parameter immer TI (für Tiefe) nennen. Aber: Falls Sie die Tiefe einmal durch INPUTs bestimmen wollen, dürfen Sie die Variable nicht TI nennen, da TI bereits »besetzt« ist. In diesem Fall arbeite ich mit TF als Variablen-Namen.

Wie funktioniert dieser Parameter genau?

---

Tiefe bestimmt, wie oft die Länge des Buchstabens multipliziert und nach UNTEN gezogen werden soll, also:

- TI = 1: Normale Buchstabengröße
- TI = 2: Doppelte Länge nach UNTEN
- TI = 3: Dreifache Länge nach UNTEN

## ABSTAND

Der nächste Parameter hat mit dem Abstand in der Breite zu tun. Er ist aber nicht analog zu TI, so daß ABSTAND 1 nicht Buchstabenbreite, sondern Pixel- (bei MULTIPUNKT) ABSTAND bedeutet. Dies ermöglicht sehr eigenartige Effekte:

1. Minimum-ABSTAND von 7 ist notwendig, damit einzelne Zeichen sich nicht überschneiden, wie bei Zahlen oder Buchstaben.
2. ABSTAND hilft uns, Zeichenketten von unterschiedlicher Länge so raffiniert am Bildschirm zu plazieren, daß quasi ein Randausgleich entsteht. Für diese Funktion ist es wunderbar, daß ABSTAND Pixel (Punkte) und nicht Zeichenbreiten bestimmt.
3. ABSTAND von weniger als 7 ist immer dann sinnvoll, wenn Sie Grafik-Zeichen durchgehend (also ohne ABSTAND) plazieren wollen. Somit können Sie z. B. das Zeichen auf der T-Taste (mit cmb) zum Unterstreichen benützen. Denn, ABSTAND von <7 bewirkt eine durchgehende Linie, während ein ABSTAND von >8 eine gestrichelte Linie ergibt!

Der folgende Bildschirmausdruck verdeutlicht dies:



*auch andere Grafik  
Symbole möglich:  
Interessante Effekte  
machbar! Bitte  
selber testen, später!*

So, und jetzt das FORMAT des TEXT-Befehls! Beachten Sie jedes KOMMA genau, vor allem vor und hinter den Hochkommata!

Das soll Text illustrieren... denn dieses ist sehr nützlich, wenn man Grafiken mit Text versehen will. Deshalb muss man diesen Befehl gut kennen!!! Und überhaupt: So viele Illustrationen zu diesem Buch wären ohne Text unmöglich gewesen. Also frische ans Werk!

```
das format:
TEXT x,y,"*...",ZT,TI,AB
```

```
TEXT:TEXT
```

Zeichen von der Tastatur: Buchstaben, Ziffern, Grafikzeichen...

linke obere Ecke des \$

Zeichentyp. s. KARTE im Anhang

Abstand: Wird in PIXEL oder PUNKTEN gezählt:  
 < 7 führt zu Überschneidungen der einzelnen Zeichen.  
 > 7 verhindert diese Überlappungen, die jedoch zu interessanteren Effekten führen

Siehe nachfolgenden Text bezüglich: CTRL+A oder CTRL+B

TIEFE  
 1=normal  
 2=doppelt  
 3=dreifach  
 u.s.w.

oder:

```
das format:
TEXT x,y,text$,ZT,TI,AB
```

Das »\*« im oberen Format bezieht sich auf die Möglichkeit, zwischen Groß- und Kleinbuchstaben auszuwählen. Es kann an jeder Stelle innerhalb der Hochkom-

mas auftreten, so daß ein Abwechseln von Groß- und Kleinbuchstaben mehrere solcher Kontrollzeichen im LISTING ergeben wird! Diese Zeichen sind genau wie Farbwahlzeichen im Print-Befehl zu verwenden.



```

2 REM TEXT: ERSTES BEISPIEL
4 REM IWT/BIRKENBIHL
6 :
8 ZT=1:REM ZEICHENTYP
10 :
12 REM TF (TIEFE) UND AB (ABSTAND) WERDEN PER INPUT GESTGELEGT
13 COLOUR 14,14:PRINTCHR$(147)"■":REM "■"=CTRL+I:SCHWARZ
14 PRINT"■■■■■■■■":REM 7 LEERZEILEN
15 PRINT"BITTE GEBEN SIE TF (TIEFE) UND "
16 PRINT"■■AB (ABSTAND) FUER DEN TEXT-BEFEHL"
17 PRINT"■■SELBER EIN!■■■■":INPUT TF,AB
18 :
36 COLOUR 14,14:HIRES 0,14
38 REC 0,0,319,100,1
40 TEXT 30,30,"ABCDEFGHI123",ZT,TF,AB
90 WAIT 653,2:REM COPY
  
```

TESTRUN:

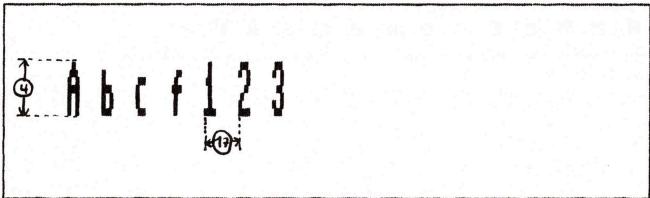
CTRL + B

CTRL + A

Achtung: TEXT und INPUT von TF und AB vertragen sich nur bedingt: Der \$string muss kürzer als 8 Zeichen sein, sonst geht es gar nicht !!

BITTE GEBEN SIE TF (TIEFE) UND  
 AB (ABSTAND) FUER DEN TEXT-BEFEHL  
 SELBER EIN!

? 4,17





---

DIES IST EINE TEXTZEILE

Nun wird es spannend: Mit den besprochenen 5 Befehlen, genau mit überwiegend 4 davon, können wir jetzt einen Arbeitsplatz simulieren: Nämlich den des Foto-Setzers! Im Nu entstehen:

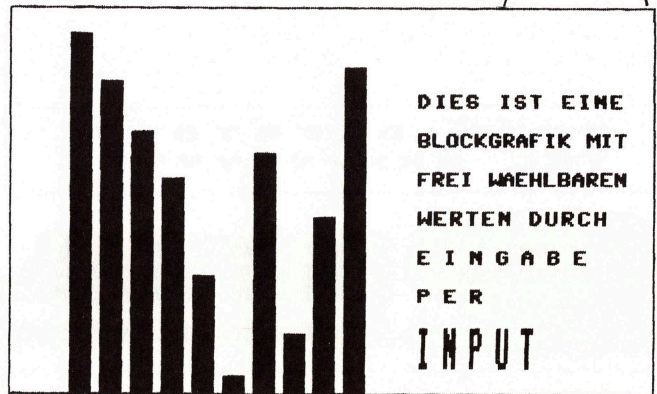
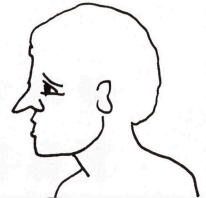
- Visitenkarten
  - Kleinanzeigen
  - Einladungskarten
  - Überschriften
- usw.

Bitte beachten Sie: so sind alle Kapitel-Überschriften, alle Unter-Überschriften, alle Formate dieses Buches entstanden. Außerdem viele Illustrationen, z.B. für ABSOLUT + RELATIV, für den Vergleich zwischen REC + BLOCK, und viele mehr. So ein »Bildschirm« kann, wenn Sie die oben beschriebenen Befehle im Griff haben, innerhalb weniger Minuten fertiggestellt werden. Dazu müssen Sie aber insbesondere den TEXT-Befehl gut beherrschen. Deswegen: Jetzt gleich experimentieren, damit Ihr *ganzes* Gehirn mühelos lernen kann!

Beginnen Sie mit dem BASIS-Programm (HIRES, REC: Rahmen und dem WAIT-Befehl), und basteln Sie jetzt ein »Schriftstück«, ähnlich den folgenden Beispielen. Hier zunächst einige Beispiele zur Einstimmung:

```
GROSSBUCHSTABEN
kleinbuchstaben
Gross / Klein
GRAFIKZEICHEN :
ABSTAND < 7 : ■ ■ ■ ■
ABSTAND > 7 : ██████████
BESONDERS TIEF
NORMALE GROESSE !
```

Erinnern Sie sich noch an unsere BLOCK-Grafik vorhin (bei BLOCK)? Da fehlten die TEXT-Befehle im Listing. Hier ist ein neuer Test-Run mit anderen Werten und, natürlich, das Listing für die TEXT-Strings:



REST DES  
LISTINGS  
DER BLOCK-  
GRAFIK

```
93 TEXT 200,45,"DIES IST EINE",1,1,8
94 TEXT 200,65,"BLOCKGRAFIK MIT",1,1,7
95 TEXT 200,85,"FREI WAEHLBAREN",1,1,7
98 TEXT 200,105,"WERTEN DURCH",1,1,8
99 TEXT 200,125,"EINGABE",1,1,13
100 TEXT200,145,"PER",1,1,13
101 TEXT200,165,"INPUT",1,3,13
```

Die Textzeilen könnten Sie –

- analysieren
- mit denselben Parameter-Werten aber veränderten Strings einmal eingeben ...

EINMAL  
EINGEBEN  
ZIG KARTEN  
SPAETER  
MACHEN !!

DISK

```
2 REM VISITENKARTE BLANKO
4 COLOUR 14,14:HIRES 0,14:MULTI 6,6,6
6 REC 0,0,159,199,1
8 :
10 LINE 0,155,320,155,1
24 TEXT 25,43,"VORNAME",3,2,13
25 TEXT 25,75,"NACHNAME",3,3,12
26 TEXT 12,165,"XXXX DIESTADT",3,1,10
27 :
28 TEXT 12,180,"TEL XXX/YYYYYY",3,1,9
100 WAIT 653,2:REM COPY
```

```
VORNAME
NACHNAME
```

```
XXXX DIESTADT
TEL XXX/YYYYYY
```

Dieses  
Blanko -  
Visiten -  
Karten -  
formular  
wird in Null  
Komma nix  
Konkret :

Achtung: Auch  
ABstände an -  
gleichen!

```
24 TEXT 25,43,"TANJA",3,2,14
25 TEXT 25,75,"MENNEN",3,3,13
26 TEXT 12,165,"0000 MUENCHEN",3,1,10
27 :
28 TEXT 15,180,"TEL 089/2131754",3,1,8
```

```
TANJA
MENNEN
```

```
0000 MUENCHEN
TEL 089/2131754
```

# QUIZ:

Schaffen Sie es,  
diese Visitenkarte  
zu programmieren?  
Dann können Sie auch auf  
"Kundenwünsche" Ihrer  
Freunde eingehen!

WER  
SCHAFFT  
ES,  
OHNE im  
Listing  
(unten)  
nachzusehen?

**JUERGEN POKORNY**

**VIDEO+HIFI+COLOR-TV**

Schloss-Str.2, TEL:08134/6961  
8063 Odelzhausen b.Muenchen

ZUR  
KON-  
TROLLE,  
nach  
Ihrem  
Erfolg...  
oder ?!

10 REM PROGRAMM VISITENKARTE  
12 COLOUR14,14:HIRE90,14  
14 REM POKORNY JUERGEN  
16 :  
18 REC 0,0,310,199,1  
20 :  
22 LINE 0,45,310,45,1  
24 TEXT 49,13,"JUERGEN POKORNY",1,3,15  
25 :  
26 TEXT 35,60,"VIDEO+HIFI+COLOR-TV",1,1,13:REM  
30 TEXT 35,157,"SCHLOSS-STR.2, TEL:08134/6961",1,1,18  
32 TEXT 35,172,"8063 ODELZHAUSEN B.MUNCHEN",1,1,19  
34 :  
35 REC 50,80,213,65,1  
100 MAIL 653,2



So sieht meine eigene Sibas-Karte jetzt aus

## VERA F. BIRKENBIHL

IMPORT & EXPORT VON WISSEN

- SEMINARE
- VORTRÄGE
- VERÖFFENTLICHUNGEN:  
BÜCHER & ARTIKEL

Muehlweg 2, 8063 Odelzshn:08134/478

Und hier ist das Listing dazu. Besonders die drei kleinen Blocks sollten Sie erst einmal zu programmieren versuchen ...?

```

1: 1 REM ENTWURF(MIT SIMON'S BASIC) FUER IWT VON: VERA F.BIRKENBIHL, 1984
2:
3: HIRES 0,14
4:
5: REC 0,0,319,199,1
6:
7: LINE 0,40,319,40,1
8:
9: TEXT 42,15,"#VERA F. BIRKENBIHL",1,3,13
10:
11: TEXT 39,50,"#IMPORT & EXPORT VON WISSEN",1,1,9
12:
13: TEXT 17,178,"#MUEHLWEG 2, 8063 #ODELZSHN:08134/478",1,1,8
14: WAIT 653,2
15: BLOCK 80,80, 94,94,1
16: GOTO 29
17: TEXT 110,80,"#SEMINARE",1,2,16
18:
19: BLOCK 80,110,94,124,1
20:
21: TEXT 113,110,"#VORTRAEGE",1,2,14
22:
23: BLOCK 80,140,94,154,1
24:
25: TEXT 110,138,"#VEROEFFENTLICHUNGEN:",1,1,7
26:
27: TEXT 110,149,"#BUECHER & ARTIKEL",1,1,8
28:
29: WAIT 653,2:REM COPY
30:
READY.
  
```

EDITIER-MODUS :  
 Edit -  
 modus :

IWT

solche Waits  
zwischen  
helfen mir,  
den ersten  
Programmteil  
besonders an-  
zusehen.

solche GOTOs  
während der  
Arbeit helfen  
ebenfalls!

Als ich meinem Vater die ersten *sibas*-Visitenkarten-Entwürfe zeigte, hatte er eine Idee: »Ich bräuchte ein kleines Formular, das ich auf ein DIN A 4-Blatt oben rechts montieren kann, für Notizen bei Telefonaten. Diese Blätter könnten dann im Aktenordner bei der Korrespondenz mit eingelegt werden. Kannst Du mir sowas machen?«

Es dauerte keine 5 Minuten. Wollen Sie mal sehen, wie lange Sie brauchen würden??

Na, wäre so ein Formular nichts für Ihre Telefon-Notizen?

**BESPRECHUNGSNOTIZ :**

.....

**DATUM:** .....

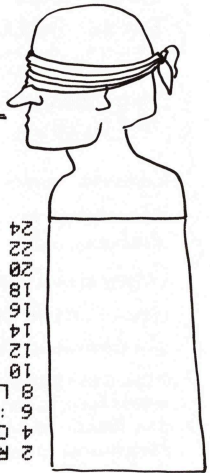
**FIRMA:** .....

**TEL.:** ..... / ..... - .....

**PARTNER:** .....

.....

Sie brauchen das Listing doch nur noch zur Kontrolle, oder, noch besser: überhaupt nicht mehr, oder ?!



```

2 REM BESPRECHUNGSNOTIZ/TELEFON FUER WATI (DEZ.1983)
4 COLOUR 14,14:HIRES 0,14:REC 0,0,319,199,1
6 :
8 LINE 0,40,319,40,1
10 TEXT 30,12,"BESPRECHUNGSNOTIZ:"",1,3,15
12 :
14 TEXT 32, 89,"FIRMA:"",1,1,11
16 TEXT 32,110,"TEL." / ..... - .....",1,1,17
18 TEXT145, 65,"PARTNER:"",1,1,10
20 TEXT 32,141,"PARTNER:"",1,1,11
22 TEXT 30,165,".....",1,1,11
24 WATI 630,2

```

HIER ein Blanko -  
Beispiel für eine  
Tabelle ...

Letztes  
TEXT-  
Beispiel

KATEGORIE EINS		KATEGORIE ZWEI	
EINTRAG	01	EINTRAG	A
EINTRAG	02	EINTRAG	B
EINTRAG	03	EINTRAG	C
EINTRAG	04	EINTRAG	D
EINTRAG	05	EINTRAG	E
EINTRAG	06	EINTRAG	F
EINTRAG	07	EINTRAG	G

Hier eine  
Variante  
davon →

KATEGORIE EINS		KATEGORIE ZWEI	
EINTRAG	1	EINTRAG	A
EINTRAG	2	EINTRAG	B
EINTRAG	3	EINTRAG	C
EINTRAG	4	EINTRAG	D
EINTRAG	5	EINTRAG	E
EINTRAG	6	EINTRAG	F

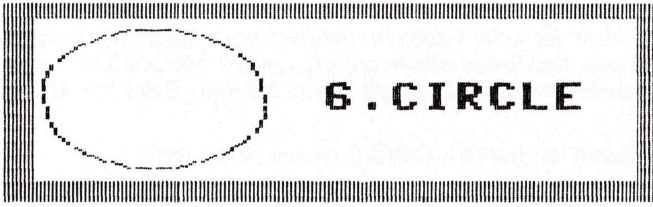
ERSTELLT MIT SIMON'S BASIC

Und hier sehen Sie eine praktische Nutzung:  
Diese Illustration zu einem Artikel

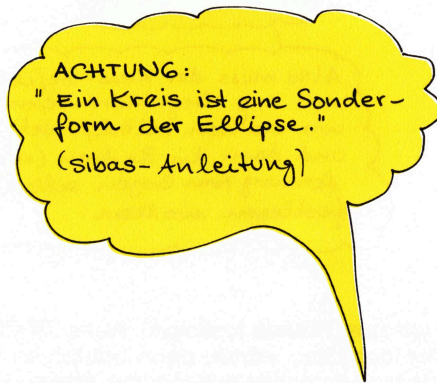
Obige Variante  
(Tabelle)  
Konnte im  
Handum-  
drehen als  
Übersicht für  
den Leser umge-  
stellt werden.  
\*Die SYS-Befehle  
beziehen sich auf  
das BASIC-KOALA  
Programm dieses  
Artikels.

SYS-BEFEHLE: KOALA PER BASIC	
SYS 52500 +BILD-NAME+,8	LOAD+SHOW Ende:Leertaste
SYS 52503	NUR ANZEIGE
SYS 52506	BILDSCH.WRML
SYS 52509 +BILD-NAME+,8	BILD wird nur geladen
IDEE+ARTIKEL: BIRKENBIHL PROGRAMM VON: M.M. MEISZL	





Falls Sie sich über den ovalen »Kreis« in der Überschrift zum Kreis(CIRCLE)-Befehl gewundert haben, darf ich auf einen Satz der *sibas*-Anleitung (Seite 30) hinweisen:





---

Nun kann es sein, daß Sie mit dem Aufbau gewisser geometrischer Figuren sehr vertraut sind. Aber wir wollen allen Anwendern von *sibas* helfen, daher gehen wir NICHT davon aus, daß Ihnen alle Angaben, die Sie bei CIRCLE machen müssen, sofort einleuchten werden. (Dies gilt auch bei den Befehlen ARC und ANGL, unten!)

Also, was müssen wir nun bei CIRCLE genau festlegen?

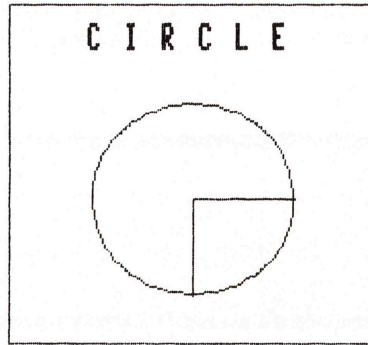
1. Mit  $x$  und  $y$  bestimmen wir einen spezifischen Punkt, nämlich den MITTELPUNKT des CIRCLE:



Jeder Kreis hat einen Mittelpunkt. Jeder Punkt kann durch  $x, y$  definiert werden!

Also muss der Kreis-Befehl mit  $x, y$  beginnen. Wenn wir erst den Mittelpunkt, und dann die Radien (=Entfernung zum Bogen selbst) festlegen wollen.

2. Nun müssen wir **zwei Radien** festlegen. Wenn CIRCLE nun wirklich »Kreis« im üblichen Sinne bedeuten würde, dann bräuchten wir nur EINEN Radius zu bestimmen, da ja beim richtigen Kreis der Abstand der Kreislinie zum Mittelpunkt an ALLEN gemessenen Stellen gleich ist:



Da aber der *sibas*-Kreis eine Sonderform der Ellipse darstellt, müssen wir zwei Angaben machen, denn bei Ovalen (= Ellipsen) sind diese 2 Werte ja auch unterschiedlich! Bei Kreis also 2\* die gleiche Zahl (bei HIREs; MULTI-Kreise, siehe unten):

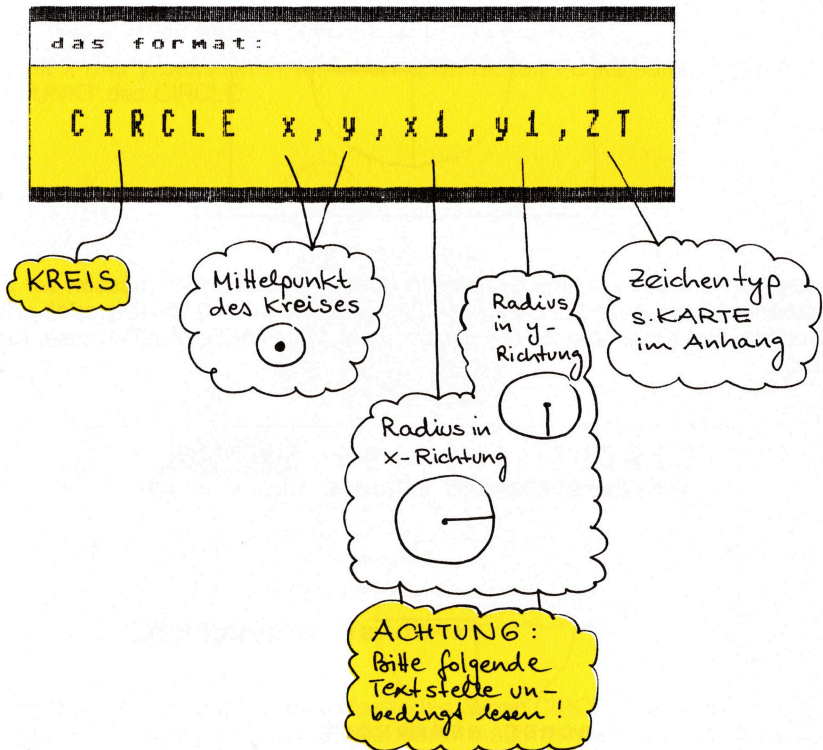
**CIRCLE 130, 100, 50, 50, 1**  
**GLEICHER RADIUS X-ACHSE UND Y-ACHSE**

X-ACHSE **WAAGERECHT**

Y-ACHSE **SENKRECHT**

**DAS ERGIBT EINEN NORMALEN KREIS**

Sollten Sie das Format in der *sibas*-Anleitung (und die Erklärung dort) mit dem Data Becker-Buch vergleichen, so finden Sie eigenartige Widersprüche:  
Hier das *sibas*-Format:



Laut *sibas* (Seite 30) ist der x-Radius der waagerechte, der y-Radius der senkrechte. Dies entspricht auch unseren bisherigen Erfahrungen auf dem Koordinatengitter.

Bei Data Becker jedoch heißt es (Seiten 245+246), daß der x-Radius senkrecht, der y-Radius hingegen waagrecht sei!

Wollen Sie dies testen:

## EXPERIMENT:

IST DER X-RADIUS  
bei CIRCLE waagerecht?

IST DER Y-RADIUS  
bei CIRCLE senkrecht?

Oder sollte es umgekehrt  
sein?

Bitte testen Sie dies:  
geben Sie einen Kreis-  
Befehl mit längerem  
X-Radius (und danach  
einen, mit längerem  
y-Radius) ein. Zum  
Beispiel, wie im folgenden  
Listing:

DISK

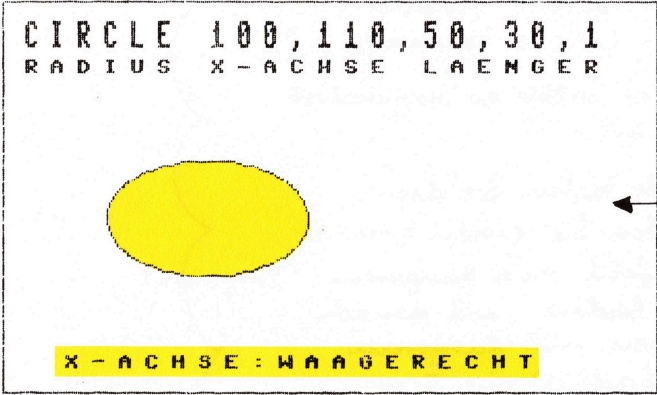
```
1 REM EXPERIMENT CIRCLE-RADIEN
2 REM INT+BIRKENBIHL,1984
4 COLOUR 14,14:HIRES 0,14
6 REC 0,0,319,199,1:REM RAHMEN
8 TEXT 10,10,"CIRCLE 100,110,50,30,1",1,2,13
10 TEXT 10,30,"RADIUS X-ACHSE LAENGER",1,1,13
12 TEXT 30,100,"X-ACHSE:WAAGERECHT",1,1,13
14 CIRCLE 100,110,50,30,1
16 COPY:Rem Bildschirmausdrue, s. Kapitelchen 5
18 :
20 COLOUR 14,14:HIRES 0,14
22 REC 0,0,319,199,1:REM RAHMEN
24 TEXT 10,10,"CIRCLE 100,110,30,50,1",1,2,13
26 TEXT 10,30,"RADIUS Y-ACHSE LAENGER",1,1,13
28 TEXT 165,100,"Y-ACHSE:SENKRECHT",1,1,8
30 CIRCLE 100,110,30,50,1
32 COPY:Rem siehe Rem in Zeile 16!
```





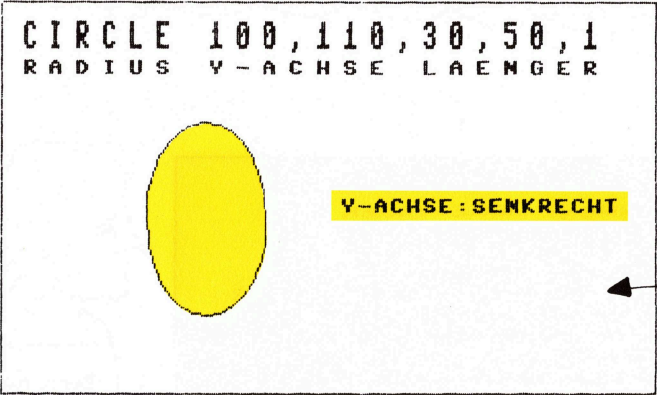
Nun, haben Sie's ausprobiert?  
Das LISTING der vorigen Seite ergibt:

Zeilen 1-16



HIRES

Zeilen 20-32



HIRES

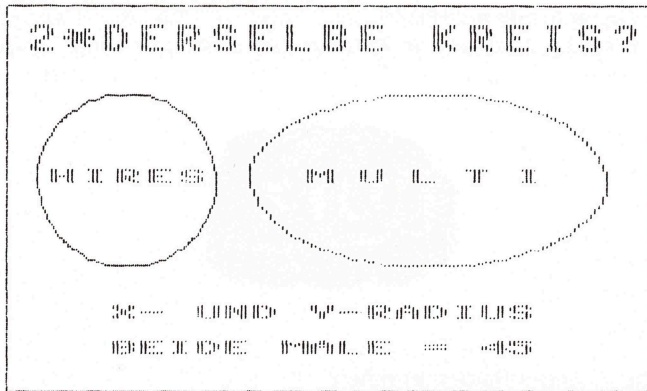
---

Alles bisher Gesagte galt nur HIRES. Jetzt überlegen Sie bitte: Wenn jeder MULTI-Punkt je 2 Pixel breit ist, wie muß dann der Befehl für einen richtigen Kreis aussehen?



1. Identische x- und y-Radien ergeben:
  - a) 2 Kreise (sowohl HIRES als auch MULTI)
  - b) HIRES: Kreis, MULTI: Ellipse
  - c) HIRES: Ellipse, MULTI: Kreis
2. x-Radius ist:
  - a) waagrecht
  - b) senkrecht
3. Welche Zeile ergibt einen HIRES-Kreis?
  - a) 50 CIRCLE 80, 80, 40, 30, 1
  - b) 50 CIRCLE 80, 80, 40, 40, 1
4. Welche Zeile ergibt bei MULTI einen Kreis?
  - a) 50 CIRCLE 80, 80, 80, 40, 1
  - b) 50 CIRCLE 80, 80, 40, 80, 1

ANTWORTEN: 1b, 2a, 3b, 4b



Hier sehen Sie es noch einmal ganz eindeutig:

HIRES:  $x$ - und  $y$ -Radius identisch = KREIS  
 MULTI:  $\frac{1}{2} x$ -,  $1y$ -Radius = KREIS

Daran ist klar abzulesen: Die komplizierten Angaben in der *sibas*-Anleitung, wie man einen Kreis macht, sind *zu* kompliziert!!

Wenn Sie mit CIRCLE komplexere Grafiken erstellen wollen, bietet es sich wieder an, mit FOR...NEXT-Schleifen zu arbeiten. Aber – hier gilt dasselbe wie bei REC: Falls Sie die Art von mathematischen Formeln der *sibas*-Beispiele (bei CIRCLE, Seite 30) noch nicht durchschauen – es geht auch mit einfacheren Zahlen.

Fühlen Sie sich angeregt, ein wenig mit diesem Befehl zu spielen?

---

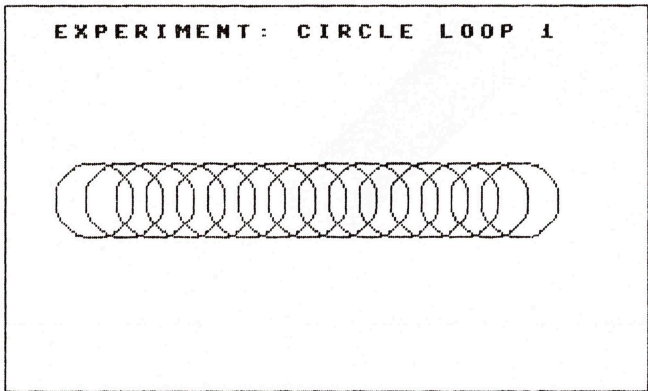
Hier folgen einige ANREGUNGEN

Zunächst: wir erhöhen den Wert der x-Achse, wandern also seitlich:



```
2 REM EXPERIMENT: CIRCLE LOOP 1
4 REM INT/BIRKENBIHL,1984
6 :
8 COLOUR 14,14:HIRES 0,14
10 REC 0,0,319,199,1:REM RAHMEN
12 :
14 TEXT 25,10,"EXPERIMENT: CIRCLE LOOP 1",1,1,10
16 :
18 X=45:Y=100
20 :
22 :::FOR ZR=1 TO 15
24 :::CIRCLE X,Y,20,20,1
26 :::X=X+15
28 :::NEXT ZR
30 :
32 :
34 WAIT 653,2:REM COPY

READY.
```





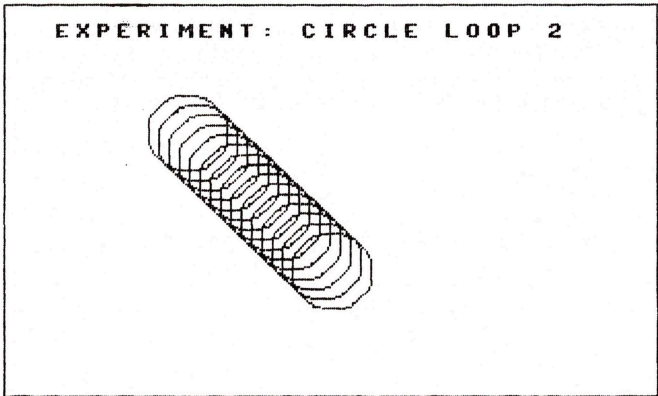
---

Diesmal erhöhen wir x (wandern also nach rechts) **und** y (nach unten). Aus der Kombination ergibt sich die schräge Anordnung:



```
2 REM EXPERIMENT: CIRCLE LOOP 2
4 REM IWT/BIRKENBIHL,1984
6 :
8 COLOUR 14,14:HIRES 0,14
10 REC 0,0,319,199,1:REM RAHMEN
12 :
14 TEXT 25,10,"EXPERIMENT: CIRCLE LOOP 2",1,1,10
16 :
18 X=90:Y= 65
20 :
22 :::FOR ZR=1 TO 15
24 :::CIRCLE X,Y,20,20,1
26 :::X=X+5:Y=Y+5
28 :::NEXT ZR
30 :
32 :
34 WAIT 653,2:REM COPY

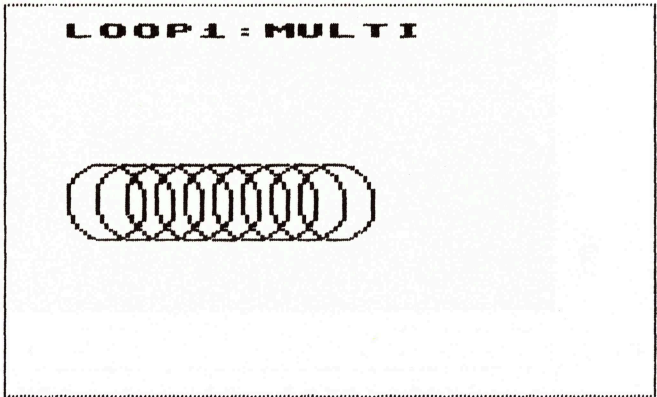
READY.
```



Probieren Sie mal, x1 oder y1 zu erhöhen!

# QUIZ:

WOLLEN SIE DAS EXPERIMENT LOOP 1 SO UMSCHREIBEN, DASS ES IN MULTI "funktioniert" ?



HIER ein Lösungsvorschlag :

Bezüglich ZT (Zeichentyp):  
Der wurde nur auf 3 geändert, wegen der HRDCPY-Ausgabe für die Illustration oben. (vgl. auch Kapitelchen 5 !)

Siehe Bemerkung rechts

```

2 REM EXPERIMENT: CIRCLE LOOP 1
4 REM IMT/BIRKENHILF,1984
6 :
8 COLOUR 14,14:HIRES 0,14:MULTI 5,6,0
10 REC 0,0,159,199, REM RAHMEN
12 :
14 TEXT 15,10,"LOOP1:MULTI",8,1,8
16 :
18 X=25:Y=100
20 :
22 : FOR ZR=1 TO 9
24 : : CIRCLE X,Y,10,20,8
26 : : X=X+7
28 : : NEXT ZR
30 :
32 :
34 WRIT 53,2:REM COPY
  
```

JETZT NOCHMAL  
ZU HIRES  
ZURÜCK, BITTE

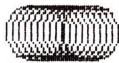
Wenn Sie alle Parameter (außer dem Zeichentyp) vorher festlegen (siehe ZEILE 16), können Sie diese jederzeit verändern, um zu sehen, was dann passiert. Beim Experimentieren sah ich, was hier unter »LOOP 3« auftaucht. Diese Form erinnerte mich an etwas...

Disk

```
2 REM EXPERIMENT: CIRCLE LOOP 3
4 REM IWT/BIRKENBIHL, 1984
6 :
8 COLOUR 14,14:HIRES 0,14
10 REC 0,0,319,199,1:REM RAHMEN
12 :
14 TEXT 25,10,"EXPERIMENT: CIRCLE LOOP 3",1,1,10
16 :::X=65:Y=55:X1=15:Y1=15
20 :
22 :::FOR ZR=1 TO 10
24 :::CIRCLE X,Y,X1,Y1,1
26 :::X=X+3
28 :::NEXT ZR
30 :
34 WAIT 653,2:REM COPY

READY.
```

EXPERIMENT: CIRCLE LOOP 3



Woran erinnert Sie das?

Und das hier kam dabei heraus:

1. Verrücken der »Sonnenbrille« nach unten + rechts
2. Circle Gesicht
3. Nase usw. (siehe LISTING)

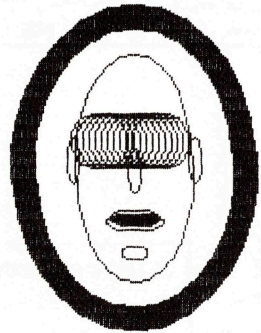
Das Schöne daran ist: Bis auf PAINT sind alle Grafik-Befehle nur CIRCLES!!

Disk

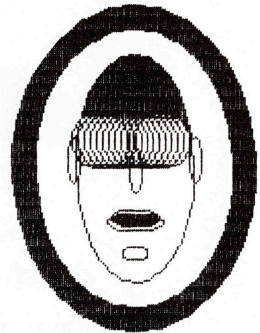
```
2 REM ZEICHNUNG MIT CIRCLE'S
6 REM IWT/BIRKENBIHL,1984
14 COLOUR 14,14:HIRES 0,14
22 :::X=135:Y=85:X1=15:Y1=15
26 :
30 :::FOR ZR=1 TO 11: REM SONNENBRILLE
34 :::CIRCLE X,Y,X1,Y1,1
38 :::X=X+3
42 :::NEXT ZR: REM SONNENBRILLE
46 CIRCLE 152,99,30,60,1: REM GESICHT
54 CIRCLE 152,100,4,13,1: REM NASE
58 CIRCLE 152,125,13,5,1: REM MUND
62 CIRCLE 152,127,15,5,1: REM MUND
63 PAINT 152,127,1
65 CIRCLE 152,143,7,4,1: REM KINN
70 CIRCLE 122,95,4,13,1: REM LINKES OHR
78 CIRCLE 182,95,4,13,1: REM RECHTES OHR
85 CIRCLE 152,99,50,73,1: REM RAHMEN IN
90 CIRCLE 152,99,62,85,1: REM RAHMEN AUS
94 PAINT 190,150,1
98 WAIT 653,2:REM COPY
```

READY.

EDITIER-MODUS:  
so gehen diese zeilen  
in Null-Komma-Nichts  
usw



Schaffen Sie's,  
ihm per  
PAINT 'ne  
schwarze  
Kappe aufzu-  
setzen?



Auflösung für Fauler →

Aber Sie probieren es doch zuerst mal?

47 PAINT 135,65,1



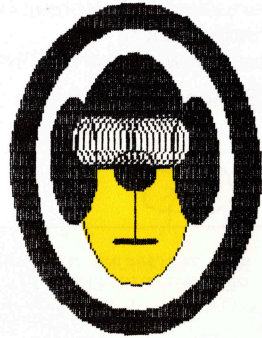
Tja, dann bellte plötzlich ein Hund nebenan, und da fiel mir Snoopy the Red Baron ein ...

Ein richtiger Snoopy ist's zwar nicht, aber doch ganz lustig, gell?

Nur mit CIRCLE, PAINT und 1 mal 1 LINE-Befehl!

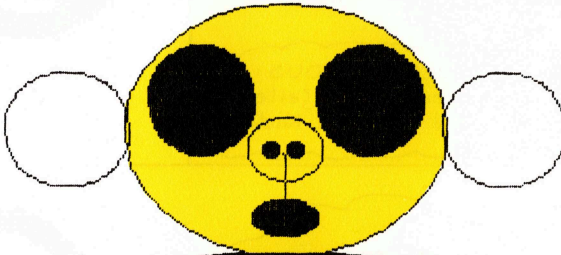
Und überhaupt: Die interessantesten Entdeckungen passieren beim Ausprobieren!

10 MINUTEN SPAETER



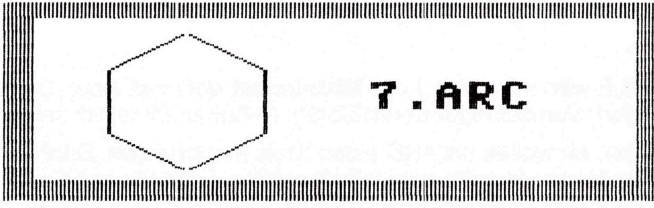
Listing im Anhang

DIESES PROGRAMM IST AUF DER DISKETTE



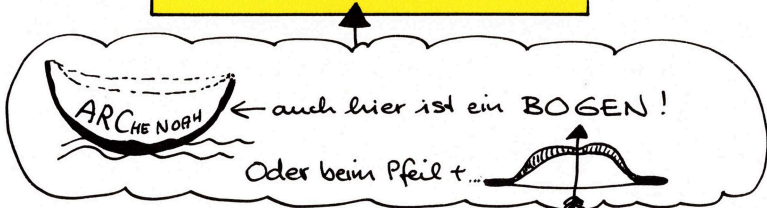
PANDA MIT CIRCLE, PAINT+LINE

Einfach drauflosprobieren!  
Das ist oft spannender als  
mandher Kriminalroman !!



Wenn man sich nie (oder schon seit Jahrzehnten nicht mehr) mit Geometrie beschäftigt hat, dann sind gerade die Befehle ARC und ANGL zunächst nicht ganz leicht. Aber, insbesondere der ARC-Befehl ermöglicht derart viele verschiedene Figuren, daß dieser Befehl unbedingt zu Ihrem Standard-Repertoire mit *sibas* gehören muß.

Zunächst einmal: **ARC** heißt »**Bogen**«:  
englisch: *ARCh*, französisch: *ARC*



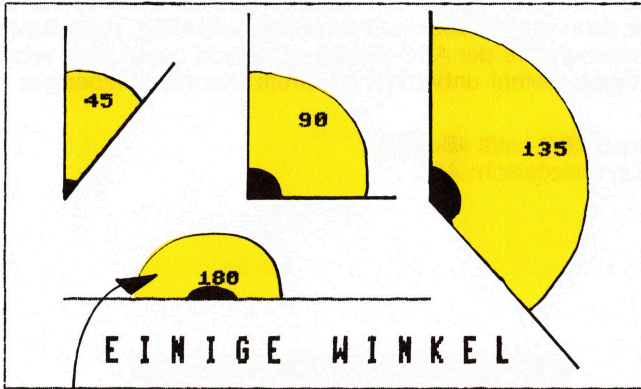
---

Weiter: Der Anfang des Bogen-Befehls ist sehr einfach:

ARC, x, y

Wie bei CIRCLE wird mit x und y der **Mittelpunkt** definiert. Aber: Der Mittelpunkt wovon? Antwort: Von der Figur, deren Bogen (= Außen-Linie) wir zeichnen wollen.

Nehmen wir an, wir wollen mit ARC einen Kreis konstruieren: Dann brauchen wir nach x, y als nächste Angabe zwei Winkelgrößen. Falls Sie seit Ewigkeiten (oder noch nie) einen Winkel gezeichnet haben sollten, hier sind einige:



Wenn  $180^\circ$  Ein Halb-Kreis (oder eine "halbe Figur") ist - dann leuchtet sicher ein, dass eine ganze Figur doppelt ( $2 \times 180$ ) also  $360^\circ$  (Grad) haben muss!

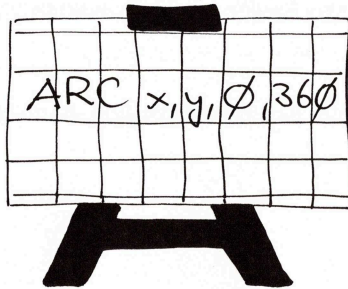
Um eine volle Figur zu konstruieren, die »ganz herum« geht (z. B. einen Kreis), müssen wir demnach folgende Winkelangaben machen:

START-Winkel: 0

END-Winkel: 360

---

(Wir kommen später darauf zurück, wie man Halb- oder Viertelfiguren zeichnet). So daß unser ARC-Befehl vorläufig so aussieht:



Nun kommt ein wichtiger Parameter: Der Abstand einzelner Punkte auf dem Bogen: Je kleiner die Abstandzahl (z. B. 4 oder 8) desto feiner wird die Linie. Aber: Jeder Punkt muß vom Computer berechnet und gezeichnet werden. Eine geringere Abstandzahl bewirkt eine sehr genaue Zeichnung, dauert aber lange.



Nehmen wir trotzdem für unseren ersten Versuch einmal Abstand 3! Damit Sie sehen, wie langsam dieser Kreis entsteht! Später werden wir andere Abstandzahlen testen.



Also sieht unser Kreis per ARC-Befehl jetzt so aus:

ARC x,y,0,360,3

Wenn Sie sich jetzt erinnern, wie Sie den CIRCLE-Befehl konstruieren müssen, dann wissen Sie, daß wir zwei Radien angeben mußten: für die x- und die y-Achse. So auch hier, also:

ARC x,y,0,360,3,x1,y1,

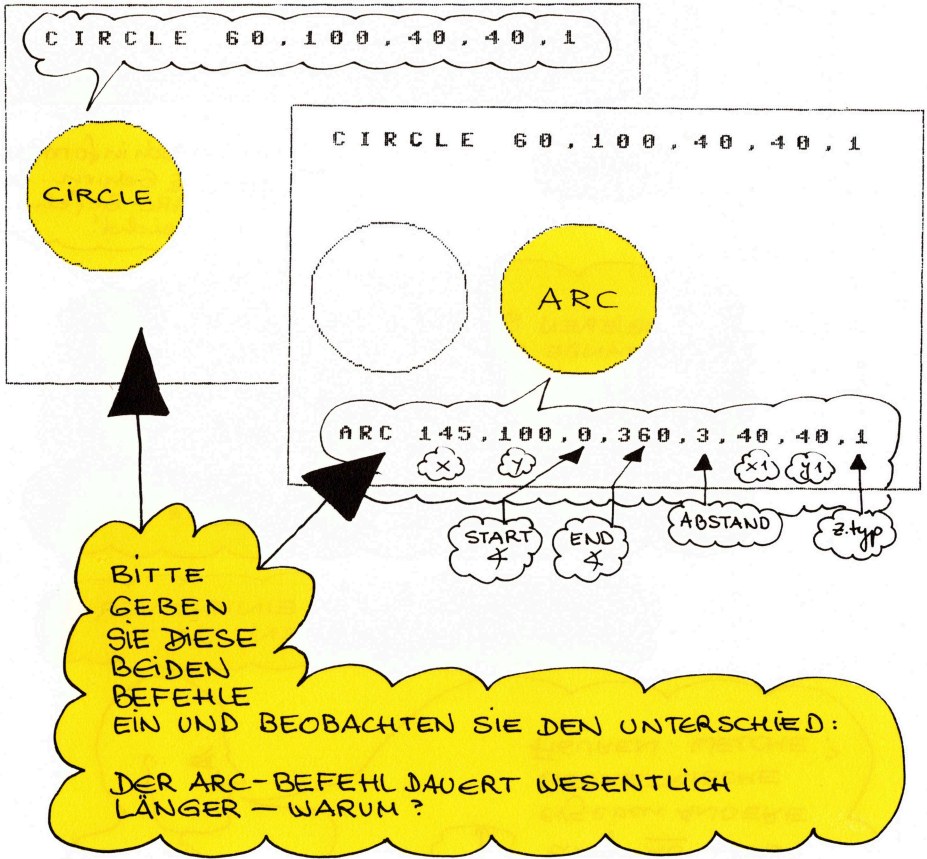
So, jetzt fehlt nur noch eine einzige Information, nämlich der Zeichentyp!

Achtung:  
Sie müssen den CIRCLE Befehl durchschaut haben, ehe Sie mit ARC beginnen !!

```
das format:  
ARC x,y,sw,ew,AB,x1,y1,ZT
```



Wenn Sie jetzt für x und y, sowie x1 und y1 Zahlenwerte einsetzen, dann erhalten Sie einen Kreis. **Denn: genaugenommen ist der CIRCLE-Befehl ein ARC-Befehl, in den einige der ARC-Parameter bereits eingebaut sind:**



ANTWORT: ABSTAND 3 bedeutet mehr Punkte auf der gezeichneten Linie; diese müssen jedoch sowohl BERECHNET als auch GEZEICHNET werden. Das aber dauert länger!

# EXPERIMENT: ARC

Dieses Experiment ist sehr wichtig!

```

2 REM EXPERIMENT ABSTAND BEI ARC
4 REM IWT/BIRKENBIHL, 1984
6 INPUT "WELCHER ABSTAND BITTE"; AB
8 COLOUR 14, 14: HIRES 0, 14: REC 0, 0, 319, 199, 1
10 TEXT 23, 25, "STARTWINKEL 0, ENDWINKEL 360", 1, 1, 10
12 ARC 125, 130, 0, 360, AB, 50, 50, 1
14 :
16 WAIT 653, 2: REM COPY
    
```

DISK

Dieser Versuch informiert Ihr ganzes Gehirn. So wird der ARC-Befehl leicht!

PROBIEREN SIE BITTE VERSCHIEDENE ABSTÄNDE AUS:


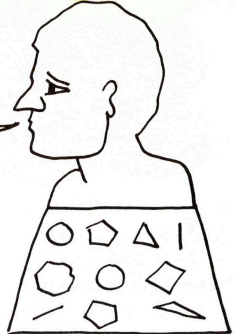
MINIMUM = 1 (sehr langsam!)

MAXIMUM = ?

Nach einem bestimmten Wert ergibt sich nur noch eine gerade Linie. Warum?\*

BITTE NICHT OHNE EINIGE VERSUCHE WEITERLESEN, JA ?

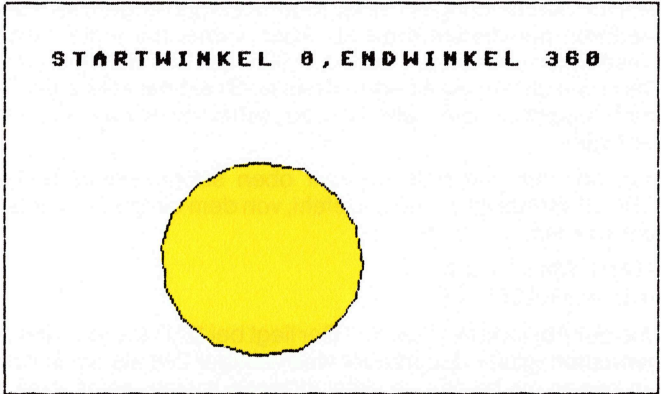
\* WEIL 180° eine Gerade ergeben. So darf die Figur dann nur noch erscheinen. Aber einige AB-Werte ergeben ANDERE GEOMETRISCHE FIGUREN. WELCHE ?

HIER  
2  
TEST-  
RUNs

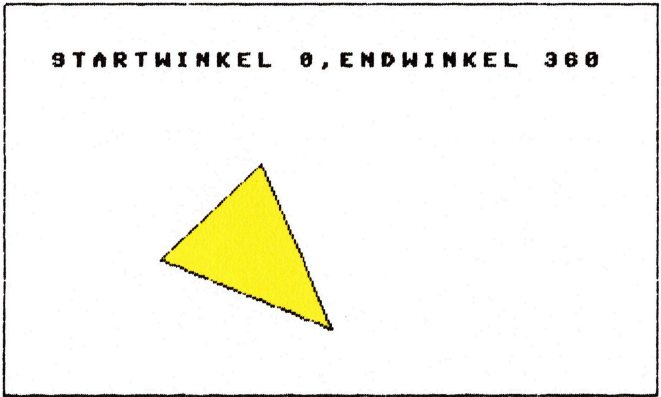
RUN

WELCHER ABSTAND BITTE? 23



Wenn Sie immer grössere AB-Werte eingeben, wird der Kreis immer ECKIGER, bis er plötzlich so wird:

WELCHER ABSTAND BITTE? 135





---

Das vorausgegangene Experiment sollte Ihnen eine gute Ahnung vermitteln über die Eigenart des Parameters AB bei ARC! Wieso kommt es zu so eigenartigen Figuren?

Eigentlich ist dies recht einfach:

Probieren Sie es einmal auf einem Blatt Papier aus: Zeichnen Sie einen Kreis, indem Sie die Linie punktieren. Zählen Sie anschließend die Punkte. Nehmen Sie jetzt ein durchsichtiges Papier (Butterbrotpapier geht im Notfall auch) und pausen Sie Ihren punktierten Kreis ab. Aber: Immer nur jeden zweiten Punkt abpausen. Diese Kopie enthält also nur noch 50% der Punkte. Wie sieht Ihre Zeichnung aus? Wenn Sie diesen punktierten Kreis noch einmal mit halber Punktezahl abpausen, dann beginnen Sie sehr klar zu sehen was passiert: Der Kreis wird immer »eckiger«!

Nun erinnern Sie sich, daß wir oben schon einmal festgestellt hatten: Jeder CIRCLE-Befehl ist ein ARC-Befehl, von dem einige Parameter automatisch festgelegt wurden:

START-Winkel auf 0

END-Winkel auf 360

Und der Abstand der Punkte? Der liegt bei 12! Dadurch wirkt der CIRCLE noch einigermaßen »glatt«, dauert aber weit weniger Zeit als unser Kreis mit Abstand 3 vorhin (wenn die beiden so entstandenen Kreise gleich groß sind).

Wieso kommt es aber zu so eigenartigen Figuren bei Arc, wie z. B. bei einem AB von 89?

(Wollen Sie den ausprobieren?)

TEILEN SIE DOCH MAL  $360 : 89$ .  
Sagt Ihnen das Ergebnis von

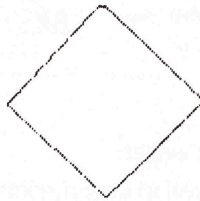
4.  $\emptyset 449438$

etwas? 4.....  $\rightarrow$  4 Ecken!

WELCHER ABSTAND BITTE? 89

Wieso  
kommen  
so  
"komische  
Figuren"  
bei Arc  
heraus,  
wenn der  
START  $\emptyset = \emptyset$   
und der  
END  $\emptyset = 360$   
war?

STARTWINKEL 0, ENDWINKEL 360



Vergleichen Sie:

$360 : 180 = 2$  (2 Punkte werden durch eine Gerade verbunden)

$360 : 120 = 3$  (Dreieck)

$360 : 90 = 4$  (Raute)

$360 : 72 = 5$  (Fünfeck)

$360 : 60 = 6$  (Sechseck)

$360 : 51 = 7.0588235$  (ein »Siebeneck«)

$360 : 45 = 8$  (Achteck)

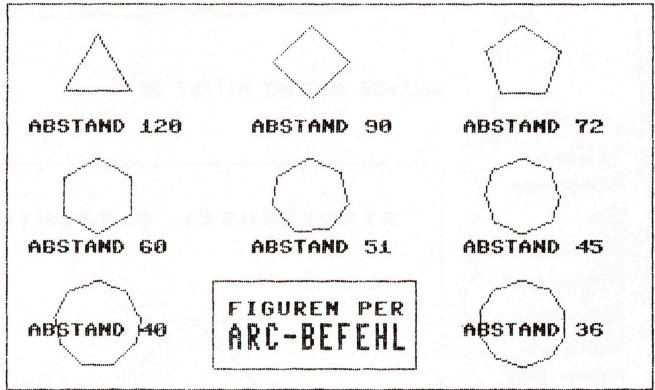
$360 : 40 = 9$  (Neuneck)

$360 : \text{weniger als } 35 = \text{Kreis(artig)}$

Hier das grafische Resultat unserer Berechnungen von eben: Diese Zahlen (AB) sollten Sie immer griffbereit halten. Daher befinden sich diese Figuren nochmal auf einer der Karten im Anhang.



Vgl. KARTE  
im Anhang

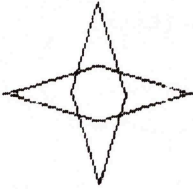


Jetzt gehen wir einen Schritt weiter:

Bis jetzt haben wir die Werte von  $x_1$  und  $y_1$  noch gleich gesetzt. Dadurch entstanden Figuren, die letztlich noch vage an den Kreis erinnern, der bei kleinem Abstand auch entstehen würde. Aber, Sie erinnern sich: Ist  $x_1$  größer als  $y_1$ , dann entsteht eine waagerechte Ellipse (beim Kreis). Ist  $y_1$  größer, dann entsteht eine senkrechte Ellipse! Dasselbe gilt für den ARC-Befehl.

**UNTERSCHIEDLICHE X1+Y1-WERTE ERÖFFEN INTERESSANTE RESULTATE:**

**1. SENKRECHTE RAUTE**



**2. WAAGERECHTE RAUTE**

- 1. X1=15; Y1=48
- 2. X1=48; Y1=15

der ARC-Befehl  
i = jeweils 90

i = ABSTAND  
(Sibas -  
Anleitung  
nennt AB-  
stand "i")

Verbinden wir wieder Gelerntes:

1. HIRES/MULTI
2. Rahmen REC (0,0,159,199,1)
3. LINE-Befehle
4. TEXT-Befehle
5. eine waagerechte Raute (siehe oben)
6. ein PAINT in dem mittleren langen REC, das durch die RAUTE unterbrochen wird. Der PAINT-Befehl ist doppelt, für den linken und für den rechten Teil.
7. INVERTIERTE Schrift im gePAINTeten REC:

Sie sehen wieder, wie vielfältig gerade auch der TEXT-Befehl ist!

HIER im MULTI-Modus

ENTWURFUNG : 20. 7. 1984

---

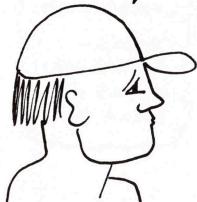
BURGWEG 7, DIEBORG

---

GEBURTSTAGS-PARTY



Wenn Sie oder ein Freund die nächste preiswerte Kleinanzeige planen, dann vergessen Sie die RAUTE nicht...



Na,  
Können  
wir  
mit  
sibas  
nicht  
schnell  
und  
leicht  
eine  
Menge  
Nützliches  
machen?



oder vielleicht so :







Aber noch immer haben wir nicht alle Möglich-  
keiten des ARC-Befehls ausgeschöpft :

Aber noch immer sind die Werte von START-Winkel (0) und END-Winkel (360) nicht geändert worden. Bitte, versuchen Sie dies nun: Welche Zahlen sind nötig, um einen Halbkreis (waagrecht bzw. senkrecht) zu machen; einen Viertelkreis? (Abstand 12).

Wie könnten Sie eine Raute nur zum Teil zeichnen, oder ein halbes Dreieck?

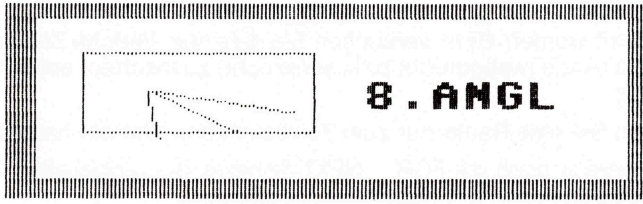
Und dann wäre ja noch die FOR...NEXT-Schleife, die wir bei allen Befehlen bis jetzt am Schluß ausprobiert haben.

Bitte experimentieren und Ergebnisse hier eintragen (bei AB = 12):

START #	END #	AB	FIGUR	Q U I Z  I  T A B E L L E
		12		
		12		
		12		
		12		

P.S. So eine vollständigere Liste könnte man sich erstellen, wenn man desöfteren mit Teilfiguren arbeiten will. Denn, dies gilt für alle AB-Werte (Dreieck, Viereck=Raute, 5-Ecke etc) auch !!

Sehen Sie, welche aufregenden Möglichkeiten der ARC-Befehl Ihnen bietet? Deshalb lohnt sich das Etwas mehr an Mühe, um ihn zu verstehen. Meinen Sie nicht auch?



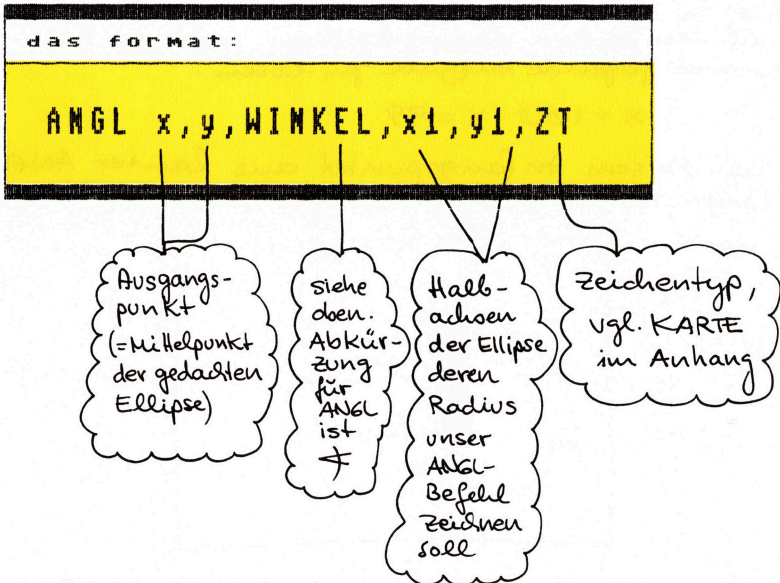
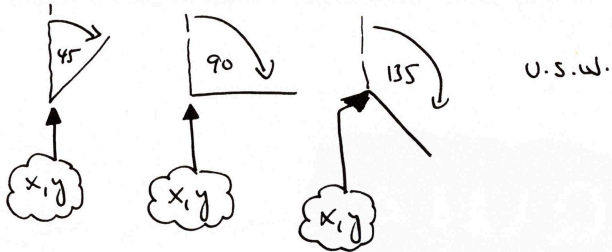
Eigentlich zeichnet dieser Befehl nur eine Linie: nämlich einen frei wählbaren Radius einer gedachten (oder vorher gezeichneten) Ellipse. Daher müssen Sie CIRCLE und ARC verstehen, denn bei ANGL handelt es sich um einen RADIUS (siehe bei CIRCLE) einer *gedachten* Ellipse, die aber nicht gezeichnet sein muß!!

Nun hat der ANGL-Befehl zwei Anwendungsgebiete:

1. Sie wollen geometrische Figuren zeichnen (so macht Geometrie sicher mehr Spaß, als mit Bleistift und Papier!). Die  $\alpha$ -Beispiele auf Seite 92 wurden mit ANGL gemacht.

Dabei müssen Sie wissen, daß der ANGL an dem von Ihnen definierten Punkt beginnt und wie folgt berechnet wird:

Auch hier hilft nur experimentieren!!



2. Sie wollen Grafiken entwerfen, wie sie beim *sibas*-Handbuch (Seite 33) oder zahlreich im Data Becker-Buch mit Beispielen vorgeführt werden. Hier sind FOR...NEXT-Schleifen absolut notwendig!!



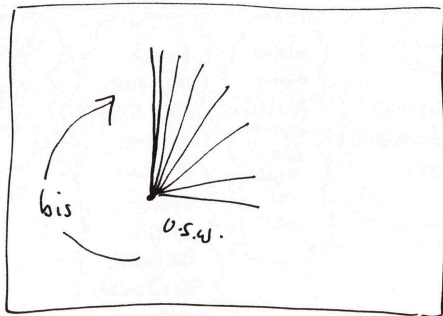
Denn: einzelne Linien könnten Sie mit dem LINE-Befehl auch ziehen! Aber mit ANGL ist es ja ein *Radius* (einer gedachten Ellipse), so daß strahlenartige, fächerartige Linien-Geflechte entstehen. Da bei der hochauflösenden Grafik die Farbe in 8\*8-Pixel-Blocks gehandhabt wird, sind Farben bei nahen Linien nie rein. Dadurch ergibt sich gerade bei solchen Grafiken ein **schillernder Regenbogeneffekt**, auch wenn Sie nur eine Farbe einsetzen. Aber das muß man selbst ausprobieren und testen. Die ausführliche Behandlung des ARC-Befehls zeigte Ihnen die Art des schrittweisen Probierens, die auch bei ANGL sinnvoll wäre. Verändern Sie immer nur einen Parameter und spielen Sie damit, bis Sie ihn im Griff haben. Dann experimentieren Sie mit einem weiteren Parameter usw.

## QUIZ:

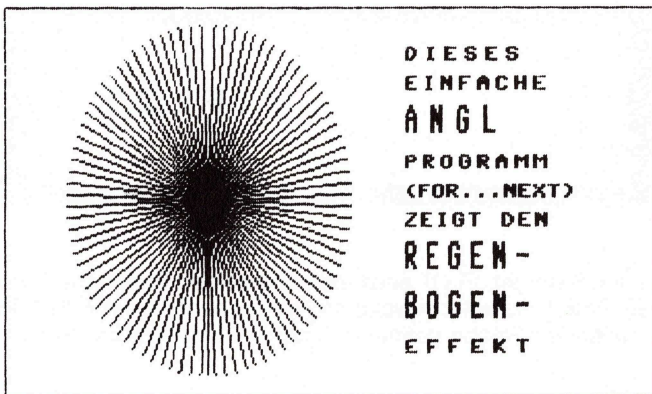
Ehe Sie die komplizierteren Beispiele der beiden erwähnten Bücher nachvollziehen, probieren Sie einmal folgende Aufgabe zu lösen:

$$X = 100 : Y = 99$$

Von diesem Anfangspunkt aus lauter ANGL-Linien, die einen fächerartigen "Kreis" bilden:



Wenn Sie dabei schwarzen HG und hellrote ZF wählen, dann können Sie den oben erwähnten "Regenbogen" schillern sehen: Das ist wirklich sehenswert. (LISTing + HRDCPY nächste Seite).



NA, schaffen Sie es, ohne Zeilen 12-24 im Listing anzusehen?

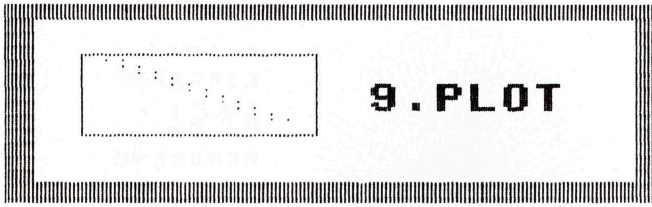
Disk

Die TEXT-ZEILEN wurden natürlich im EDITIER-Modus erstellt. Wenn Sie sie weglassen, dann erhöhen Sie den Wert von X, so dass die Figur in die Mitte des Rahmens bewegt wird...

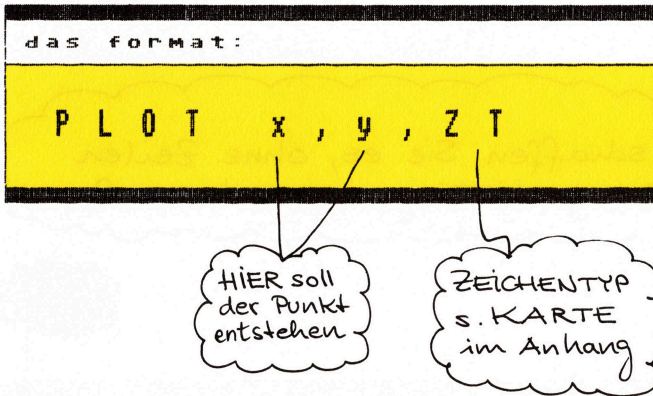
```

2 REM DEMO FUER ANGL MIT FOR...NEXT
4 REM IMT/DIRKENBILH
6 COLOUR0,0:HIRRES10,0:REM SCHWARZER HINTERGRUND MIT HELLROTER ZEICHENFARBE
8 REC 0,0,319,199,1
12 X=100:V=98/M=4
16 ::FOR ZR=1 TO 89
20 :::ANGL X,Y,M,I,70,90,1
22 :::MI=M+4
24 :::NEXT ZR
28 TEXT 195,20,"DIESES",1,1,10
30 TEXT 195,35,"EINFACHE",1,1,9
32 TEXT 195,50,"ANGL",1,2,13
34 TEXT 195,75,"PROGRAMM",1,1,9
36 TEXT 195,90,"(FOR...NEXT)",1,1,7
38 TEXT 195,105,"ZEIGT DEN",1,1,8
40 TEXT 195,120,"REGEN-",1,2,11
42 TEXT 195,145,"BOGEN-",1,2,11
44 TEXT 195,170,"EFFEKT",1,1,11
48 WAIT 653,2:REM COPY
READY
  
```

DIES IST DER ANGL-TEIL der die Fig. zeichnet

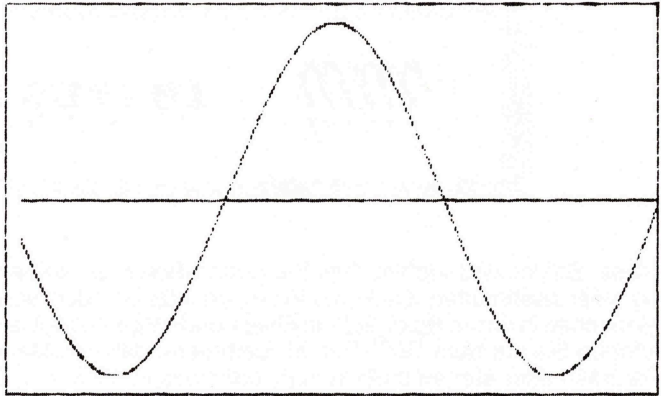


Jetzt wird es kinderleicht: PLOT setzt einen einzelnen Punkt. Die Parameter sind identisch zum PAINT-Befehl, bei welchem wir ja ebenfalls nur EINEN Punkt innerhalb der zu füllenden Fläche definieren mußten. Erinnern Sie sich?



Einzelne Punkte sind fast unsichtbar und wenig sinnvoll. Also werden Sie hier nur mit FOR...NEXT-Schleifen sinnvoll arbeiten können. Eine solche Schleife könnte eine gepunktete Linie bewirken; wobei diese Punkte auch so eng gesetzt werden können, daß eine Linie entsteht (vgl. AB-stand Experimente bei ARC)!

So sieht das PLOT-Beispiel aus der sibas-Anleitung aus.



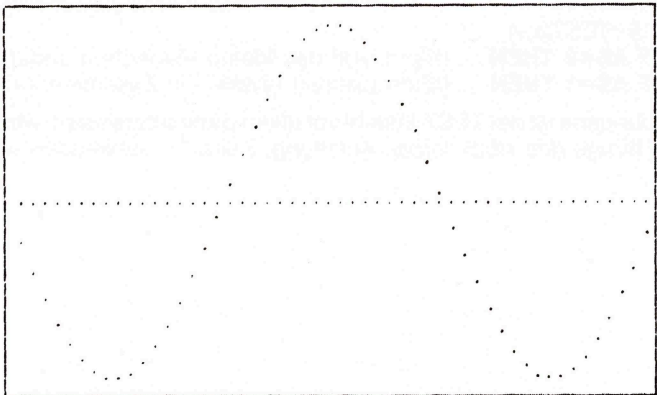
Dies ist das LISTING, Zeilen 10-18 exakt nach der Anleitung

```

2 REM DAS P L O T - B E I S P I E L A U S D E R S I B A S - A N L . ( S . 2 8 )
4 REM MIT EINIGEN KLEINEN AENDERUNGEN
6 :
8 COLOUR 14,14:HIRES0,14:REC 0,0,319,199,1
10 ::: FOR X=8 TO 320 STEP 0.5
12 ::: Y=100+SIN(X/34)*90
14 ::: PLOT X,Y,1
16 ::: PLOT X,100,1
18 ::: NEXT X
20 WAIT 653,2:REM COPY
  
```

# QUIZ:

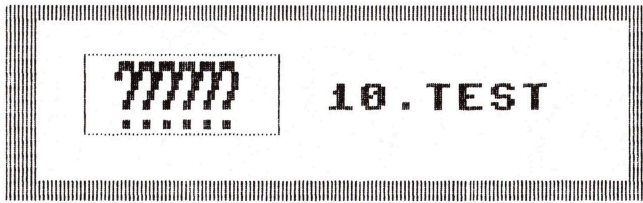
Welche Zeile muss wie verändert werden, damit die SINUS-Kurve von oben in etwa so wie hier aussieht ??



Falls Sie wirklich gucken wollen, die Lösung ...

```
10 ::: FOR X=8 TO 320 STEP 5
```





Dieser Befehl wird wichtig, falls Sie Action-Spiele schreiben wollen. Er testet, ob an einer bestimmten Stelle ein Punkt gesetzt ist oder nicht. Angenommen, ein Männchen in Ihrem Spiel jagt ein Etwas und es gibt eine Mauer im Gelände. Dann können Sie mit dem TEST-Befehl »befehlen«, daß das Männchen an der Mauer »abprallt« oder stehen bleiben muß, oder was auch immer. Der TEST-Befehl wird wie eine mathematische Formel gehandhabt:

Ist ein Punkt gesetzt, so »antwortet« der Computer mit 1; ist kein Punkt gesetzt, so »sagt« der Computer Null. Aber zuerst das Format:



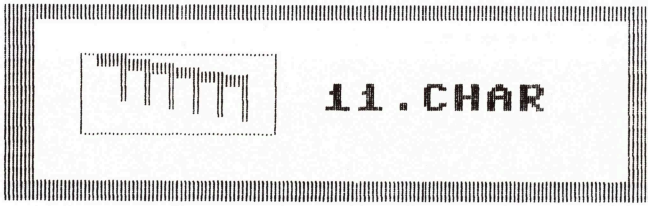
Also lauten dann die Gedankengänge für einen solchen Programmabschnitt zum Beispiel:

```
A$=TEST(x,y)
```

```
IF A$=0 THEN ... (dann läuft das kleine Männchen ungehindert weiter)
```

```
IF A$=1 THEN ... (dann passiert etwas: ein Zusammenprall zum Beispiel).
```

Übrigens ist der TEST-Befehl vor allem dann interessant, wenn Sie die JOYSTICK-Abfrage des *sibas* (*sibas*-Anleitung, Seite 71) verwenden wollen!



Wenn man den TEXT-Befehl bereits kennt, dann wundert man sich zunächst über den CHAR-Befehl, der **einen einzigen Charakter** auf den HIRES oder MULTI-Bildschirm bringt. Noch mehr wundert es einen dann, wenn die Data Becker-Autoren dazu schreiben (Seite 279):

»Ihre Graphiken stehen von nun an nicht mehr anonym im Raum. Sie können sie jetzt beschriften... Reden an Ihr Vaterland oder einfache, knappe Kommentare...«

Nun, eine Rede an mein Vaterland würde ich nur mit dem TEXT-Befehl schreiben, da ich bei TEXT jeweils eine ganze Zeichenkette unterbringen kann!

Also, was soll der CHAR-Befehl?

Nehmen wir an, daß sich der Autor von *sibas* etwas dabei gedacht hat, als er den CHAR – und den TEXT – schuf!

Also, vergleichen wir. Wichtigste Unterschiede:

TEXT	CHAR
Alle Zeichen (Tastatur) möglich: Groß – Klein Grafik REVERSE	Nicht alle Zeichen möglich: Keine Kleinbuchstaben
\$string wird über die Tastatur innerhalb von Hochkommas »gebaut«.	Zeichen wird durch POKE-Code angegeben.

Bei beiden kann das einzelne Zeichen nach unten verlängert werden (Parameter: TIEFE).

```
das format:
CHAR x,y,POKE,ZT,TI
```

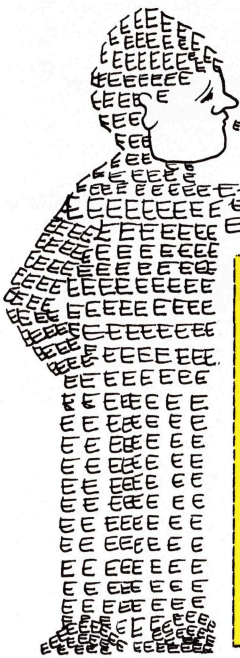
einzelner CHARakter

linke obere Ecke

Ø-255 s. Tabelle im Anhang

TIEFE wie bei TEXT

Zeichentyp s. KARTE im Anhang



Ein Beispiel

E = 5  
POKE

```
format:CHAR X,Y,POKE,ZT,TIEFE
[
hier: CHAR 155,55,5,1,8
```

# QUIZ:

disk

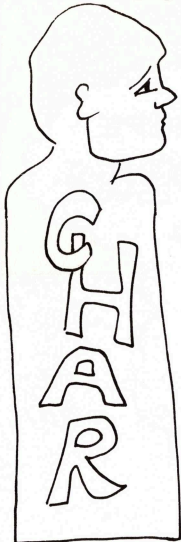
WIE MUSS  
ZEILE 20  
(ungefähr)  
LAUTEN, um  
unterschiedliches:

CHAR  
E

zu produzieren?

```

2 REM VGL.CHAR+TEXT:IMT/BIRKENBIHL, 1984
4 COLOUR14,14:HIRES0,14:REC3,0,310,196,1
6 TEXT 27,10,"CBM+GRAFIK",1,2,13
8 TEXT 27,30,"SHFT+GRAFIKX",1,2,13
10 TEXT 27,50,"REVERSE UND OHNE",1,2,13
12 TEXT 27,70,"KLEINBUCHSTABEN",1,2,13
14 REC 128,100,165, 86,1
16 CIRCLE 43,141,25,47,1
18 PRINT 44,141,1
20 CHAR ..... ?
22 TEXT 97,130,"E",1,6,8
24 TEXT 27,110,"CHAR",2,1,7
26 TEXT 85,110,"TEXT",1,1,7
28 :
30 TEXT 144,110,"VERGLEICH",1,2,14
32 TEXT 158,133,"SIBAS",1,2,19
34 TEXT 144,160,"CHAR/TEXT",1,2,14
36 WAIT 653,2: REM COPY
    
```



```

CBM+GRAFIK",1,2,13
SHFT+GRAFIKX",1,2,13
REVERSE UND OHNE
kleinbuchstaben
    
```

CHAR

TEXT



VERGLEICH  
s i b a s  
CHAR/TEXT

Sie brauchen die Lösung nicht, oder ?!  
Falls Sie jedoch vergleichen wollen,  
hier ist sie

20 CHAR 40,130,5,2,6



Ich persönlich finde CHAR hochinteressant um eigenartige Abbildungen zu schaffen. Diese hängen davon ab, wie Sie die einzelnen CHARaktere auf dem Bildschirm plazieren. Ein Input-Beispiel läßt Sie mit POKE-Werten und dem Parameter TIEFE experimentieren.

Disk

```
1 REM === DEMO: C H A R + INPUT (POKE/TIEFE)
2 REM COPYRIGHT: IWT/BIRKENBIHL, 1984
3 REM ***** C H A R A C T E R
4 COLOUR 0,0:PRINTCHR$(147)"XXXXXXXXXXXXXXXXXXXX":REM"X"=CBM+3;" " =LEERZEILEN
5 :
6 REM POKE=PO;TIEFE=TF
7 INPUT"POKE-WERT (0-255) ";PO
8 INPUT"X TIEFE (0-17)";TF
9 :
10 HIRSE7,0:REC 0,0,319,199,1:REM DIESES RECHTECK=BILDRAHMEN
11 :
12 MULTI 14,3,4
13 :
14 TEXT 3,20,"XCHAR X,Y,POKE,ZT,TIEFE",1,2,7
15 :
16 CHAR 60,50,PO,1,TF
17 CHAR 66,55,PO,2,TF
18 CHAR 72,60,PO,3,TF
19 CHAR 78,65,PO,1,TF
20 CHAR 84,60,PO,2,TF
21 CHAR 90,55,PO,3,TF
22 CHAR 96,50,PO,1,TF
23 :
24 :
25 :
26 WAIT 653,2
```

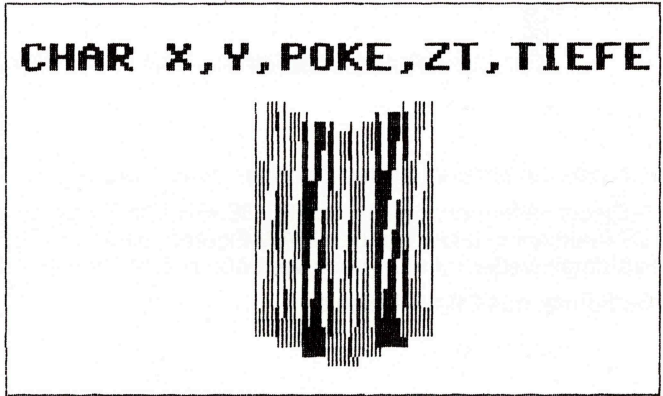
Da T1 als Variablenamen besetzt ist, heißt T1 hier TF.

INPUT : POKE + TIEFE  
CHAR-EXPERIMENT

Auf der nächsten Seite sehen Sie zwei TEST-RUNS :

POKE-Wert (0-255)? 165

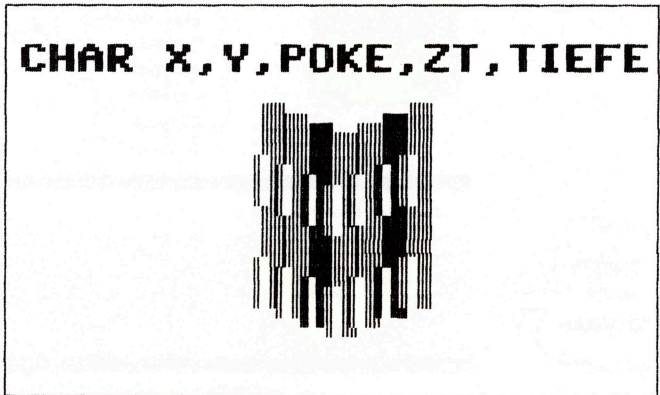
TIEFE (0-17)? 15

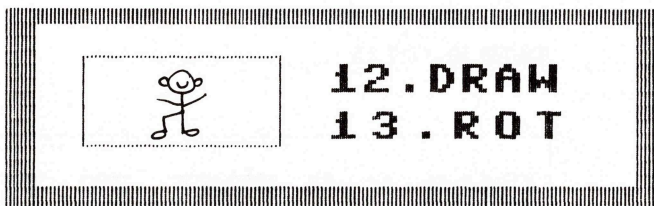


POKE-Wert (0-255)? 220

TIEFE (0-17)? 13

Theoretisch ist eine TIEFE bis zu 255 möglich, aber sinnvoll ist das nicht! Finden Sie selbst heraus, welches max. jeweils sinnvoll ist. Das hängt von der y-Position am Bildschirm ab!

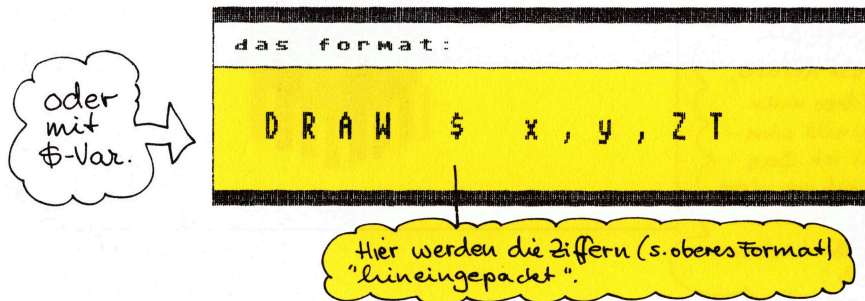
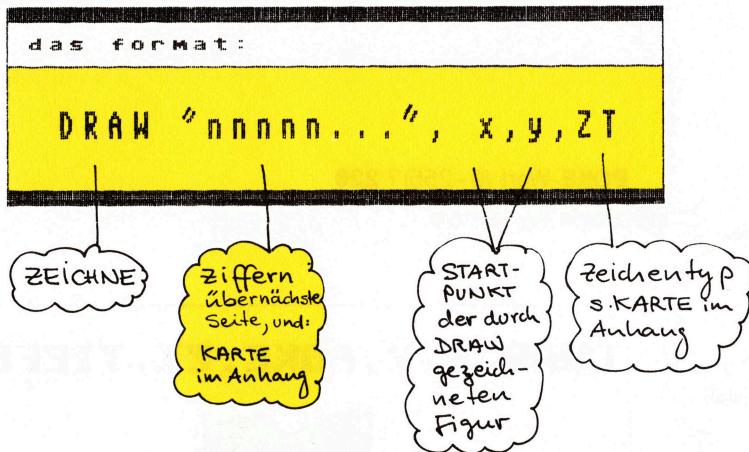




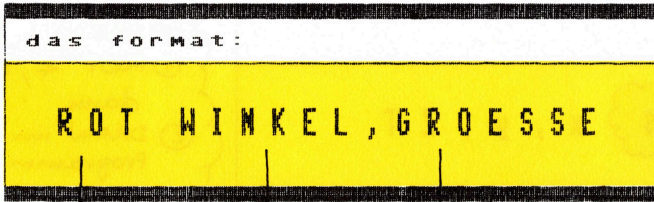
12. Dieser Befehl erlaubt das Zeichnen einer Figur.

13. Dieser Befehl ermöglicht das ROTIEREN und Vergrößern der DRAW-Figur. ROT funktioniert leider nur bei DRAW-Figuren; einen CIRCLE oder ARC etc. kann man damit weder rotieren noch vergrößern. Schade eigentlich.

Das Format des DRAW-Befehls:



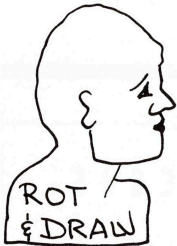
Und das FORMAT des ROT-Befehls:



RO-  
TA-  
TI-  
ON

Leider wird hier nicht, wie bei ARC oder ANGL ein Winkel angegeben, sondern eine Code-Ziffer. S. nächste Seite + KARTE im Anhang  
Ø - 7

Dieser Faktor vergrössert die gesamte Figur in alle Richtungen.  
1 = so, wie im DRAW-String festgelegt  
2 = doppelte Grösse  
3 = dreifache Grösse  
u.s.w.  
theoretisch bis 255,  
praktisch meist weit weniger.



Achtung: DRAW KANN OHNE vorherigen ROT-Befehl nicht angewandt werden!  
(Resultat wäre nur ein einziger Punkt).  
Wollen Sie ohne ROTATION in Originalgrösse DRAWen, dann bitte vorher ein ROT Ø,1 setzen, damit Sie was sehen!

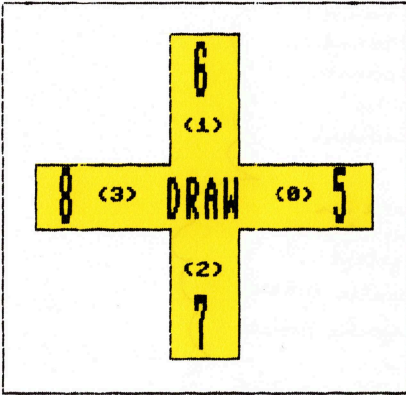


Jetzt die Ziffern für den DRAW-String und für den ROTations-Winkel:

```
das format:
DRAW $ x , y , Z T
```

**NICHT VERGESSEN:**

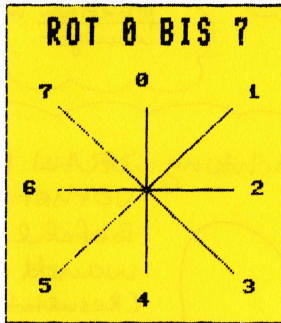
- ① ROT  $\neq$ , GR dann
- ② DRAW im Programm!



Die grossen Ziffern ZEICHNEN in der vorgegebenen Richtung.  
 Die Ziffern in Klammern verrücken den "Stift" unsichtbar; dienen also dem Absetzen und Neuansetzen des "Stiftes".  
 Die (2) und die (4) haben identische Wirkung, also reicht eine Angabe in unserem Schaubild.  
 Die (9) beendet einen \$, muss aber nicht gesetzt werden.

$\emptyset$  = Keine ROTation  
 1 = 45°  
 2 = 90°  
 3 = 135°  
 4 = 180°  
 5 = 225°  
 6 = 270°  
 7 = 315°

Dies-selben  $\neq$  haben wir bei ARC kennengelernt.  
 Hier leider nur als Code-Ziffer.

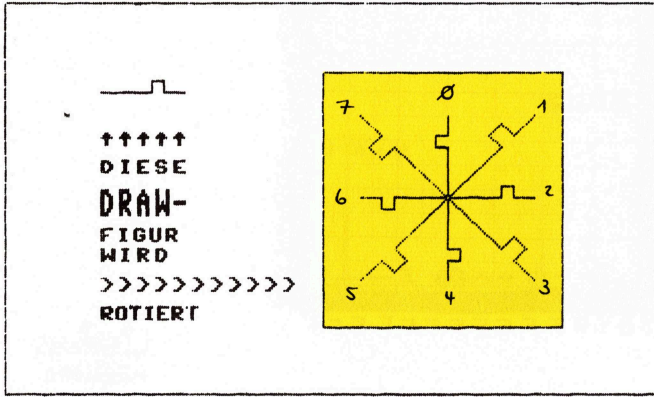


HIER WURDE EINE LINIE VON  $\emptyset$ -7 ROTIERT

```
das format:
ROT WINKEL, GROESSE
```

# ACHTUNG

Bei den »schrägen« ROTations-Lagen (1,3,5,7) wird die Figur etwas in die Länge gezogen. Das liegt daran, daß ein Pixel am Bildschirm etwas höher als breit ist. Dieser minimale Unterschied wird insbesondere bei einer (vergrößerten) Figur wie bei DRAW sichtbar:



```

2 REM ROT UND DRAW: VERZERRUNG BEACHTEN
6 REM INT/BIRKENBIHL, 1984
10 :
14 COLOUR 14,14:HIRES 0,14
18 REC 0,0,319,199,1
22 A$="55555555555555555555555555555555"
30 ROT 0,2: DRAW A$,45,45,1
    
```

Wollen Sie den A\$\*, der in obiger Abbildung rotiert wird, selber zu konstruieren versuchen?

Unten sehen Sie den Rotations-Teil des Listing's, links ist die Konstruktion, sowie das einmalige Zeichnen (links oben in der Abbildung) zu sehen, falls Sie nachsehen wollen!

\*Eine noch einfachere Figur gibt's auf der nächsten Seite, falls Sie noch nicht klar kommen sollten...

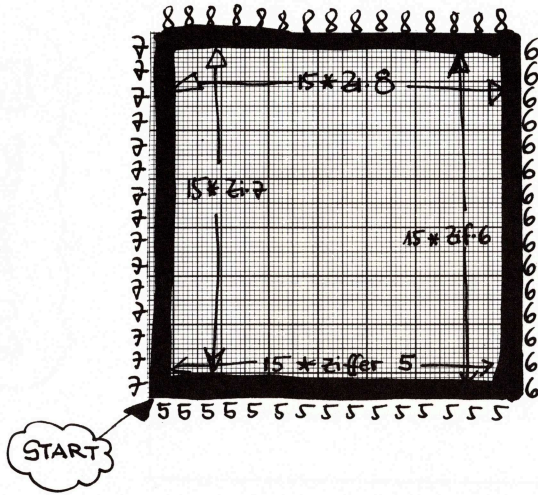
```

54 :
58 : : : FOR RO=0 TO 7
62 : : : ROT RO,2: PAUSE 1
66 : DRAW A$,215,99,1
70 : : : NEXT RO
74 :
75 : REC 155,35,116,130,1
78 WAIT 653,2: REM COPY
    
```

Hier zeige ich Ihnen, wie ich meine DRAW-\$'s immer konstruiere:

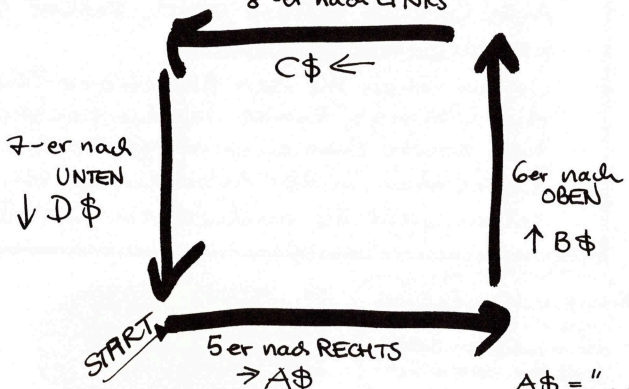
1. Zeichnen (auf kariertes oder mm-Papier)
2. Nummern auf Papier (+\$-Namen) schreiben
3. Eintippen

Die erste Zeichnung werden wir mit Größe 1 (also *ohne* Vergrößerung) konstruieren und zeichnen, damit Sie ein Gefühl dafür bekommen, wie viele gleiche Ziffern für ein sehr kleines Stück gerader Linie nötig sind:



LISTING  
nächste  
Seite

DER "STIFT" WIRD ALSO SO GEFÜHRT :  
8 -er nach LINKS



Jetzt kann eingetippt werden :

A\$ = " . . . . . " (15\*5)  
B\$ = " . . . . . " (15\*6)  
u.s.w.



Disk

```

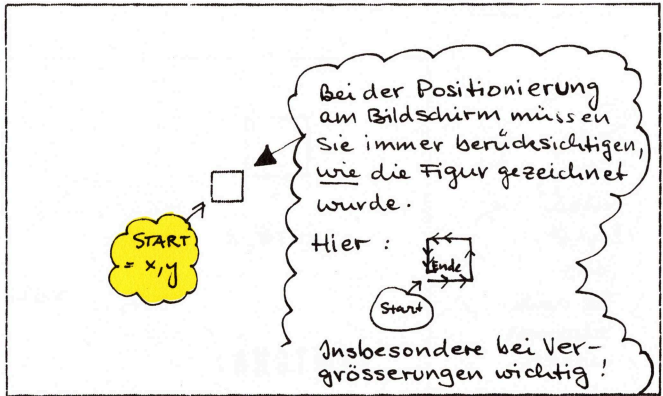
2 REM DRAW-DEMO 1: IWT/BIRKENBIHL, 1984
4 COLOUR 14,14:HIRES 0,14
6 :
8 A$="5555555555555555":REM 15*NACH RECHTS
10 B$="6666666666666666":REM 15* RAUF
12 C$="8888888888888888":REM 15* NACH LINKS
14 D$="7777777777777777":REM 15*RUNTER
18 REM DR$=DRAW-STRING
20 DR$=A$+B$+C$+D$ ← s. unten
21 :
22 ROT0,1:REM ROT KEINE,GRÖSSE 1=KEINE VERGRÖSSETERUNG
23 :
24 DRAW DR$,100,100,1
26 :
28 WAIT 653,2:REM COPY

READY.

```

s. unten

Achtung:  
Auch wenn Sie die Diskette zum Buch besitzen: Falls Sie mit DRAW arbeiten wollen, wäre ein exaktes Nachvollziehen dieser Zeichnung sinnvoll, wenn Sie die Quizaufgabe davor nicht lösen konnten...



Wenn Sie mehrere kurze Strings definieren, müssen Sie diese noch zu einem Gesamtstring zusammensetzen:

siehe Zeile 20 !

Max. können 256 Ziffern zu einem Gesamtstring zusammengebaut werden. Aber in der Regel reicht das, wenn wir statt "5555555555555555" später mit dem Vergrößerungsfaktor arbeiten!



Jetzt probieren Sie bitte folgendes mit dem Programm von der letzten Seite:

① Einmal mit Vergrößerung (2)

② Einmal mit Rotation (1),

so dass die Figuren so erscheinen:

① linke Figur

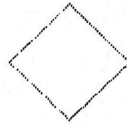
② rechte Figur.



Das Listing unten hilft, falls Sie nachschauen wollen.



ROT0,2



ROT1,2

**ACHTUNG:**

**ROT 0 + 1 & GROESSE 2**

Lösung für Faule ...

(Aber Sie sind das ja nicht, oder ?!)

Achtung: Durch ein RENUMBER passen die Zeilennummern nicht zu dem des vorigen Programms. Aber Sie können ja nochmal RENUMBERN!

```
20 ROT0,2
22 DRAW DR$, 60, 60,1
24 TEXT 54,89,"ROT0,2",1,1,8
26
28 ROT1,2
30 DRAW DR$,170, 60,1
32 TEXT 180,119,"ROT1,2",1,1,8
34
```

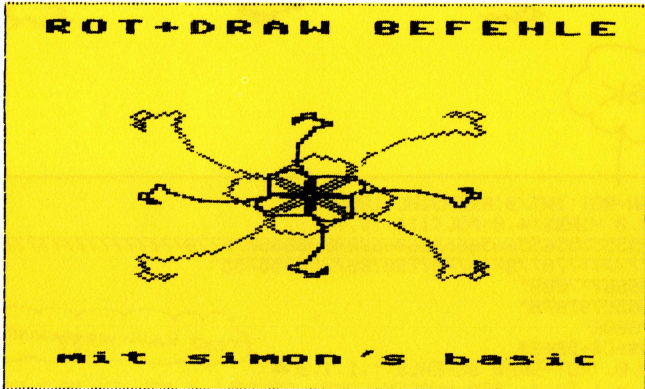
Vielleicht wundern Sie sich darüber, daß wir bis jetzt »unnötige« Ziffern-Wiederholungen gemacht haben. Ein

A\$ = »5687«

hätte, mit starker Vergrößerung, auch ein Quadrat ergeben. Richtig. Aber wenn Sie Figuren mit geschwungenen Bögen entwerfen wollen, müssen Sie diese etwas größer anlegen, sonst werden die Linien zu eckig (Treppen statt leichtem Schwung). Etwas »eckig« wird die Linie beim Computer immer (bei einer Auflösung von max. 320\*200 Punkten), aber man kann den Effekt mildern.

```
5 A$="5555555555555666666688888888887878777777777775555755777775"
8 B$="87777777778787877787878787795795755"
10 C$="555666777888"
12 D$="5556667787878"
14 E$="5778886"
16 M$=A$+B$+C$+D$+E$
```

Falls Sie das Buch nicht seitlich legen wollen, auf der nächsten Seite ist ein kleines Programm; dort sind es die Zeilen 3-9, die den string ergeben.

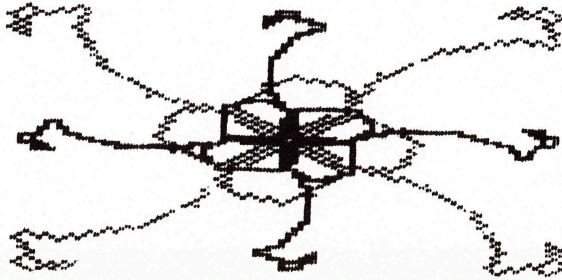


Diese Figur wurde 7 x rotiert.  
 Wenn Sie den \$string (linkes) einmal eintippen und dann mit  
 ROT Ø, 1 : DRAW MØ, 76, 99  
 einmal ohne ROTATION zeichnen, dann sehen Sie die Basisfigur. Bitte das Ergebnis Ihrer Tipparbeit SAVEIN (und vor das zeichnen noch  
 COLOUR Ø, Ø : HIRRES 14, Ø : MULTI 14, Ø, 7 setzen.

Es folgt ein komplettes Programm, das folgendes bewirkt:

1. DRAWING der Figur
2. Farbenspiel mit derselben  
(es gibt einen »rad-artigen« Effekt).

Das »Feuerrad« müssen Sie am Bildschirm gesehen haben, das können Worte (für Herrn Links im Gehirn) dem Herrn Rechts eben *nicht* beschreiben...



```
1 REM DRAW+ROT:IWT/BIRKENBIHL,1003
2 COLOUR0,0:HIRES14,0:MULTI14,10,7
3 A$="55555555555565665666666686688888888887877877777777777775757577757775"
4 B$="8777777777877877877877887887887887755755755"
5 C$="555666777888"
6 D$="565656787878"
7 E$="5778886"
9 M$=A$+B$+C$+D$+E$
12 TEXT10,0,"ROT+DRAW BEFEHLE:",1,1,9
13 :FOR I = 1 TO 4 :REM AUSSEN-LOOP: ES SOLL 4 MAL IN JE 3 FARBEN LAUFEN
15 :: FOR X = 0 TO 7
16 :: ROT X,1
17 :: DRAW M$,76,99,2:REM DRAW MIT ZEICHENFARBE 2
18 :: NEXTX
19 :
20 :: FOR X = 0 TO 7
21 :: ROT X,1
22 :: DRAW M$,76,99,1:REM DRAW MIT ZEICHENFARBE 1
23 :: NEXTX
24 :
25 :: FOR X = 0 TO 7
26 :: ROT X,1
27 :: DRAW M$,76,99,3:REM DRAW MIT ZEICHENFARBE 3
28 :: NEXTX
30 :NEXT I:PAUSE1:REM AUSSEN LOOP : BIS HIERHER SOLL ALLES 4 MAL LAUFEN
31 TEXT 13,183,"@MIT SIMON'S BASIC",1,1,8
32 WAIT 653,2
```

TEXT kann weggelassen werden.

TEXT, s. oben









## Kapitelchen

### 4

## BEISPIEL-PROGRAMME

### KAP. 4 BESTEHT AUS 2 TEILEN:

#### TEIL I :

ANALOG ZU KAP. 3 (ZU JEDEM DORT  
BESPROCHENEN BEFEHL) JE EIN  
LISTING PLUS BILDSCHIRMAUSDRUCK;  
UND WIEDER: JE EIN MINI-QUIZ!  
LOESUNGSVORSCHLAEGE IM ANHANG,  
ABER DAS SCHAFFEN SIE AUCH SO, ODER?!

#### TEIL II :

EIN GROSSES DEMO-PROGRAMM, IN DEM  
(FAST) ALLE BEFEHLE VON KAP. 3  
BUNT GEMISCHT EINGESETZT WURDEN.  
DIESES PROGRAMM ZEIGT DIE BEFEHLE  
GEHIRN-GERECHT: WIE IN EINEM TRICKFILM!

1. REC = RECHTECK

```
REC 40, 40, 199, 119, 1  
REC X, Y, X1, Y1, ZT
```

NICHT VERGESSEN:

```
(X1, Y1): REC=RELATIV
```

QUIZ: MACHEN SIE EIM QUADRAT DRAUS!



```
1 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V.KAP.3
2 REM INT/BIKENBIHL, 1984
3 :
4 REM GOSUB (TITEL FUER DIES PROGRAMM)
5 :
6 COLOUR14,14:REM RA,HG HELLBLAU
7 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
8 REC 0,0,319,199,1: REM RAHMEN-RECHTECK
9 :
10 :
11 REM ++++++
12 REM ++ DER R E C - BEFEHL ++
13 REM ++++++
14 :
15 REM ::::: REC=RELATIV :::::
16 :
17 TEXT 40,15,"1.REC = RECHTECK",1,1,8
18 REC 40,40,199,119,1
19 TEXT 60,60,"REC 40,40,199,119,1",1,2,8
20 TEXT 60,80,"REC X, Y, X1, Y1,Z",1,1,8
21 TEXT 60,110,"NICHT VERGESSEN:",1,1,9
22 TEXT 60,130,"(X1,Y1):REC=RELATIV",1,2,8
23 TEXT 17,180,"QUIZ:MACHEN SIE EIN QUADRAT DRAUS!",1,1,8
24 WAIT 653,2: REM COPY
25 :
```



2. BLOCK = AUSGEFUELLTES REC

```
BLOCK 40, 40, 199, 119, 1  
BLOCK X, Y, XL, YL, ZT
```

NICHT VERGESSEN -

```
(XL, YL): BLOCK=ABSOLUT
```

QUIZ: MACHEN SIE EIN QUADRAT DRAUS!



```
27
28 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
29 REM INT/BIRKENBIHL, 1984
30
31 REM GOSUB (TITEL FUER DIES PROGRAMM)
32
33 COLOUR14,14:REM RA, HG HELLBLAU
34 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
35 REC 0,0,319,199,1: REM RAHMEN-RECHTECK
36
37
38 REM ++++++
39 REM ++ DER B L O C K - BEFEHL ++
40 REM ++++++
41
42 REM : : : : : BLOCK=ABSOLUT : : : : :
43
44 TEXT 40,15,"2.BLOCK = AUSGEFUELLTES REC",1,1,7
45 BLOCK 40,40,239,159,1
46 TEXT 50,60,"BLOCK 40,40,199,119,1",2,2,8
47 TEXT 50,80,"BLOCK X, Y, X1, Y1,ZT",2,1,8
48 TEXT 50,110,"NICHT VERGESSEN:",2,1,9
49 TEXT 50,130,"(X1,Y1):BLOCK=ABSOLUT",2,2,8
50 TEXT 17,180,"QUIZ:MACHEN SIE EIN QUADRAT DRAUS!",1,1,8
51 WAIT 653,2: REM COPY
52
```

3. LINE = LIMIE

LINE 40, 40, 199, 40, 1  
LIME X, Y, X1, Y1, ZT

MICHT VERGESSEM:

(X1, Y1):LINE=ABSOLUT

QUIZ: 5 PARALLELE LIMIEM!





```
54  
55 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3  
56 REM INT/BIRKENBIHL, 1984  
57 :  
58 REM GOSUB <TITEL FUER DIES PROGRAMM>  
59 :  
60 COLOUR14,14:REM RA,HG HELLBLAU  
61 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU  
62 REC 0,0,319,199,1: REM RAHMEN-RECHTECK  
63 :  
64 :  
65 REM ++++++  
66 REM ++ DER L I N E - BEFEHL ++  
67 REM ++++++  
68 :  
69 REM : LINE=ABSOLUT :  
70 :  
71 TEXT 50,15,"3.LINE = LINIE",1,1,9  
72 LINE 40,40,239, 40,1  
73 TEXT 50,60,"LINE 40,40,199,40,1",2,2,8  
74 TEXT 50,80,"LINE X, Y, X1, Y1,ZT",2,1,8  
75 TEXT 50,110,"NICHT VERGESSEN:",2,1,9  
76 TEXT 50,130,"(X1,Y1):LINE=ABSOLUT",2,2,8  
77 TEXT 20,180,"QUIZ: 5 PARALLELE LINIEN!",1,1,10  
78 WAIT 653,2: REM COPY  
79
```



4. PAINT = ANMALEN

PAINT T, T, ZT  
PAINT X, Y, ZT

NICHT VERGESSEN :

(X, Y) : IN DER VORGESEHENEN FLÄCHE

QUIZ : MOHL KAUM NOETIG!

```
81 :
82 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
83 REM INT/BIRKENBIHL, 1984
84 :
85 REM GOSUB <TITEL FUER DIES PROGRAMM>
86 :
87 COLOUR14,14:REM RA,HG HELLBLAU
88 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
89 REC 0,0,319,199,1: REM RAHMEN-RECHTECK
90 :
91 :
92 REM ++++++
93 REM ++ DER P A I N T - BEFEHL ++
94 REM ++++++
95 :
96 REM X,Y: IRGENDWO INNERHALB DER ZU FUELLENDEN FLAECHEN!
97 :
98 TEXT 40,25,"4. PAINT = ANMALEN",1,1,9
99 REC 10,10,309,189,1:PRINT 7,7,1
100 TEXT 40,60,"PAINT 7, 7, ZT",2,2,10
101 TEXT 40,80,"PAINT X, Y, ZT",2,1,10
102 TEXT 40,110,"NICHT VERGESSEN:",2,1,9
103 TEXT 40,130,"(X,Y): IN DER VORGESEHENEN FLAECHEN",2,2,8
104 TEXT 40,180,"QUIZ: WOHL KAUM NOETIG!",1,1,10
105 WAIT 553,2: REM COPY
106 :
```

5.TEXT = SCHRIFT-STRING

TEXT 40,47,....., 2, 2, 8

TEXT X, Y, \$ ' , ZI, TI, AB

NICHT VERGESSEN :

(X, Y):LINKE OBERE ECKE DES \$

QUIZ : EIM ZITAT SETZEM!



```

109 :
109 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
110 REM INT/BIRKENBIHL, 1984
111 :
112 REM GOSUB <TITEL FUER DIES PROGRAMM>
113 :
114 COLOUR14,14:REM RA,HG HELLBLAU
115 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
116 REC 0,0,319,199,1:REM RAHMEN-RECHTECK
117 :
118 :
119 REM ++++++
120 REM ++ DER T E X T - BEFEHL ++
121 REM ++++++
122 :
123 REM EINER DER WICHTIGSTEN BEFEHLE!
124 :
125 TEXT 30,25,"5.TEXT = SCHRIFT-$TRING",1,1,8
126 BLOCK 30,40,279,70,1
127 TEXT 40,47,"TEXT 40,47, ' . . . / , 2, 2, 8",2,2,8
128 TEXT 40,75,"TEXT X, Y, $ /ZT, TI, AB",2,1,8
129 TEXT 40,115,"NICHT VERGESSEN:",2,1,9
130 TEXT 40,130,"(X,Y):LINKE OBERE ECKE DES #",2,2,8
131 TEXT 40,180,"QUIT: EIN ZITAT SETZEN!",1,1,10
132 WAIT 653,2:REM COPY
133 :

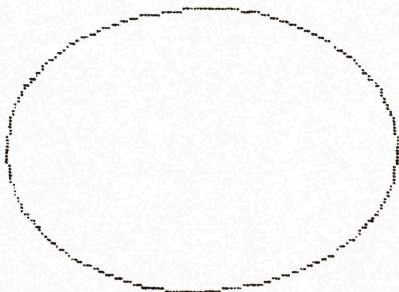
```



6. CIRCLE = KREIS

**WILHELM STRÖMVERBAND**

CIRCLE X, Y, X1, Y1, Z1



NICHT VERGESSEN :

**KREIS=SONDERFORM VON ELLIPSE**

QUIZ: MACHEN SIE EINEN KREIS DRAUS!

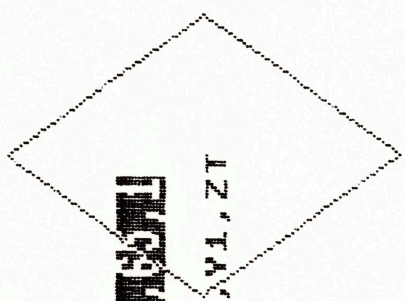


```
135 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
136 REM INT/BIKENBIHL. 1984
137 :
138 REM GOSUB <TITEL FUER DIES PROGRAMM>
139 :
140 COLOUR14,14:REM RA,HG HELLBLAU
141 HIRES0,14:REM ZF=0 <SCHWARZ>, HG=HELLBLAU
142 REC 0,0,319,199,1: REM RAHMEN-RECHTECK
143 :
144 :
145 REM ++++++
146 REM ++ DER C I R C L E - BEFEHL ++
147 REM ++++++
148 :
149 REM EIGENTLICH "ELLIPSE": KREIS = EINE SONDERFORM DER ELLIPSE!
150 :
151 TEXT 20,25,"6.CIRCLE = KREIS",1,1,8
152 CIRCLE 259,75,45,65,1
153 TEXT 20,47,"6CIRCLE 259,75,45,65,1",2,2,8
154 TEXT 20,76,"CIRCLE X, Y,X1,Y1,Z1",2,1,8
155 TEXT 20,135,"NICHT VERGESSEN:",2,1,9
156 TEXT 20,150,"KREIS=SONDERFORM VON ELLIPSE",2,2,8
157 TEXT 20,180,"QUIZ: MACHEN SIE EINEN KREIS DRAUS!",1,1,8
158 WAIT 653,2: REM COPY
159 :
```

7. ARC = BOGEN

~~ARC 250, 75, 0, 360, 90, 45, 315, 15~~

ARC X, Y, SM, EM, AB, XI, YI, ZT



NICHT VERGESSEN:

SW(0)+EW(360)+AB>> VERSCH. FIGUREN

QUIZ: MACHEM SIE EIN DREIECK DRAUS!





```
161 :
162 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
163 REM INT/BIKENBIHL, 1984
164 :
165 REM GOSUB <TITEL FUER DIES PROGRAMM>
166 :
167 COLOUR14,14:REM RA,AG HELLBLAU
168 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
169 REC 0,0,319,199,1: REM RAMMEN-RECHTECK
170 :
171 :
172 REM ++++++
173 REM ++ DER A R C - BEFEHL ++
174 REM ++++++
175 :
176 REM ACHTUNG: AB >> VERSCHIEDENEN GEOMETRISCHEN FIGUREN !
177 :
178 TEXT 20,25,"7.ARC = BOGEN",1,1,10
179 ARC 250,75,0,360,90,45,65,1
180 TEXT 20,47,"3.ARC 250,75, 0,360,90,45,65,1",2,2,8
181 TEXT 20,76,"ARC X, Y,SM, EM,AB,X1,Y1,ZT",2,1,8
182 TEXT 20,135,"NICHT VERGESSEN:",2,1,9
183 TEXT 20,150,"SM(0)+EM(360)+AB>> VERSCH. FIGUREN",2,2,8
184 TEXT 20,180,"QUIZ: MACHEN SIE EIN DREIECK DRAUS!",1,1,8
185 WAIT 653,2: REM COPY
186 :
```



8. ANGL = WINKEL

ANGL 2Y07Y5, 45, 45, 45, 45

ANGL X, Y, WINKEL, X1, Y1, ZT



NICHT VERGESSEN:

ANGL + FOR...NEXT >>TOLLE GEBILDE!

QUIZ: EIM FOR...NEXT-ANGL PROGRAMM!



```
188 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
189 REM INT/BIRKENSTIHL, 1984
190
191 REM GOSUB <TITEL FUER DIES PROGRAMM>
192
193 COLOUR14,14:REM RA,HG HELLBLAU
194 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
195 REC 0,0,319,199,1: REM RAMMEN-RECHTECK
196
197
198 REM ++++++
199 REM ++ DER A N G L - BEFEHL ++
200 REM ++++++
201
202 REM ACHTUNG: ANGL = RADIUS EINER G E D A C H T E N FIGUR!
203
204 TEXT 20,25,"8.ANGL = WINKEL",1,1,10
205 LINE 270,40,270,90,1:ANGL 270,75,45,45,45,1
206 TEXT 20,47,"8ANGL 270,75,45,45,45,1",2,2,8
207 TEXT 20,75,"ANGL X, Y,WINKEL,X1,Y1,ZT",2,1,8
208 TEXT 20,135,"NICHT VERGESSEN:",2,1,9
209 TEXT 20,150,"ANGL + FOR...NEXT >>TOLLE GEBILDE!",2,2,8
210 TEXT 20,180,"QUIZ: EIN FOR...NEXT-ANGL PROGRAMM!",1,1,8
211 WAIT 553,2: REM COPY
212
```

9. PLOT = ZEICHNE E. PUNKT

PLOT 270,40, 1

PLOT X, Y,ZT

NICHT VERGESSEN:  
NUR MIT FOR...NEXT SINNVOLL!

QUIZ: EINE GEPUNKTETE LINIE!





```
214 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
215 REM INT/BIKENBIHL, 1984
216 :
217 REM GOSUB <TITEL FUER DIES PROGRAMM>
218 :
219 COLOUR14,14:REM RA,HG HELLBLAU
220 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
221 REC 0,0,319,199,1: REM RAMMEN-RECHTECK
222 :
223 :
224 REM ++++++
225 REM ++ DER P L O T - BEFEHL ++
226 REM ++++++
227 :
228 REM ACHTUNG: PLOT ZEICHNET JEWEILS NUR EINEN EINZIGEN PUNKT!
229 :
230 TEXT 20,25,"9.PLOT = ZEICHNE E. PUNKT",1,1,8
231 PLOT 270,40,1
232 TEXT 20,47,"PLOT 270,40, 1",2,2,8
233 TEXT 20,76,"PLOT X, Y,Z",2,1,8
234 TEXT 20,139,"NICHT VERGESSEN:",2,1,9
235 TEXT 20,143,"NUR MIT FOR,...NEXT SINNVOLL!",2,2,9
236 TEXT 20,180,"QUIZ: EINE GEPUNKTETE LINIE!",1,1,10
237 WAIT 653,2: REM COPY
238 :
```



10. TEST = IST E. PUNKT GESETZT

A = TEST(270, 40) ?

VAR. = TEXT( X, Y )

NICHT VERGESSEN:  
INSBES. MIT JOY INTERESSANT!

QUIZ: PUNKT VERSETZEN+TESTEM

```
240 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3
241 REM INT/BIKENBIHL, 1984
242 :
243 REM GOSUB <TITEL FUER DIES PROGRAMM>
244 :
245 COLOUR14,14:REM RA,HG HELLBLAU
246 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
247 REC 0.0,319,199,1: REM RAHMEN-RECHTECK
248 :
249 :
250 REM ++++++
251 REM ++ DER T E S T - BEFEHL ++
252 REM ++++++
253 :
254 REM RICHTUNG: TEST SINNVOLL MIT J O Y UND "ACTION"
255 TEXT 267,45,"?",1,4,8
256 TEXT 20,25,"10.TEST = IST E. PUNKT GESETZT",1,1,8
257 PLOT 270,40,1:A=TEST (270,40)
258 TEXT 45,47," A=TEST(270,40)",2,2,11
259 TEXT 45,76,"VAR.=TEXT( X, Y)",2,1,11
260 TEXT 20,130,"NICHT VERGESSEN:",2,1,9
261 TEXT 20,143,"INSBES. MIT JOY INTERESSANT!",2,2,10
262 TEXT 20,180,"QUIZ: PUNKT VERSETZEN+TESTEN",1,1,10
263 WAIT 653,2: REM COPY
264 :
```

11. CHAR = EINEN CHARAKTER SETZEN

CHAR 270,40, 220, 1, 5

CHAR X, Y, POKE,ZT, TI >>



NICHT VERGESSEN:

MEHRERE CHAR'S >> MUSTER!

QUIZ: 5 CHAR'S ALS MUSTER






```
266 :  
267 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V. KAP. 3  
268 REM IMT/BIKENBIHL, 1984  
269 :  
270 REM GOSUB (TITEL FUER DIES PROGRAMM)  
271 :  
272 COLOUR14,14:REM RA,HG HELLBLAU  
273 HIRES0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU  
274 REC 0,0,319,199,1: REM RAHMEN-RECHTECK  
275 :  
276 :  
277 REM ++++++  
278 REM ++ DER C H A R - BEFEHL ++  
279 REM ++++++  
280 :  
281 REM ACHTUNG: CHAR SETZT NUR EIN EINZELNES ZEICHEN  
282 :  
283 TEXT 30,25,"11,CHAR = EINEN CHARAKTER SETZEN",1,1,8  
284 CHAR 270,40,220,1,5  
285 TEXT 30,50,"CHAR 270,40, 220, 1, 5",2,2,8  
286 TEXT 30,69,"CHAR X, Y, POKE,ZT, TI >>",1,1,8  
287 TEXT 30,130,"NICHT VERGESSEN:",2,1,9  
288 TEXT 30,143,"MEHRERE CHAR'S >> MUSTER!",2,2,10  
289 TEXT 30,180,"OUT2:5 CHAR'S ALS MUSTER",1,1,11  
290 WAIT 653,2: REM COPY  
291 :
```



12. ROT = DREHEM EINER (DRAW)-FIGUR  
13. DRAW = ZEICHNEN EINER FIGUR

R 0 T 0, 5   
DREHUNG, GROESSE UND:

DRAW '556360636568877775', 270, 65, 1  
DRAW 'STRING ... ..', X, Y, ZT

NICHT VERGESSEN:

OHNE ROT-BEFEHL GEHT'S NICHT!

QUIZ: 5 VERVOLLKOMMEN SIE DAS 'E'



```

293 :
294 REM DEMO-PROGRAMM FUER DIE EINZELNEN HIRES/MULTI-BEFEHLE V.KAP.3
295 REM INT/BIKENBIHL, 1984
296 :
297 REM GOSUB <TITEL FUER DIES PROGRAMM>
298 :
299 COLOUR14,14:REM RA,HG HELLBLAU
300 HIRES0,14:REM ZF=0 <SCHWARZ>, HG=HELLBLAU
301 REC 0,0,319,199,1: REM RAHMEN-RECHTECK
302 :
303 :
304 REM ++++++
305 REM ++ R O T + D R A W - BEFEHLE +++
306 REM ++++++
307 :
308 REM ACHTUNG: DRAW OHNE ROT IST N I C H T M O E G L I C H !
309 :
310 TEXT 30,12,"12.ROT = DREHEN EINER <DRAW>-FIGUR",1,1,8
311 TEXT 30,22,"13.DRAW = ZEICHNEN EINER FIGUR",1,1,9
312 ROT0,5:REM ROTATION KEINE, GROESSE 5
313 DRAW "556360636568877775",270,65,1
314 :
315 PAUSE10
316 TEXT 20,09,"ROT 0,5"/2,2,15
317 TEXT 20,56,"DREHUNG,GROESSE UND:",1,1,9
318 TEXT 20,80,"DRAW /556360636568877775/,270,65,1",1,2,8
319 TEXT 20,99,"DRAW /#TRING ... .. /, X, Y,ZT",1,1,8
320 TEXT 20,135,"NICHT VERGESSEN:",2,1,9
321 TEXT 20,152,"OHNE ROT-BEFEHL GEHT'S NICHT!",2,2,8
322 TEXT 20,180,"QUIZ:5 VERVOLLKOMMEN SIE DAS /E/",1,1,8
323 WAIT 650,2: REM COPY

```

DIESE TRICKFILM-ARTIGE DEMO  
 SOLLTE MEHRMALS GEGEHEN UND  
 ANALYSIERT WERDEN. HIER WERDEN  
 DIE WICHTIGSTEN BEFEHLE GEHÖRIG  
 GERECHT DEMONSTRIERT. VIEL SPASS!

```

1 REM SIBAS DEMO: IWT/BIRKENBHL, 1984
2 COLOUR14, 14: HIRE0, 14: REC0, 0, 319, 199, 1
3 TEXT 22, 40, "MIT RETURN KOENNEN SIE DIE", 1, 1, 10
4 TEXT 22, 60, "PAUSEN VERKUEZTEN. NUR NICHT", 1, 1, 10
5 TEXT 22, 80, "INNERHALB VON FOR...NEXT LOOPS.", 1, 1, 9
6 TEXT 75, 140, "VIEL SPASS!", 1, 1, 15: PAUSE27
7 COLOUR0, 0: HIRE0, 0: MULTI 3, 14, 7: PAUSE1
8 ARC79, 30, 0, 360, 90, 23, 26, 3: PRINT79, 30, 3
9 TEXT66, 16, "IWT", 0, 2, 8: TEXT75, 35, "+", 0, 2, 8
10 TEXT13, 64, "VERA F. BIRKENBHL", 2, 1, 7: PAUSE2
11 TEXT22, 83, "PRAESENTIEREN:", 3, 2, 8: PAUSE2
12 FORX=1 TO 3 STEP .3: REM START LOOP
13 F=F+1: LOW COL F, F+2, F+4: IFF=5 THEN F=0
14 TEXT 25, 105, "GRAFIK", 1, 5, 19
15 TEXT 13, 151, "MIT SIMON'S BASIC", 1, 4, 9
16 NEXT X: PAUSE3: REM END LOOP
17 HIRE0, 0: MULTI3, 4, 6: BLOCK1, 140, 199, 190, 3: PAUSE1
18 BLOCK1, 1, 60, 127, 3: PAUSE2
19 TEXT 4, 9, "GRAPHIK", 1, 4, 22: PAUSE1
20 TEXT 4, 75, "&SCHRIFT", 2, 3, 19: PAUSE1
21 TEXT7, 155, "MIT SIMON'S BASIC", 1, 3, 9: PAUSE4
22 HIRE0, 0: MULTI 4, 14, 6
23 TEXT6, 5, "BLOCK X,Y,X1,Y1,ZT", 2, 2, 8: PAUSE1: BLOCK4, 25, 196, 40, 1
24 TEXT4, 170, "ROEBIGER BLOCK-BEFEHL:", 1, 2, 8: PAUSE1
25 TEXT13, 190, "BLOCK 4, 25, 196, 40, 1", 1, 1, 7: PAUSE1
26 TEXT9, 47, "X: START X-ACHSE", 3, 2, 9
27 PAUSE1
28 TEXT9, 67, "Y: START Y-ACHSE", 3, 2, 9: PAUSE1
29 TEXT9, 87, "X1: STOP X-ACHSE", 3, 2, 9: PAUSE1
30 TEXT9, 107, "Y1: STOP Y-ACHSE", 3, 2, 9: PAUSE1
31 TEXT10, 127, "ZT=ZEICHENTYP", 3, 2, 11: PAUSE2
32 HIRE0, 0: MULTI 14, 4, 5
33 TEXT4, 1, "MEHR BLOCKS:", 1, 3, 13: PAUSE2
34 BLOCK124, 155, 196, 180, 3: PAUSE1
35 TEXT 4, 165, "TEXT MOEGLICH", 3, 1, 8: PAUSE1

```



36 BLOCK135,110,180,150,1:PAUSE1  
37 TEXT4,127,"#GROSSBUCHSTABEN",1,1,8:PAUSE1  
38 BLOCK145,50,196,105,2:PAUSE1  
39 TEXT4,50,"#KLEINBUCHSTABEN",2,1,8:PAUSE1  
40 TEXT4,65,"#ZAHLEN 01,2,3...",2,1,8:PAUSE1  
41 TEXT4,85,"#SYMBOLER #\*#\*#\*",2,1,8:PAUSE1  
42 TEXT5,190,"#GAR NICHT SO SCHWER",1,1,8:PAUSE5  
43 COLOUR0,0:HIRE10,0  
44 TEXT0,10,"#DIESER BILDSCHIRM IST IM HIRE",1,2,10:PAUSE1  
45 TEXT0,30,"#MODUS. AUFLEESUNG 360(X)\*200(Y)",1,2,10:PAUSE1  
46 TEXT0,50,"#PUNKTE PRO MONITOR:NUR ZWEI",1,2,12:PAUSE1  
47 TEXT0,70,"#FARBEN MOEGLICH:ZEICHENFARBE",1,2,11:PAUSE1  
48 TEXT0,90,"#\* HINTERGRUND. HIER IST DIES:",1,2,11:PAUSE2  
49 TEXT0,120,"#ZEICHENFARBE 10=HELLROT",1,1,14:PAUSE1  
50 TEXT0,135,"#HINTERGRUNDF. 0=SCHWARZ",1,1,14:PAUSE2  
51 TEXT25,170,"#EXAKT:HIRE,10,0",1,2,17:PAUSE2  
52 COLOUR0,0:HIRE14,0  
53 BLOCK56,99,250,160,1:PAUSE2  
54 TEXT9,15,"#DIESER BILDSCHIRM",1,2,18:PAUSE1  
55 TEXT9,33,"#IST EBENFALLS:HIRE",1,2,16:PAUSE1  
56 TEXT60,175,"#HIRE 14,0",1,3,20:PAUSE1  
57 TEXT43,70,"#ZF 14 = #HELLBLAU",1,2,14:PAUSE9  
58 HIRE80,0:PAUSE1:MULTI6,10,3  
59 BLOCK1,1,60,199,3:PAUSE2  
60 BLOCK109,160,199,199,2:PAUSE1  
61 BLOCK65,109,105,199,1:PAUSE1  
62 TEXT11,10,"#MULTI 6,10,3",4,3,12  
63 TEXT9,168,"+++++++++++++++++++++",4,2,10:PAUSE8  
64 COLOUR0,0:HIRE4,0  
65 TEXT10,10,"#DER NAECHSTE BEFEHL HEISST",1,2,12  
66 TEXT10,29,"#LINE X,Y,X1,Y1,ZF",1,2,18:PAUSE3  
67 TEXT12,55,"#ZB:LINE10,80,309,80,1",1,1,14:PAUSE2  
68 LINE10,80,309,80,1:PAUSE3  
69 TEXT10,120,"#NATUERLICH GEHT MIT 'LINE',1,2,12  
70 TEXT10,140,"#MEHR MIT #FOR...NEXT!",1,2,16:PAUSE2



```

71 TEXT10,160,"#SIEHE NAECHSTES BEISPIEL",1,2,12
72 TEXT10,180,"#MIT FOR...NEXT-LOOP",1,2,15:PAUSE5
73 HIRES4,0
74 FORX=0 TO 820 STEP 3 :REM START LOOP
75 M=110/820*X:LINE820-X,M,X,100+M,1
76 NEXTX:PAUSE12:REM ENDE LOOP
77 CSET0:COLOUR0,0
78 PRINT "3000"
79 CENTRE"ZUER BEFEHL"
80 PRINT"000"
81 CENTRE"CSET 0"
82 PRINT"000"
83 CENTRE"BEWIRKT DIE"
84 PRINT"000"
85 CENTRE"RUECKKEHR ZU"
86 PRINT"000"
87 CENTRE"LOW RESOLUTION."
88 PRINT"000":PAUSE3
89 CENTRE"TROTZDEM: WEITER MIT HIRES!":PAUSE 9
90 COLOUR0,0:HIRES 14,0:LINE0,25,360,25,1
91 TEXT0,3,"#DIES IST HIRES 14,0",1,2,17:PAUSE2
92 TEXT13,178,"#DIES IST EIN TEXT-BEFEHL:",1,2,13:PAUSE2
93 TEXT45,90,"#ES FOLGT:BLOCK 30,60,278,148,1",1,1,7:PAUSE 1
94 BLOCK30,60,278,148,1:PAUSE 2
95 TEXT45,99,"#DIES IST:BLOCK 30,60,278,148,1",2,1,7:PAUSE 3
96 BLOCK30,90,278,108,1:REM LOESCHBLOCK
97 TEXT65,95,"#TEXT:INVERTIERT",2,2,12:PAUSE3
98 BLOCK30,90,278,108,1:REM LOESCHBLOCK:PAUSE 3
99 TEXT80,95,"#MEHR TEXT",2,2,15:PAUSE 3
100 BLOCK30,90,278,108,1:REM LOESCHBLOCK:PAUSE 3
101 TEXT 55,95,"#GRAFIK-SYMBOL:",2,2,12:PAUSE 3
102 BLOCK53,94,270,114,1:REM LOESCHBLOCK M.FARBE DES GROSSBLOCKS
103 TEXT55,95,"#####",2,2,12
104 TEXT55,95,"|",2,2,12:PAUSE2
105 TEXT55,95,"#####",1,2,12

```

106 TEXT55,95,"|",1,2,12  
107 TEXT55,95,"#####",2,2,12:PAUSE2  
108 TEXT55,95,"#####",1,2,12:PAUSE3  
109 TEXT50,95,"FALL DIES IST HIRES",2,2,12:PAUSE3  
110 TEXT50,95,"FALL DIES IST HIRES",1,2,12:PAUSE3  
111 TEXT50,95,"WEDERNAECHST : MULTI",2,2,12:PAUSE3  
112 HIRES0,0:MULTI 14,3,4  
113 TEXT0,0,"BEI MULTI IST",1,3,12: PAUSE3  
114 TEXT9,40,"JEDER PUNKT",2,3,13:PAUSE3  
115 TEXT0,80,"DOPPELT SO BREIT",3,3,10:PAUSE3  
116 TEXT3,120,"WIE BEI HIRES",1,3,12: PAUSE2  
117 TEXT9,175,"WAS HEISST:",3,2,13: PAUSE2  
118 HIRES0,0:MULTI14,3,4  
119 TEXT12,69,"M:150 PUNKTE",3,3,11:PAUSE3  
120 TEXT12,99,"M:BLEIBT 200",1,3,11:PAUSE4  
121 HIRES0,0:MULTI 14,3,4  
122 TEXT1,5,"MULTI - MODUS",1,4,12  
123 TEXT1,168,"83 ZEICHENFARBEN",2,1,10:PAUSE3  
124 TEXT20,80,"B L O C K",3,2,12  
125 TEXT20,115,"115,60,128,148,3",3,2,7:PAUSE2  
126 BLOCK15,60,128,148,3:PAUSE5  
127 TEXT25,95,"WIEDER HIRES:",4,2,7:PAUSE3  
128 HIRES 10,0  
129 TEXT125,1,"YES FOLGT CIRCLE",1,2,10:PAUSE3  
130 TEXT125,17,"CIRCLE 80,100,48,78,1",1,1,7:PAUSE3  
131 TEXT125,39,"MIN SIMON'S BASIC IST:",1,1,7  
132 TEXT125,50,"ELLIPSE = CIRCLE",1,1,9:PAUSE3  
133 TEXT125,39,"MIN SIMON'S BASIC IST:",0,1,7  
134 TEXT125,50,"ELLIPSE = CIRCLE",0,1,9  
135 CIRCLE80,100,48,78,1:PAUSE 3  
136 TEXT95,110,"YES FOLGT PAINT",1,2,12:PAUSE3  
137 TEXT95,110,"YES FOLGT PAINT",0,2,12  
138 PAINT 90,110,1: PAUSE 3  
139 TEXT122,170,"WAS SAGEN SIE JETZT?",1,2,8:PAUSE 3  
140 COLOUR0,0:HIRES14,0

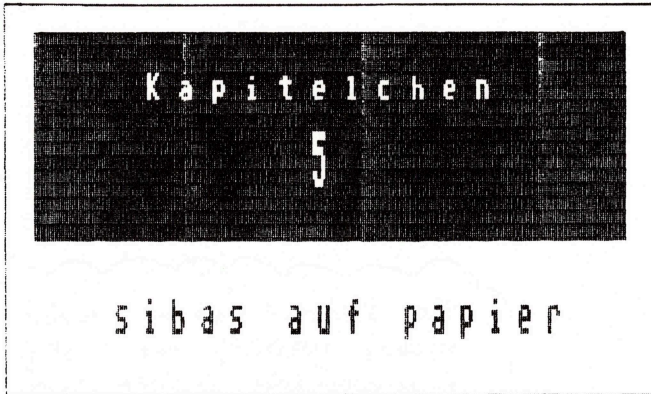
141 TEXT5,20,"#NOCHMAL IN MULTI",1,3,19:PAUSE2  
142 TEXT5,70,"#DIESELBEN ZAHLEN",1,3,19:PAUSE2  
143 TEXT9,120,"#FUER DEN KREIS!",1,3,19:PAUSE2  
144 HIRES0,0:MULTI10,2,7  
145 TEXT5,90,"#MULTI 10,2,7",1,4,13:PAUSE2  
146 TEXT5,90,"#MULTI 10,2,7",0,4,13  
147 TEXT25,1,"#ES FOLGT:CIRCLE",1,2,3:PAUSE3  
148 TEXT25,1,"#ES FOLGT:CIRCLE",0,2,9  
149 CIRCLE80,100,48,78,2:PAUSE2  
150 TEXT47,90,"#PAINT:",3,2,10:PAUSE3  
151 TEXT47,90,"#PAINT:",0,2,10  
152 PAINT 90,110,3:PAUSE2  
153 HIRES0,0:MULTI 7,14,15:REM NEUER MULTI BILDSCHIRM  
154 TEXT5,80,"#ES FOLGT:REC",3,2,12  
155 TEXT5,110,"#REC = #RECHTECK",2,2,10:PAUSE3  
156 TEXT5,80,"#ES FOLGT:REC",0,2,12  
157 TEXT5,110,"#REC = #RECHTECK",0,2,10  
158 REC1,1,60,60,1:PAUSE2:REC5,5,65,65,2:PAUSE2  
159 REC10,10,70,70,3:PAUSE3  
160 TEXT95,1,"#MEHR",2,3,11  
161 TEXT95,30,"#FARBEN",2,3,10:PAUSE3  
162 LOW COL2,13,4  
163 REC15,15,75,75,1:PAUSE3  
164 TEXT95,1,"#MEHR",0,3,11  
165 TEXT95,30,"#FARBEN",0,3,10  
166 REC20,20,80,80,2  
167 REC25,25,85,85,3  
168 REC30,30,90,90,1  
169 REC35,35,95,95,2  
170 REC40,40,100,100,3:PAUSE3  
171 TEXT5,170,"#PAINT50,50,2",2,3,12  
172 PAINT50,50,2:PAUSE2  
173 TEXT5,170,"#PAINT50,50,2",0,3,12  
174 TEXT28,170,"#ACHTUNG:",2,3,12  
175 FILL 0,0,40,20,160,7:PAUSE2



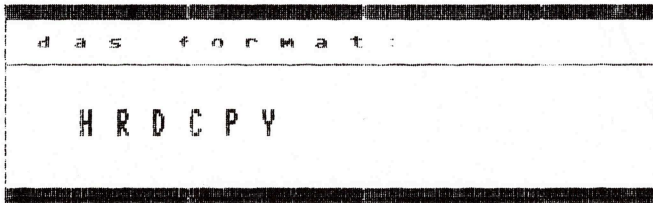


```
211 :: FOR X = 0 TO 7
212 :: ROT X,1
213 :: DRAW M$,76,99,3:REM DRAW MIT ZEICHENFARBE 3
214 :: NEXT X
215 :: FOR X = 0 TO 7
216 :: ROT X,1
217 :: DRAW M$,76,99,1:REM DRAW MIT ZEICHENFARBE 1
218 :: NEXT X
219 :: FOR X = 0 TO 7
220 :: ROT X,1
221 :: DRAW M$,76,99,2:REM DRAW MIT ZEICHENFARBE 2
222 :: NEXT X
223 :: NEXT I:PAUSE1:REM AUSSEN LOOP : BIS HIERHER SOLL ALLES 4 MAL LAUFEN
224 :: TEXT 19,179,"MIT SIMON'S BASIC",3,1,8
225 :: PAUSE6: RETURN
```

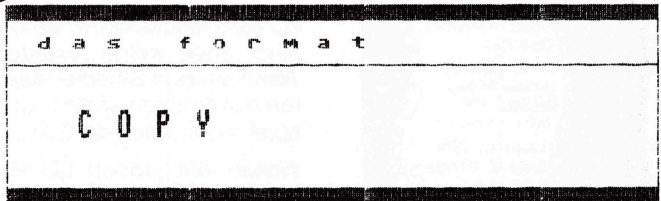
READY.



In der *sibas*-Anleitung wird der Eindruck vermittelt, daß das Ausdrucken von Bildschirmen gar kein Problem sei:

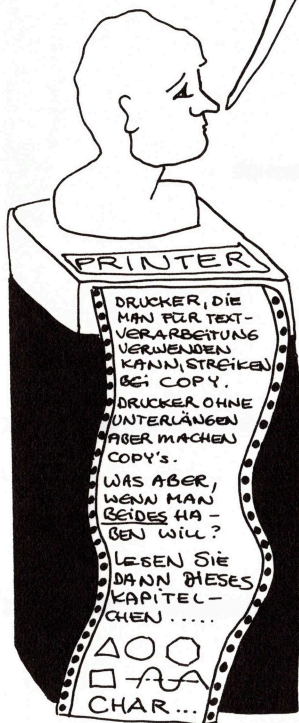


Bildschirmausdruck  
**LORES**  
(TEXT- od.  
LoRES Grafik)  
auf Papier



Bildschirmausdruck von  
HIRES und/oder MULTI

So leicht ist es aber nicht, **HRDCPY** oder **COPY** anzuwenden, insbes. auf Commodore-Druckern. Falls Sie also Probleme haben, lesen Sie hier weiter!



## 1. COMMODORE-DRUCKER

### Printer/Plotter 1520

Mit dem Printer/Plotter 1520 können Sie überhaupt keine *sibas*-Grafiken ausdrucken.

### 1525 bzw. MPS 801

Mit dem (alten) 1525 oder seinem neuen Nachfolger (MPS 801) ist die Situation wie folgt:

LO RES-Bildschirme werden zwar ausgedruckt, sind aber keine exakten Bildschirmkopien. Wenn sie zum Beispiel einen **TEST-RUN** mit Worten ausdrucken wollen, ist dies problemlos (Beispiel siehe nächste Seite, Abb. A).

Wollen Sie jedoch **LO RES**-Grafik, also Bildschirme mit Grafikzeichen per **PRINT** oder **FILL**-Befehl ausdrucken, dann wird Ihr Bild durch kleine Abstände »auseinandergerissen«. Statt so (Abb. B) sieht dann Ihr Ausdruck so aus (Abb. C).

WIE HEISST DU? TANJA  
 WIE ALT BIST DU? 12  
 GUTEN TAG, TANJA!  
 DANN BIST DU AN DEINEM  
 LETZTEN GEBURTSTAG GENAU  
 4380 TAGE ALT GEWESEN...

LORES HRDCPY von einem  
 TEST-RUN bei einem  
 WORT-PROGRAMM -  
 - problemlos

DRUCKER: CBM MPS 801

ABB. A

HRDCPY MIT COMMODORE:MPS 801:

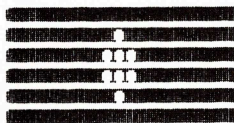


ABB. C

LORES HRDCPY von einer  
 GRAFIK wird von CBM-  
 DRUCKERN "zerrissen".

HIER CBM MPS-801, aber  
 der 1526 macht's genau so,  
 nur halb so klein!

So SOLTE es aus-  
 sehen: Auf EPSON  
 ging's.

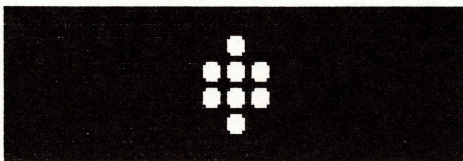
HRDCPY MIT EPSON: EINFACH:



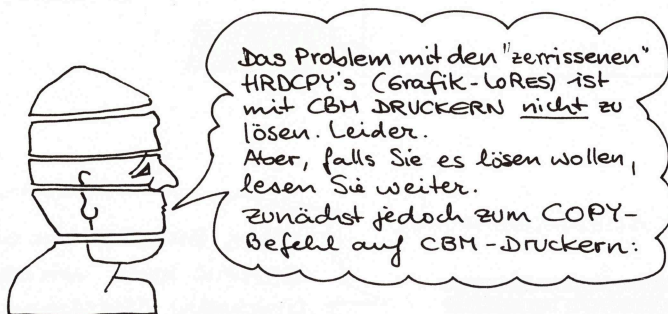
ABB. B

Das geht  
 auch, aber auf  
 EPSON:  
 doppelte Grösse  
 von TEXT oder  
 Grafik ...

HRDCPY MIT EPSON: DOPPELT:







HIRES-Bildschirme sind völlig problemlos (siehe zahlreiche Beispiele in diesem Buch!). MULTI sind eigentlich auch problemlos, d. h. sie werden anstandslos ausgedruckt. Aber, es gilt dabei folgendes zu beachten:

### **MULTI-Ausdruck:**

Bei MULTI sind die Farben in unterschiedlichen Farbspeichern untergebracht: Die Hintergrundfarbe und die Zeichenfarbe 3 kommen aus dem normalen Farbspeicher (in dem auch LO RES-Farben liegen). Zeichenfarben 1 und 2 hingegen kommen aus dem Video-Ram. Diese Farben werden auf Papier nicht »solide«, sondern senkrecht-schraffiert ausgedruckt, was insbesondere bei gewissen Grafiken zu sehr schönen Effekten führen kann (vgl. Ausdruck bei CHAR in Kapitelchen 3). Sollten Sie jedoch TEXT-Zeilen in diesen beiden Zeichenfarben haben, so können diese z. T. total unleserlich werden. Daher gehe ich wie folgt vor: Für ein Bildschirmprogramm wähle ich die Farben, die ich tatsächlich haben will. Vor einem Ausdruck auf Papier hingegen gehe ich im Editier-Modus noch mal in das Programm hinein und Sorge dafür, daß TEXT-Befehle immer in Zeichentyp 3 oder 0 stehen. 3 bei normalen TEXT-Befehlen, 0 bei invertierter Schrift, die in einen BLOCK oder eine gePAINTete Fläche hineingeschrieben wird. Wenn Sie sich die KARTE im ANHANG zum Zeichentyp einmal vornehmen, dann finden Sie in der Rubrik MULTI (rechts) bei Zeichentyp 4 (invertieren) auch angegeben, welche Zeichenfarben invertiert werden (nämlich 0 und 3, sowie 1 und 2). Mit Hilfe dieser Tabelle können Sie dafür sorgen, daß auch TEXTE in MULTI auf dem Papier lesbar werden!

---

## 1526

Falls Sie einen 1526 besitzen, dann können Sie LO RES-Ausdrucke machen, allerdings mit derselben Einschränkung wie unter dem 1520 (MPS 801) beschriebenen (kleine Zwischenräume!).

Kopien von HIRES und/oder MULTI können Sie mit *sibas* nicht machen! (Allerdings gibt es hier die Möglichkeit, das Problem mit einem Maschinenprogramm zu lösen, wiewohl der 1526 eigentlich NICHT grafikfähig ist! Dieses Programm hat Herr Arnd Wängler für dieses Buch geschrieben. Das Maschinen-Programm ist auf der Diskette zu diesem Buch enthalten.)

### Alte Commodore-Drucker

Sollten Sie eins der Uralt-Modelle (3022 oder 4022) herumstehen haben, dann ist Ihre Situation genauso wie beim 1526, nur mit dem Unterschied, daß die alten auch per Maschinenprogramm nicht grafikfähig werden! Zumindest haben einige Fachleute dies behauptet.

## 2. EPSON

Wie Sie sicher bereits festgestellt haben, besitzen die meisten C 64-Besitzer, die »ernsthaft« mit ihrem Gerät arbeiten, über kurz oder lang einen EPSON. Dies ist nicht verwunderlich, denn der bringt all das, was man sich bei einem Commodore-Printer gewünscht hätte. Sollten Sie also noch keinen Drucker besitzen, dann wäre zu überlegen, ob Sie ein paar Mark mehr ausgeben und gleich Nägel mit Köpfen machen. Denn mit dem EPSON können Sie sowohl Textverarbeitungs-Probleme (damit wird der 1526 voll ersetzt) als auch GRAFIK-Probleme lösen; und zwar macht der EPSON die oben beschriebenen »unnötigen« und unerwünschten Zwischenräume bei LO RES Grafik NICHT!!

Somit ist er besser als der MPS 801 (der wegen fehlender Unterlängen zur Textverarbeitung überhaupt nicht geeignet ist!).

Deshalb wollen wir kurz auf 2 EPSONs eingehen:

### EPSON FX 80

Er ist die beste Lösung, die ich bisher gesehen habe. Wiewohl ich bereits alle oben beschriebenen Commodore-Drucker besitze, habe ich ihn dazugekauft. Hätte ich eher gewußt, was ich jetzt weiß, hätte ich ihn GLEICH gekauft (und viel Geld gespart). Er kostet um die DM 1800 herum und kann mit Hilfe von kleinen Maschinesprachen-Routinen, folgendes:

1. Textverarbeitung (besser als der 1526)
2. GRAFIK mit allen Grafik-Programmen (also nicht nur *sibas*, aber auch *sibas*!)
3. Er kann doppelt so groß, kursiv, Breitschrift, unterstreichen und man kann einen völlig eigenen Zeichensatz definieren! Außerdem hat er einen eigenen RAM (für eigene Zeichensätze). Wenn Sie aber derzeit keinen eigenen Zeichensatz verwenden, wird dieser RAM als Puffer gebraucht, d. h.: Ihr Computer wird wieder READY während der EPSON noch druckt!!!

---

## **EPSON RX 80**

Dieser kleine Bruder zu ca. DM 1100 kann fast alles, wie oben beschrieben, nur mit folgenden Einschränkungen:

1. Er hat keinen PUFFER (RAM)
2. Er ist viel langsamer (statt 160 Zeichen nur 100 Zeichen).

So Sie also weder eigene Zeichensätze verwenden wollen noch unbedingt Zeit sparen müssen, könnte er Ihr Drucker werden...

Falls Sie Probleme bei der Anwendung von EPSON-Druckern mit dem C 64 haben sollten: Der 64-ACM (Anwender-Club-München) hat diese gelöst. Im Notfall schreiben Sie an:

64-ACM  
z. Hd. Herrn Arnd Wängler  
Rotkäppchenstraße 71  
8000 München 83

# Kapitelchen

## 6

### sibas: tips + tricks

Zwar liegt der Schwerpunkt dieses Buches bei den Grafik-Befehlen, aber ich möchte Ihnen doch einige Tips für einige der Programmierhilfen geben. (KEY, RENUMBER, FIND, AUTO, MERGE). Plus Tips für PAUSE, DIR und DISK, die sich aus der Praxis ergeben haben. Falls Sie mit diesen Befehlen nicht schon seit längerer Zeit viel arbeiten, wird dieses Kapitelchen Ihnen sicher etwas bieten können ...

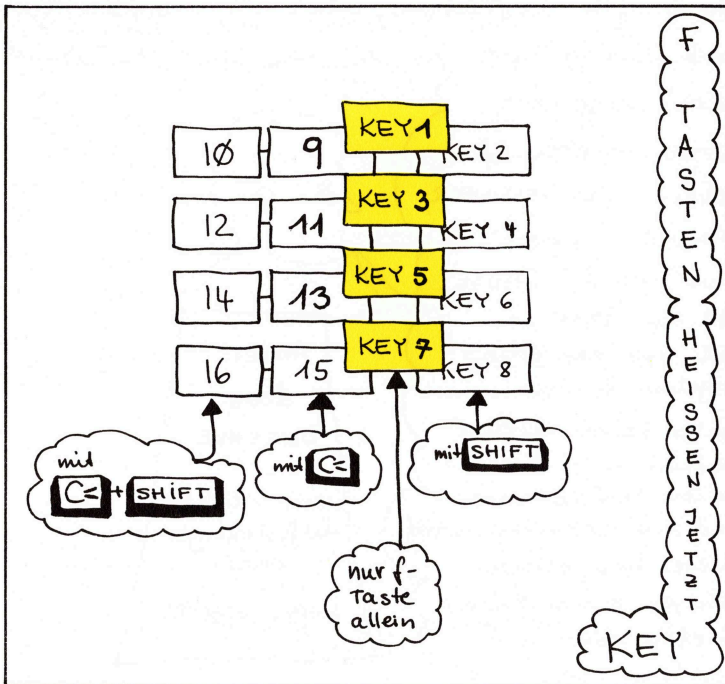
Andere  
sibas-  
BEFEHLE  
+  
nutzliche  
Hilfestellungen  
dazu  
folgen sofort.





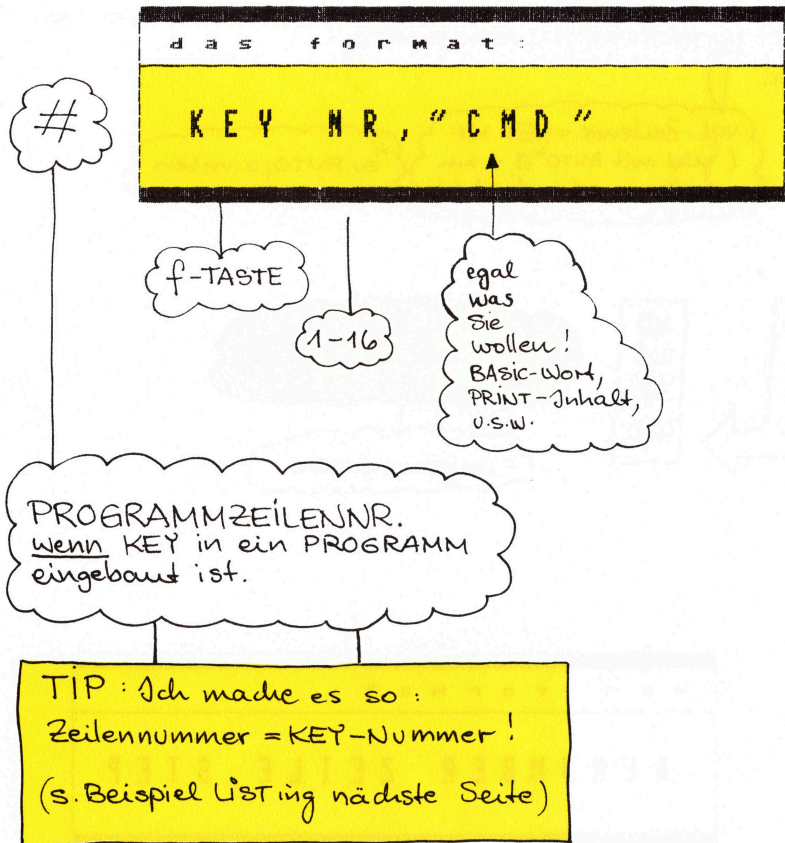
Wiewohl es in der *sibas*-Anleitung nicht ausdrücklich gesagt wird, können Sie die Funktionstasten per KEY-Befehl auch in einem kleinen Programm belegen. Dieses wird jedoch nicht im normalen BASIC-Speicher abgelegt (da es ja dann nach jedem NEW oder STOP+RESTORE gelöscht werden würde!). Das heißt, dieses BASIC-Programm verhält sich wie ein Maschinenprogramm!

Ich arbeite immer mit allen 16 Tasten:



Ich habe mehrere KEY-Programme geschrieben, mit unterschiedlicher Tastenbelegung: Will ich zum Beispiel normale BASIC-Programme schreiben, dann bewirken die F-Tasten normale BASIC-Worte, zum Beispiel: PRINT, INPUT, POKE53280,0 und POKE53281,0 (für schwarzen Rahmen und Bildschirm) und ähnliches. Will ich hingegen *sibas*-Programme erstellen, dann sind eben *sibas*-Worte (TEXT, BLOCK) etc. einprogrammiert. Wenn Sie nun doch, nach dem Laden, eine Taste verändern wollen, so können Sie dies nicht durch nochmaliges Laden des (geänderten) Programms (ohne den Computer abzuschalten), aber im Direkt-Modus geht dies sehr wohl:

KEY Nr, ».....«



## BEISPIEL für ein KEY-Programm

```

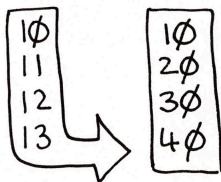
0 REM ***** ZWEITES KEY-PROG:KEY2
1 KEY 1,"CSET2:PAUSE50"+CHR$(13):REM LETZTE GRAFIK,PAUSE+RETURN IN EINEM
2 KEY 2,"CSET1"+CHR$(13):REM KLEINBUCHSTABEN
3 KEY 3,"LIST"+CHR$(13):REM LIST + RETURN
4 KEY 4,"WAIT653,2":REM = WARTEN AUF (C=) COMMODORE-TASTE
5 KEY 5,"COLOUR 0,0: REM RAHMEN + BILDSCHIRM SCHWARZ
6 KEY 6,"PAUSE": REM JEDE PAUSE = CA. 1 SEKUNDE LANG
7 KEY 7,"RUN"+CHR$(13):REM RUN + RETURN
8 KEY 8,"CSET0":REM = NORMALER BILDSCHIRM (NACH HIRES-MODUS)
9 KEY 9,"REC0,0,319,199,":REM ZEICHENTYP WIRD SEPARAT EINGESETZT
10 KEY 10,"RENUMBER":REM PARAMETER WERDEN SEPARAT EINGESETZT
11 KEY 11,"COLOUR14,14":REM HL.BLAU
12 KEY 12,"HIRES 14,0"
13 KEY 13,"OPEN4,4:CMD4":REM DRUCKER OEFFNEN; KEY 3 BEWIRKT DAS WEITERE...
14 KEY 14,"MULTI 14,3,4"
15 KEY 15,"PRINT#4:CLOSE4"+CHR$(13):REM SCHLIESSEN DRUCKER+RETURN
16 KEY 16,"PRINTCHR$(147):REM=CLR SCREEN
  
```

READY.

vgl.: Zeilennr.=KEY Nr.  
(geht mit AUTO\*0,1 am  
Anfang, sehr leicht)

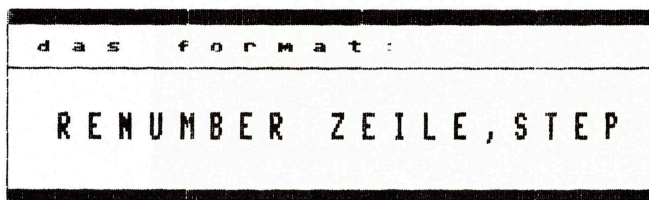
\* zu AUTO, s.unten

Achtung:  
RETURN  
gleich da-  
bei!



2. RENUMBER

=UMNUMERIEREN

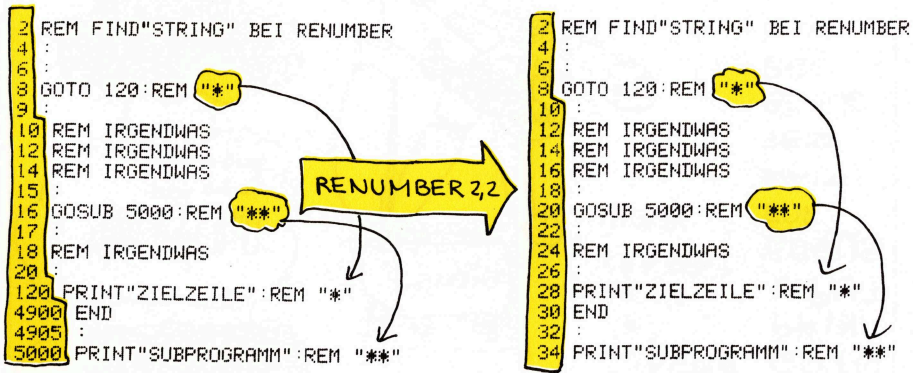


Eigentlich wäre RENUMBER ganz wunderbar. Aber, im Gegensatz zu anderen »Dienstleistungsprogrammen«, welche ein RENUMBER anbieten, ist das des *sibas* nicht perfekt. GOTOS und GOSUBs werden nämlich nicht neu berechnet. Nun kann man zwar die meisten Sprünge mit Hilfe der strukturierten Programmierung (CALL, EXEC, usw.) umgehen, aber: Wann immer ich ein Programm austeste, arbeite ich kurzzeitig mit GOTOS, die ich nach dem Testen wieder herausnehme. Dadurch kann ich bestimmte Aspekte schnell überprüfen. Sollten Sie also aus irgendeinem Grund doch einmal GOTOS oder GOSUBs in einem Programm haben, das Sie dann RENUMBERn wollen, dann können Sie den folgenden Trick anwenden: Geben Sie in jede Zeile VON DER gesprungen werden soll zusätzlich ein:

```
:REM >>*<<
```

Dasselbe machen Sie mit der Zeile, IN WELCHE gesprungen werden soll.

Sollten Sie mehrere ABSPRUNGZEILEN haben, dann bekommt die nächste einen \$string mit 2 Sternchen, die folgende einen \$string mit 3 Sternchen usw. Da ja die jeweiligen ZIELZEILEN immer denselben \$string bekommen, wird das Umnummerieren jetzt leicht, nämlich durch den FIND-Befehl (siehe unten).

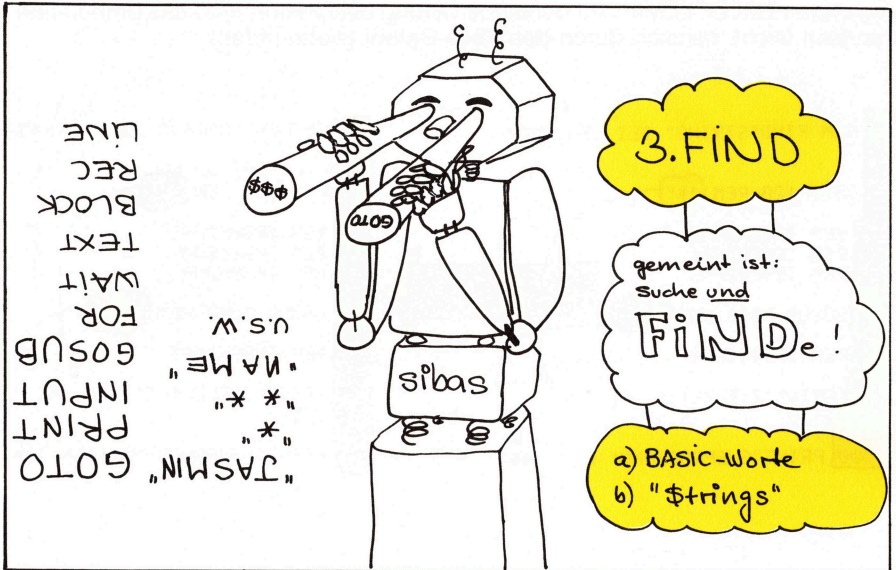




Noch ein Tip: Bei einem englischen Autor hatte ich gelesen, daß ein RENUMBER mit Schrittweite 0 zu einem Verlust des Programmes führen würde. Im Data Becker-Buch aber steht die Wahrheit: Solch ein Programm ist lauffähig, kann aber nicht mehr editiert werden. Aber es beginnt **jede** Programmzeile mit Null. Die Data Becker-Autoren weisen auch darauf hin, daß dies einen guten Programmschutz abgibt, wenn man sicherstellen will, daß nichts mehr verändert werden kann. Das gleiche gilt für einen RENUMBER-Befehl in »verbotene« Bereiche, zum Beispiel:

RENUMBER 63800,100

Hier wird die oberste Grenze (Zeilennummer 63999) überschritten. Jeder normale Versuch, solch eine Programmzeile einzugeben, würde zu einem SYNTAX-ERROR führen. Mit RENUMBER ist es möglich, aber auch solche Zeilen können nicht mehr editiert oder gelöscht werden!



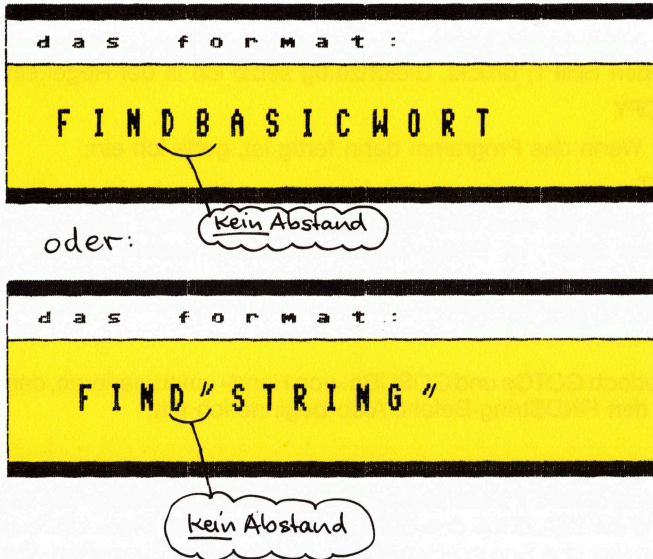
Achtung: Wenn Sie sich an das FORMAT in der sibas - Anleitung (S.13) halten, dann FINDen Sie genau gar nichts! Bitte weiterlesen, nächste Seite, hier!

---

Dieser Befehl kann zwei verschiedene Dinge FINDen: Erstens BASIC-Befehls-wörter und zweitens \$trings. Falls Sie aber bereits versucht haben damit zu arbei-ten, hatten Sie möglicherweise Probleme, da das FORMAT in der *sibas*-Anleitung (wie auch im Data Becker-Buch) falsch angegeben ist. Bei beiden steht:

FIND BASICwort, oder  
FIND \$STRING bzw. FIND Charakter-\$tring.

Beide betonen, daß Leerstellen als Zeichen gezählt werden. Ein FIND-Befehl mit einem Leerzeichen vor dem BASICwort bzw. \$STRING wird aber genau NICHTS FINDen! Also muß es heißen:



z.B.: FINDWAIT, FINDPRINT, FINDINPUT ...

z.B.: FIND "\*", FIND "\*\*", FIND "JA" ...

---

Desweiteren kann ein \$string nur gefunden werden, wenn Sie ihn exakt und vollständig angeben. Sollten Sie also mehrere PRINT-Befehle, wie zum Beispiel

»Bitte suchen Sie die richtige Taste aus«

haben, dann können Sie diesen keinesfalls durch **FIND»Taste«** FINDen! Also müssen Sie eingeben:

FIND »Bitte suchen Sie die richtige Taste aus«. Da dies langweilig ist, nutze ich die FIND-Funktion hauptsächlich für 2 Ziele:

1. Alle *sibas*-Programme, von denen ich später Bildschirmausdrucke machen will, bekommen an die Stelle zunächst ein

WAIT 653,2

Das ist normales CBM-BASIC und bedeutet: Warten, bis jemand die CBM-Taste (links neben SHIFT) drückt. Gleichzeitig setze ich in der Regel ein

:REM COPY

dahinter. Wenn das Programm dann fertig ist, gebe ich ein:

FINDWAIT

und alle Zeilen mit dem WAIT-Befehl werden angezeigt. Jetzt brauche ich nur eine dieser Zeilen im Editier-Modus verändern (indem ich bis COPY alles lösche) und dann die anderen Zeilennummern, die ja noch am Bildschirm stehen, in die editierte Zeile zu setzen. In Null Komma nix ist jeder WAIT-Befehl in COPY geändert worden.

2. Will ich jedoch GOTOs und GOSUBs »von Hand« umnummerieren, dann brauche ich jetzt den FIND\$string-Befehl. Also beginne ich mit:

FIND»\*«

Jetzt werden exakt 2 Zeilennummern ausgeworfen, nämlich die **ABSPRUNG-ZEILE** und die **ZIELZEILE** des GOTOs oder GOSUBs. Diese LISTe ich jetzt auf und verändere den Sprung-Befehl. Das REM mit dem Sternchen-\$string LASSE ich aber bestehen, falls ich später wieder einmal RENUMBERn will. . . Sollten es mehrere sein, dann wird die zweite Kombination durch

FIND»\*\*«

gesucht und gefunden usw.

## BILDSCHIRMAUSDRUCK VON FIND"\*":

```
FIND"*"  
8          28  
READY.
```

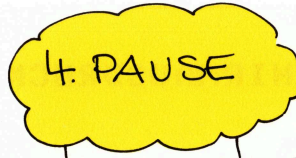
```
FIND"***"  
20         34  
READY.
```

ERINNERN SIE SICH AN DAS RENUMBER'te  
Beispiel vor 4 Seiten? Danach kann man  
die GOTO/GOSUB zeilen mittels der REM"\*"  
leicht finden und "zu Fuss" umnumerieren!

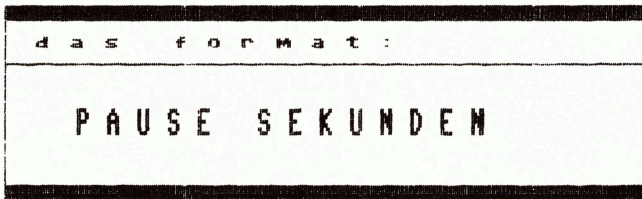




PAUSE 10  
= ca 10 Sekunden Pause  
PAUSE 50  
= ca 50 Sekunden Pause



Ersetzt die LEER-Schleife  
FOR X = 1 TO 1000 : NEXT



Natürlich ist dieser Befehl kinderleicht und braucht keine Erklärung. Jedes PAUSE ergibt eine Pause von ca. 1 Sekunde. Aber ich möchte Ihnen noch einen Programmiertrick verraten, den ich mit PAUSE anwende:

Wenn ich ein Programm teste, insbesondere ein Grafik-Programm, dann baue ich am Anfang öfter mal ein

PAUSE 10

ein; wobei jeder PAUSE-Befehl eine eigene Programmzeile erhält.

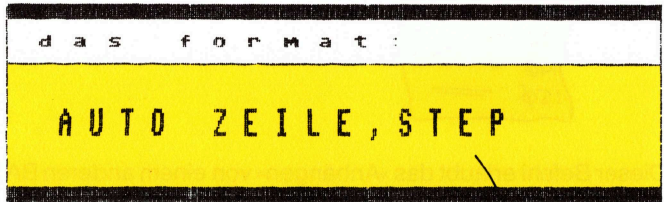
Zum Beispiel:

```
.  
. .  
. .  
40 PAUSE 10  
. .  
. .  
55 PAUSE 10  
. .  
. .  
78 PAUSE 10  
usw.
```

Da ja jeder PAUSE-Befehl beim RUN durch Drücken der RETURN-TASTE unterbrochen werden kann, kann ich genau sehen was passiert. Angenommen ich habe 6 LINE-Befehle. Wenn jede LINE einzeln am Bildschirm erscheint, dann sehe ich viel schneller, bei welchem LINE-Befehl mir ein Fehler unterlaufen ist. Oder wenn ich eine Visitenkarte entwerfe, dann sehe ich, wie diese ohne zusätzliche LINES wirkt, ehe diese LINES tatsächlich erscheinen. Wenn ich mich dann endgültig entschieden habe, finde ich alle diese unnötigen PAUSE-Befehle sofort durch FINDPAUSE.

Nun kann ich diese Zeilen schnell löschen und das Programm ist fertig.

5.  
AUTO



z.B. AUTO 1Ø,

1Ø

Schritt-  
weite

Die automatische Zeilenummerierung ist sehr angenehm. Nur einen Gedanken hierzu: Bedenken Sie, ehe Sie AUTO eingeben, ob Sie nicht einen Großteil der zu folgenden Zeilen besser im **Editier-Modus** eingeben könnten. Ich habe am Anfang festgestellt, daß AUTO mich manchmal dazu verführt hat, ähnliche Zeilen jeweils neu zu schreiben. Da ich sehr schnell tippe ist es egal, wer aber langsam eingibt, sollte vielleicht so viel wie möglich im **Editier-Modus** machen...

PS: Im Editier-Modus (wie auch bei einem LOAD aus dem »\$«-LISTING heraus) wechsele ich öfter die CURSOR-Farbe. Dann sehe ich viel besser was gerade geschieht.



## 6. MERGE = zusammenfügen

Dieser Befehl erlaubt das »Anhängen« von einem anderen BASIC-Programm (dessen Zeilen-Nummern natürlich höher sein müssen). Beispiel: Ihr erstes Programm endet mit Zeilennummer 100, und das zu MERGEnde Programm beginnt mit Zeile 500. Also, kinderleicht und sehr bequem. Hier mein Tip: Wenn ich mehrere Programmteile schreibe, die ich später per MERGE zusammensetzen möchte, dann Sorge ich dafür, daß die Zeilennummern »passen«: Der erste Teil wird nach Fertigstellung durch RENUMBER mit 10,1 für den START bestimmt. Er wird mit »Name, Start« geSAVED. Der zweite Teil wird nach Fertigstellung mit 500,1 unnumeriert und mit »Name-Mer 500« geSAVED. Der dritte Teil heißt dann »NAME, MER-600« usw. Damit ist es später ein leichtes, die Einzelprogramme in der richtigen Reihenfolge hintereinander zu MERGEN.

```
das format :  
MERGE "NAME" , 8
```

funktioniert genau wie LOAD, nur dass das zu MERGEnde Programm zusätzlich zum in Speicher befindlichen geladen wird!

s.Tip mit Programm-Namen zu MERGEnde Programme in Text, oben

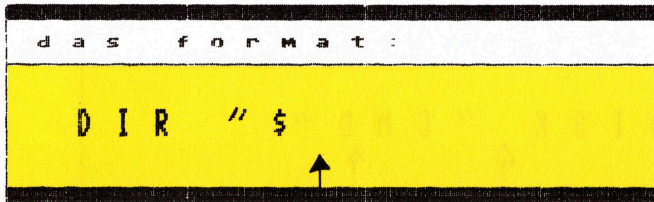


Z. DIR

DIR = DIRECTORY

Ganz wundervoll finde ich diesen Befehl:

DIR»\$



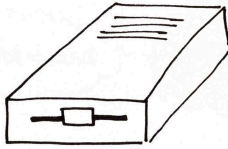
DIRECTORY  
(Abkürzung \$)  
= Inhaltsverzeichnis der Diskette

ACHTUNG: Kein Hochkomma  
am Ende dieses Formats !!

Er erlaubt jederzeit das Einlesen des Directory der Diskette, ohne daß das derzeit im Arbeitsspeicher befindliche Programm gelöscht wird. Denn: bei DIR wird das Directory nur »gelesen«, bei LOAD»\$«,8 wird es jedoch geladen (was ja jedes im Speicher befindliche BASIC-Programm löscht)! Mein Tip zu DIR: Angenommen, Sie haben ein Programm geladen und per RUN gestartet. Plötzlich fällt Ihnen noch ein Fehler auf, den Sie zuvor übersehen hatten. Also ändern Sie. Ehe Sie *sibas* mit DIR hatten, gab es nur zwei Möglichkeiten: Entweder Sie wußten haargenau wie das Programm im Directory hieß, dann konnten Sie es mit  
SAVE»C0:Name«,8

an dieselbe Stelle wieder abSAVEN. Oder aber, Sie mußten dem geänderten Programm einen neuen Namen zuweisen, hatten aber dann das fehlerhafte Programm zusätzlich auf der Diskette drauf. Jetzt geben Sie kurz ein DIR ein und Sie sehen genau, wie das Programm hieß. So kann es immer in seiner korrigierten Version abgespeichert werden! Das ist sehr schön, insbesondere bei vielen ähnlichen Programmen deren Namen auch ähnlich sind.





Dieser *sibas*-Befehl ist fantastisch, aber sein Format in der *sibas*-Anleitung ist falsch gegeben worden. Beim Format finden Sie nach DISK ein Komma (wie beim KEY-Befehl). Im Beispiel darunter fehlt das Komma. Diese Version ist richtig:

```
das format :  
  
DISK "CMD"
```

**ACHTUNG:**  
Die *sibas*-Anleitung zeigt im **FORMAT** (S. 23) ein Komma, im Beispiel jedoch keins.  
Der Befehl funktioniert jedoch nur ohne Komma. Er ist also nicht wie der **KEY**-Befehl aufgebaut!

alle Disketten-Befehle möglich  
**CoMmanDs**  
z.B. DISK "NØ:Name, id"  
bzw. DISK "N:Name, id" } \*  
(\* beide Versionen gehen)

---

## UEBERSICHT: ANHANG

- A: QUIZ-LOESUNGEN (KAP. 4)
- B: PANDA-LISTING (KAP. 3)
- C: POKE-WERTE IM BLOCK (TABELLE)
- D: LITERATUR- UND STICHWORTVERZ.
- E: **KARTEN** ZUM HERAUSTRENNEN!

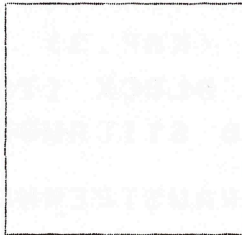
DIESE KARTEN SOLLTEN IMMER  
GRIFFBEREIT SEIN; SIE SIND  
GEHIRN-GERECHT GESTALTET:  
ZUM SCHNELLEN NACHSCHLAGEN!

## Anhang A: Lösungen (zu Kap. 4)

**NA, HABEN SIE ES GELOEST?**

**DIESE LOESUNG IST SELBSTVERSTAENDLICH  
NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR**

1.REC = RECHTECK



QUADRAT  $X1=Y1$

**NA, WAR DOCH LEICHT, ODER?!**

disk

```
4 REM IMT/BIRKENBIHL, 1984
5
6 REM L O E S U N G : R E C
7
8 COLOUR14,14:REM RA,HG HELLBLAU
9 HIRSES0,14:REM ZF=0 (SCHWARZ),HG=HELLBLAU
12 REC 0,0,319,199,1: REM RAHMEN-RECHTECK
16
18 REM ++++++
20 REM ++ DER R E C - BEFEHL ++
22 REM ++++++
24
26 REM ++++++ REC=RELATIV ++++++
28
29 TEXT 40,15,"1.REC = RECHTECK",1,1,8
30 REC 40,40,119,119,1
33 TEXT 177,87,"QUADRAT X1=Y1",1,1,8
38 TEXT 17,180,"NA, WAR DOCH LEICHT, ODER?",1,1,10
40 WAIT 650,2: REM COPY
```

# NA, HABEN SIE ES GELOEST?

DIESE LOESUNG IST SELBSTVERSTAEENDLICH  
NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR

2.BLOCK = AUSGEFUELLTES REC



QUADRAT  $X1=Y1$

KINDERLEICHT, NICHT WAHR?

Disk

```
101 REM IMT/BIRKENBHL, 1984
102 :
103 REM L O E S U N G : B L O C K
104 :
105 COLOUR14,14:REM RA,HG HELLBLU
106 HIRSE0,14:REM ZF=0 (SCHWARZ),HG=HELLBLAU
107 REC 0,0,319,199,1:REM RAHMEN-RECHTECK
109 :
110 REM ++++++
111 REM ++ DER B L O C K - BEFEHL ++
112 REM ++++++
113 :
114 REM ::::: BLOCK=RELATIV :::::
115 :
116 TEXT 40,15,"2.BLOCK = AUSGEFUELLTES REC",1,1,7
117 BLOCK 40,40,159,159,1
118 TEXT 127,87,"QUADRAT X1=Y1",2,1,8
119 TEXT 40,180,"KINDERLEICHT, NICHT WAHR?",1,1,10
120 MIT 653,2:REM COPY
```



## NA, HABEN SIE ES GELOEST?

DIESE LOESUNG IST SELBSTVERSTAENDLICH  
NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR

3.LINE = LINIE

---

---

---

---

---

LEICHT, WENN MAN'S KANN!



```
201 REM IWT/BIRKENBIHL, 1984
202 :
203 REM L O E S U N G : L I N I E
204 :
205 COLOUR14,14:REM RA,HG HELLBLAU
206 HIRE90,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
207 REC 0,0,319,199,1 : REM RAHMEN-RECHTECK
209 :
210 REM ++++++
211 REM ++ DER L I N I E - BEFEHL ++
212 REM ++++++
213 :
214 REM DENKEN SIE AN DEN EDITIER-MODUS FUER BEHNLICHE
215 :
216 TEXT 50,15,"3.LINE = LINIE",1,1,9
217 LINE 40,70,239,70,1
218 LINE 40,90,239,90,1
219 LINE 40,110,239,110,1
220 LINE 40,130,239,130,1
221 LINE 40,150,239,150,1
222 TEXT 40,180,"LEICHT, WENN MAN'S KANN!",1,1,9
223 WAIT 653,2 : REM COPY
```

**NA, HABEN SIE ES GELOEST?**

**DIESE LOESUNG IST SELBSTVERSTAENDLICH  
NUR EIN MOEGLICHER VORSCHLAG, ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR**

5.TEXT = SCHRIFT-STRING

**K A R L K R A U S :**

**MAN DARF NICHT NUR KEINE  
GEDANKEN HABEN,**

**MAN MUSS AUCH UNFAEHIG SEIN,  
SIE AUSZUDRUECKEN.**

**gilt auch fuer's programmieren!**

**ACHTUNG:**  
FÜR  
PAINT  
GAB'S  
KEIN  
QUIZ  
und  
daher  
keine  
Lösung

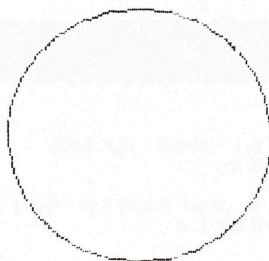
**disk**

```
401 REM IMT/BIRKENBILH, 1984
402 :
403 REM L O E S U N G : T E X T
404 :
405 COLOUR14,14:REM RA,HG HELLBLU
406 HIRE$0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
407 REC 0,0,319,199,1 : REM RAHMEN-RECHTECK
408 :
409 :
410 REM ++++++
411 REM ++ DER T E X T - BEFEHL ++
412 REM ++++++
413 :
416 TEXT 30,22,"S.TEXT = SCHRIFT-$STRING",1,1,8
417 BLOCK 30,40,279,20,1
418 TEXT 39,47,"KARL KRAUS:",2,2,22
419 TEXT 30,100,"MAN DARF NICHT NUR KEINE",1,1,9
420 TEXT 30,110,"GEDANKEN HABEN,",1,1,9
421 TEXT 30,130,"MAN MUSS AUCH UNFAEHIG SEIN,",1,1,9
422 TEXT 30,140,"SIE AUSZUDRUECKEN.",1,1,9
433 COPY
432 TEXT 30,180,"NICHT AUCH FUER'S PROGRAMMIEREN!",1,1,8
```

## NA, HABEN SIE ES GELOEST?

DIESE LOESUNG IST SELBSTVERSTAENDLICH  
NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR

6.CIRCLE = KREIS



EINFACHER ALS IN DER SIBAS-ANLEITUNG!

disk

```
501 REM INT/BIRKENBIHL, 1984
502 :
503 REM L O E S U N G : C I R C L E
504 :
505 COLOUR14,14:REM RA,HG HELLBLAU
506 HIRE$0,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
507 REC 0,0,019,199,1 : REM RAHMEN-RECHTECK
509 :
510 REM ++++++
511 REM ++ DER C I R C L E - BEFEHL ++
512 REM ++++++
513 :
514 REM
515 :
516 TEXT 88,13,"6.CIRCLE = KREIS",1,1,8
517 CIRCLE 159,99,65,65,1
522 TEXT 10,180,"EINFACHER ALS IN DER SIBAS-ANLEITUNG!",1,1,8
523 WAIT 658,2:REM COPY
```

**NA, HABEN SIE ES GELOEST?**

**DIESE LOESUNG IST SELBSTVERSTAENDLICH  
NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR**

**7.ARC = BOGEN**



**WIE WAER'S MIT WEITEREN FIGUREN?**



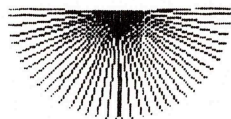
```
601 REM IMT/BIRKENBILH, 1984
602 :
603 REM L O E S U N G : A R C
604 :
605 COLOUR14,14:REM RA,HG HELLBLAU
606 HIRSO,14:REM ZF=0 (SCHWARZ),HG=HELLBLAU
607 REC 0,0,319,199,1:REM RAHMEN-RECHTECK
609 :
610 REM ++++++
611 REM ++ DER A R C - BEFEHL ++
612 REM ++++++
613 :
616 TEXT 20,25,"7.ARC = BOGEN",1,1,10
617 ARC 200,105,0,360,120,45,45,1
622 TEXT 20,180,"WIE WAER'S MIT WEITEREN FIGUREN?",1,1,9
623 WAIT 658,2:REM COPY
```



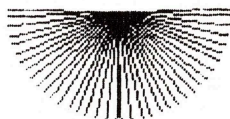
# NA, HABEN SIE ES GELOEST?

DIESE LOESUNG IST SELBSTVERSTAEENDLICH  
 NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
 LOESUNGEN SIND NATUERLICH DENKBAR

8 . ANGL = WINKEL



VORWAERTS!



RUECKWAERTS!

```

701 REM INT/BIRKENBIHL, 1984
702 :
703 REM L O E S U N G :  A N G L
704 :
705 COLOUR,14:REM RA,HG HELLBLAU
706 HIRE$0,14:REM ZF=0 (<SCHWARZ),HG=HELLBLAU
707 REC 0,0,319,199,1:REM RAHMEN-RECHTECK
709 :
710 REM ++++++
711 REM ++ DER  A N G L - BEFEHL ++
712 REM ++++++
713 :
715 TEXT 80,25,"8.ANGL = WINKEL",1,1,10
716 :
717 :MI=87
718 :FOR ZR=0 TO 37
719 :ANGL 77,85,MI,55,55,1
720 :MI=MI+5
721 :NEXT ZR
722 TEXT 30,180,"VORWAERTS!",1,1,10
723 :
724 :
725 :MI=272
726 :FOR ZR=0 TO 37
727 :ANGL 221,85,MI,55,55,1
728 :MI=MI-5
729 :NEXT ZR
730 :
742 TEXT 170,180,"RUECKWAERTS!",1,1,10
743 WHIT 653,2:REM COPY
    
```

D-5

**NA, HABEN SIE ES GELOEST?**

**DIESE LOESUNG IST SELBSTVERSTAENDLICH  
NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR**

**9.PLOT = ZEICHNE E. PUNKT**

.....

**IDEAL FUER FORMULARE MIT SIBAS!**



```
801 REM IMT/BIRKENBILH, 1984
802 :
803 REM L O E S U N G : P L O T
804 :
805 COLOUR14,14:REM RA,HG HELLBLU
806 HIR50,14:REM ZF=0 (SCHWARZ), HG=HELLBLAU
807 REC 0,0,319,199,1 : REM RAHMEN-RECHTECK
808 :
809 REM *****
810 REM *****
811 REM ++ DER P L O T - BEFEHL ++
812 REM *****
813 :
814 :
815 :
816 TEXT 60,25,"9.PLOT = ZEICHNE E. PUNKT",1,1,8
817 :
818 :
819 :
820 :
821 :
822 TEXT 20,180,"IDEAL FUER FORMULARE MIT SIBAS!",1,1,9
823 WAIT 650,2 : REM COPY
```

# LÖSUNGSVORSCHLAG: TEST

```

RUN

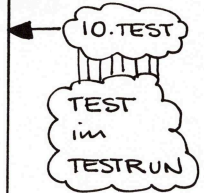
IST EIN PUNKT GESETZT?

DIES PROGRAMM TESTET DIES: VARIABLE A:

I=GESETZT;0=NICHT GESETZT.

IN DIESEM FALLE:A=TEST(250,40)

1
    
```



**ACHTUNG: TEST**  
 WIRD IMMER IN LORES VOM  
 COMPUTER "BEANTWORTET".  
 ES MUSS NOCH KEIN HIRES  
 GESETZT WORDEN SEIN: DER  
 PUNKT WIRD ALSO "UNSICHTBAR"  
 GEZEICHNET UND DURCH TEST BESTÄ-  
 TIGT.



```

851 REM IMT/BIRKENBILH, 1984
852 :
853 REM L O E S U N G : T E S T
855 COLOUR14,14:REM RA,HG HELLBLU
856 :
858 :
859 :
860 REM *****
861 REM ++ DER T E S T - BEFEHL ++
862 REM *****
863 :
866 PRINTCHR$(147)"XXXXXXXXXX IST EIN PUNKT GESETZT?"
867 PLOT 250,40,I:A=TEST (250,40)
868 PRINT"XXXXXXXXX DIES PROGRAMM TESTET DIES: VARIABLE A:"
869 PRINT"XXXXXXXXX I=GESETZT;0=NICHT GESETZT."
870 PRINT"XXXXXXXXX IN DIESEM FALLE:A=TEST(250,40)"
871 PRINT"XXXXXXXXX I:"A:PRINT"XXXXXXXXX"
    
```







# NA, HABEN SIE ES GELOEST?

DIESE LOESUNG IST SELBSTVERSTAEENDLICH  
NUR EIN MOEGLICHER VORSCHLAG. ANDERE  
LOESUNGEN SIND NATUERLICH DENKBAR

12.ROT = DREHEN EINER (DRAW)-FIGUR  
13.DRAW = ZEICHNEN EINER FIGUR



STATT 'UMSICHTBAR':SICHTBAR ZEICHNEN

DISK

```
951 REM IMT/BIKENBIBL, 1984
952 :
953 REM L O E S U N G : ROT+ D R A M
954 :
955 COLOUR14,14:REM RM,HG HELLBLAU
956 HIRE$0,14:REM ZF=0 (SCHMRZ); HG=HELLBLAU
957 REC 0,0,319,199,1 : REM RAHMEN-RECHTECK
958 :
959 REM *****
960 REM *****
961 REM ++ R O T + D R A M - BEFEBLE ***
962 REM *****
963 :
964 TEXT 30,12,"12.ROT = DREHEN EINER (DRAW)-FIGUR",1,1,8
965 TEXT 30,22,"13.DRAW = ZEICHNEN EINER FIGUR",1,1,9
966 ROT0,7:REM ROTATION KEINE, GROSSE 5
967 DRAW "55686568686877775",157,109,1
968 :
969 TEXT 20,180,"STATT 'UMSICHTBAR':SICHTBAR ZEICHNEN",1,1,8
970 WAIT 653,2 :REM COPY
```



## Anhang B: Panda-Listing (zu Kap. 3)

```
2 REM ZEICHNUNG MIT CIRCLE'S
6 REM IWT/BIRKENBIHL,1984
14 COLOUR 14,14:HIRES 0,14:REC0,0,319,199,1
42 TEXT 12,10,"DIESES PROGRAMM IST AUF DER DISKETTE",1,1,8
46 CIRCLE 152,90,79,65,1:REM GESICHT
54 CIRCLE 152,100,19,17,1:REM NASE
58 CIRCLE 152,135,17,10,1:PAINT 152,135,1:REM MUND
60 :
62 CIRCLE 145,100,5,5,1:PAINT 145,100,1:REM LINKES NASENLOCH
63 CIRCLE 157,100,5,5,1:PAINT 157,100,1:REM RECHTES NASENLOCH
64 :
65 LINE 151,102,152,130,2
69 PAINT 152,127,1
70 CIRCLE 111,75,27,29,1:PAINT 115,75,1:REM LINKES AUGE
78 CIRCLE 193,75,27,29,1:PAINT 192,75,1:REM RECHTES AUGE
86 CIRCLE 46,90,31,30,1:REM LINKES OHR
90 CIRCLE 258,90,31,30,1:REM RECHT.OHR
98 CIRCLE 152,194,120,42,1:PAINT 152,180,1
99 TEXT 10,178,"PANDA MIT CIRCLE,PAINT+LINE",2,2,11
198 WAIT 653,2:REM COPY

READY.
```

## Anhang C: POKE-Werte im Block (Tabelle)

Gehirngerechte Nachschlagetabellen für sämtliche POKE-Werte (für Zeichen von der Tastatur)

z.B. zu benutzen bei:  
HIRES/MULTI: CHAR; LO RES: FILL

PO 0

```

cccccc
cccccc
cccccc
cccccc

```

PO 1

```

AAAAAA
AAAAAA
AAAAAA
AAAAAA

```

PO 2

```

BBBBBB
BBBBBB
BBBBBB
BBBBBB

```

PO 3

```

CCCCCC
CCCCCC
CCCCCC
CCCCCC

```

PO 4

```

DDDDDD
DDDDDD
DDDDDD
DDDDDD

```

PO 5

```

EEEEEE
EEEEEE
EEEEEE
EEEEEE

```

PO 6

```

FFFFFF
FFFFFF
FFFFFF
FFFFFF

```

PO 7

```

GGGGGG
GGGGGG
GGGGGG
GGGGGG

```

PO 8

```

HHHHHH
HHHHHH
HHHHHH
HHHHHH

```

PO 9

```

IIIIII
IIIIII
IIIIII
IIIIII

```

PO 10

```

JJJJJJ
JJJJJJ
JJJJJJ
JJJJJJ

```

PO 11

```

KKKKKK
KKKKKK
KKKKKK
KKKKKK

```

PO 12

```

LLLLLL
LLLLLL
LLLLLL
LLLLLL

```

PO 13

```

MMMMMM
MMMMMM
MMMMMM
MMMMMM

```

PO 14

```

NNNNNN
NNNNNN
NNNNNN
NNNNNN

```

PO 15

```

OOOOOO
OOOOOO
OOOOOO
OOOOOO

```

PO 16

```

PPPPPP
PPPPPP
PPPPPP
PPPPPP

```

PO 17

```

QQQQQQ
QQQQQQ
QQQQQQ
QQQQQQ

```

PO 18

```

RRRRRR
RRRRRR
RRRRRR
RRRRRR

```

PO 19

```

$$$$
$$$$
$$$$
$$$$

```

PO 20

```

TTTTT
TTTTT
TTTTT
TTTTT

```

PO 21

```

UUUUUU
UUUUUU
UUUUUU
UUUUUU

```

PO 22

```

VVVVVV
VVVVVV
VVVVVV
VVVVVV

```

PO 23

```

WWWWWW
WWWWWW
WWWWWW
WWWWWW

```

PO 24

```

XXXXXX
XXXXXX
XXXXXX
XXXXXX

```

PO 25

```

YYYYY
YYYYY
YYYYY
YYYYY

```

PO 26

```

ZZZZZ
ZZZZZ
ZZZZZ
ZZZZZ

```

PO 27

```

llllll
llllll
llllll
llllll

```

PO 28

```

llllll
llllll
llllll
llllll

```

PO 29

```

llllll
llllll
llllll
llllll

```

PO 30

```

llllll
llllll
llllll
llllll

```



P0 31	P0 32	P0 33	P0 34	P0 35
+++++			.....	#####
+++++			.....	#####
+++++			.....	#####

P0 36	P0 37	P0 38	P0 39	P0 40
\$\$\$\$\$	#####	#####	//////	((((((
\$\$\$\$\$	#####	#####	//////	((((((
\$\$\$\$\$	#####	#####	//////	((((((

P0 41	P0 42	P0 43	P0 44	P0 45
)))))	*****	+++++	.....	-----
)))))	*****	+++++	.....	-----
)))))	*****	+++++	.....	-----

P0 46	P0 47	P0 48	P0 49	P0 50
.....	//////	00000	11111	22222
.....	//////	00000	11111	22222
.....	//////	00000	11111	22222

P0 51	P0 52	P0 53	P0 54	P0 55
33333	44444	55555	66666	77777
33333	44444	55555	66666	77777
33333	44444	55555	66666	77777

P0 56	P0 57	P0 58	P0 59	P0 60
88888	99999	.....	.....	<<<<<
88888	99999	.....	.....	<<<<<
88888	99999	.....	.....	<<<<<

P0 61	P0 62	P0 63	P0 64	P0 65
=====	>>>>>	22222	=====	#####
=====	>>>>>	22222	=====	#####
=====	>>>>>	22222	=====	#####

P0 66	P0 67	P0 68	P0 69	P0 70
	=====	=====	=====	=====
	=====	=====	=====	=====

P0 71	P0 72	P0 73	P0 74	P0 75
		#####	#####	#####
		#####	#####	#####



P0 76



P0 77



P0 78



P0 79



P0 80



P0 81



P0 82



P0 83



P0 84



P0 85



P0 86



P0 87



P0 88



P0 89



P0 90



P0 91



P0 92



P0 93



P0 94



P0 95



P0 96



P0 97



P0 98



P0 99



P0100



P0101



P0102



P0103



P0104



P0105



P0106



P0107



P0108



P0109



P0110



P0111



P0112



P0113



P0114



P0115



P0116



P0117



P0118

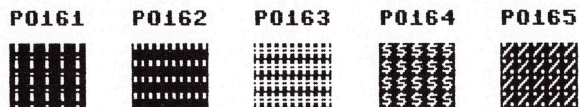
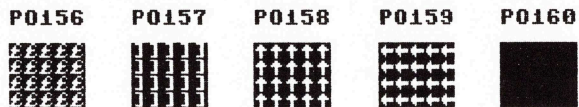
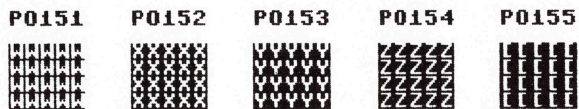
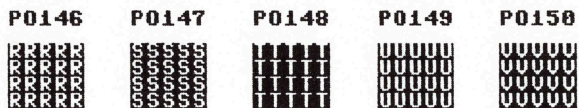
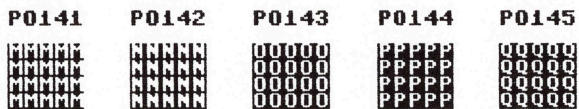
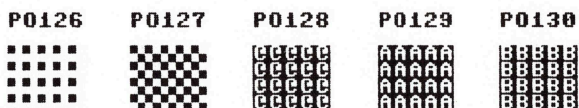
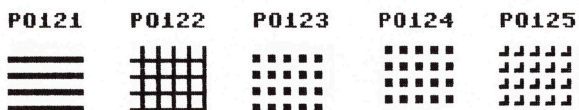


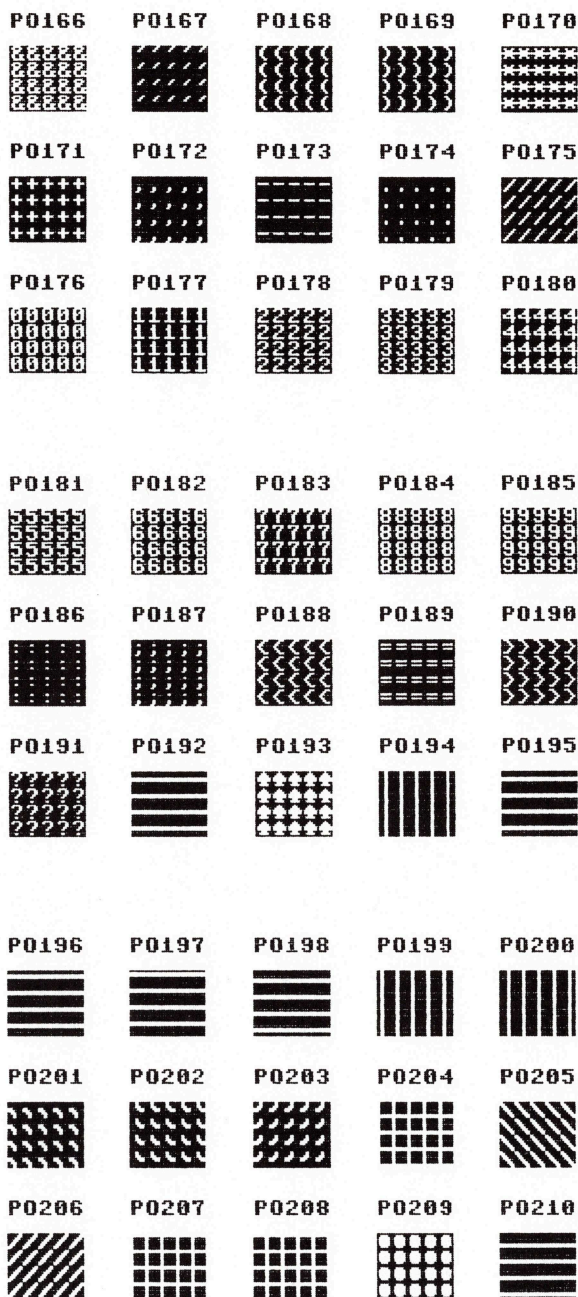
P0119

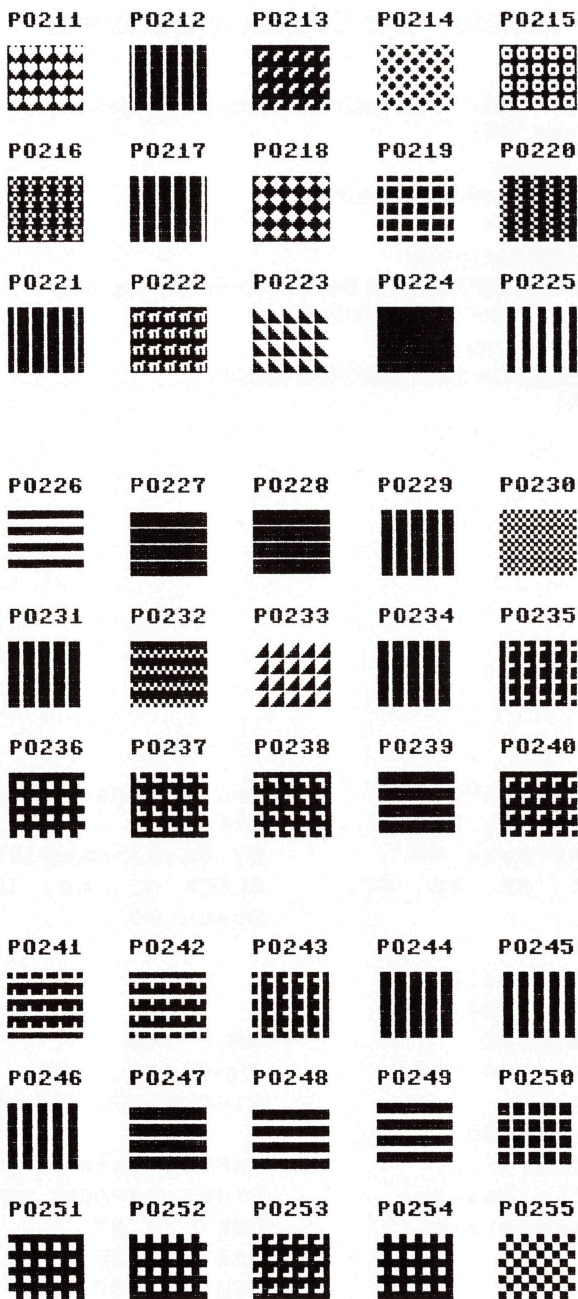


P0120











---

## Anhang D: Literatur- und Stichwortverzeichnis

BIRKENBIHL, Vera F.

**STROH IM KOPF?** Oder: Gebrauchsanweisung für's Gehirn  
2. Auflage, Speyer, 1984

COMMODORE (Hrsg.)

**SIMON'S BASIC 64** (offizielle Anleitung),  
Commodore Computer

ELSING/STERNER/WAGNER

**BASIC AUF DEM COMMODORE 64:** BASIC-Einführung und  
Erläuterung spezifischer Eigenschaften

PLENGE/SZCZEPANOWSKI

**DAS TRAININGSBUCH ZUM SIMON'S BASIC**  
Düsseldorf, 1984

Absprungszeile 169, 172  
Absolut 39, 46, 59  
absolute Adresse 40  
Abstand 21, 22, 66, 67,  
93  
Achteck 99  
Action-Spiele 110  
ANGL 78, 91, 104, 143  
Arbeitsplatz 70  
ARC 78, 91, 104, 141  
ARC-Befehl 34  
ARC-Parameter 95  
Ausdruck 159  
AUTOMatische Zeilen-  
numerierung 175

Basis-Programm 44, 70  
Bild 7  
Bildschirmkopien 160  
BLOCK 42, 54, 131  
Bogen 91  
CALL 169  
CHAR 111, 149  
CIRCLE 77, 78, 84, 104  
139  
CIRCLE-Befehl 95  
Club: 64-ACM 164  
CSET 1, 37  
CSET 2, 37  
CSET 0, 37

---

COLOUR 28, 44, 51  
COPY 172  
CURSOR 175  
CURSOR-Farbe 53

DATA 57  
DISK 178  
DIRectory 177  
DRAW 151  
DRAW-Figuren 116  
DRAW-String 118  
Dreieck 99  
DRUCKER:  
    CBM, alte 163  
    EPSON Fx 80, 163  
    EPSON Rx 80, 164  
    1525 (MPS 801) 160  
    1526 (CBM) 163

EXEC 169  
Editier-Modus 172, 175  
Ellipse 79, 83, 100, 104  
END 45  
End-Winkel 92, 98, 103  
Entfernungen 39

Farb-Änderung 30  
Farben 106, 162  
Farbram 21  
Farbenspiel 124  
Farbspeicher 162  
Feuerrad 124  
F-Tasten 35, 166, 167  
Figur 116  
FIND 171, 172  
FINDPAUSE 175

FINDWAIT 172  
FOR...NEXT 60, 84, 103,  
    108  
Format 171  
Formular 75  
Foto-Setzer 70  
Fuenfeck 99

Gerade 99  
gehirn-gerecht 6  
Geometrie 91  
geometrische Figuren 104  
Gesicht 89  
GOSUB 169, 172  
GOTO 169, 172  
Grafik 163

Halbkreis 103  
Hauswand 30  
Herr Links 8, 124  
Herr Rechts 8, 124  
HG 25  
HI COL 27, 35  
HIRES 15, 16, 21, 28,  
    32, 43, 49, 51, 79,  
    162, 163

INPUT 57, 114  
Invertierte Schrift 101  
JOYSTICK 110

Keller 30  
KEY 44, 166, 167  
Kleinanzeigen 70

---

---

Koordinaten-System 40  
Koordinatengitter 80  
Kreis 77, 83, 93, 98, 99

Leerstellen 171  
LINE 42, 59, 133  
LOOP 88  
LO RES 15, 21, 33, 160,  
163  
LOW COL 32,  
LOW COL + HI RES 31

Maschinenprogramm 163,  
166  
MERGE 176  
Mittelpunkt 78, 92  
Modus 25  
MPS 801, 163  
MULTI 15, 17, 21, 43, 49  
51, 79, 162, 163  
MULTI-Ausdruck 162  
MULTI-Befehl 28

Name 176  
NAME-MER 176  
Neue Farben 30  
Neuneck 99  
NEW COL 30 (s.LOW COL)  
NRM 21

Papier 53, 162  
Parameter 20, 88, 111  
PAUSE 37, 174, 175  
PAINT 21, 64, 135, 162

Pixel 16, 17  
PLOT 108, 145  
POKE 114, 167  
PRINT 172  
Printer/Plotter 160  
Programmschutz 170  
Puffer 163  
Punkte 108, 110

Quadrat 123

Radius 78, 104, 106  
Rahmen 25, 101  
Raute 99, 101  
REC 42, 129  
Rechteck 44, 51  
Regenbogeneffekt 106  
Relativ 39, 47  
Relativ "Adresse" 41  
REM 52, 172  
RENUMBER 45, 169, 170  
172  
RETURN 44, 175  
RND 57  
ROT 151  
ROTations-Winkel 118  
ROTIEREN 116

SAVE 177  
Schleifenzaehler 60  
Schrittweite 170  
Sechseck 99  
Senkrecht 80, 83  
Siebeneck 99  
"Sonnenbrille" 89  
Speicher 177

---

Start 176  
Start-Winkel 92, 98, 103  
Strecken 46  
Strings 171, 172

Telefonate 75  
TEST 110, 147  
TEST-Befehl 70  
TEST-RUN 160  
TEXT 66, 111, 137, 162  
Textverarbeitung 163  
Tiefe 21, 22, 66, 111

Ueberschriften 70  
Unterprogramm 45

Variable 60  
Videoram 21, 162  
Visitenkarten 70

waagrecht 80, 83  
WAIT 44, 172  
Warte-Funktion 44  
Winkel 92

x-Achse 40  
x-Radius 80  
x,y 46, 78  
x- und y-Achse 18, 39  
y-Achse 40  
y-Radius 80

Zeichenfarbe 51  
Zeichenketten 67  
Zeichensatz 163  
Zeichentyp 21, 55, 64,  
88, 94  
Zeichnen 116  
Zeilennummer, hohe 170  
Zielzeile 169, 172



● ○ ○

**J.Elsing D.Herrmann**

**WIRTSCHAFT  
AUF DEM  
COMMODORE  
64**

BASIC-Programme für den  
Anwender mit grafischer  
Darstellung

Programme auf Diskette/Kassette erhältlich

**iwv**

Eine Hilfestellung für wirtschaftliche Entscheidungen sind Programmsammlungen, die die guten Grafik- und Formmöglichkeiten des Computers nutzen. Diagramme, Sprites, optische Darstellungen von Simulationen werden eingesetzt, die die Ergebnisse verdeutlichen. Die finanzmathematischen Grundlagen sind zu jedem Programm beschrieben.

1983. 224 Seiten. Mit mehr. Abb. **Spiralhb.**  
DM 38,-/Fr. 38,-/S 342,-  
ISBN 3-88322-030-2

● ○ ○

**J.Elsing H.Sterner A.Wagner**

**MUSIK  
MIT DEM  
COMMODORE  
64**

Möglichkeiten  
der Musikprogrammierung

Programme auf Diskette/Kassette erhältlich

**iwv**

Bekanntlich verfügt der C 64 »von Haus aus« über einen Baustein, der die Erzeugung von mehrstimmiger Musik erlaubt. Sowohl der Anfänger ohne musikalische Vorkenntnisse wird angesprochen, als auch der Musiker, der seine Ideen mit Hilfe des Computers umsetzen möchte.

In Vorb. Juni 1984. Ca. 200 Seiten. **Spiralhb.**  
Ca. DM 38,-/ca. Fr. 38,-/ca. S 342,-  
ISBN 3-88322-046-9

● ○ ○

**H.Sterner T.Tutughamiarsa**

**MASCHINEN-  
SPRACHE  
AUF DEM  
COMMODORE  
64**

Eine Einführung  
mit zahlreichen Anwendungs-  
beispielen

Programme auf Diskette/Kassette erhältlich

**iwv**

Dieses Buch führt nach einer Kurzbeschreibung der Grundlagen direkt in die Maschinensprachen-Programmierung ein, ohne erst die Befehle »auf dem Trockenen« zu besprechen. Der Benutzer arbeitet sofort mit lauffähigen Programmen. Befehle werden dann eingeführt, wenn sie erforderlich sind.

In Vorb. Juli 1984. Ca. 260 Seiten. **Spiralhb.**  
Ca. DM 56,-/ca. Fr. 56,-/ca. S 498,-  
ISBN 3-88322-047-7

● ○ ○

**H.Dürkopf H.Sterner**

**CHEMIE  
MIT DEM  
COMMODORE  
64**

Anregungen und  
Programmbeispiele aus  
verschiedenen Bereichen

Programme auf Diskette/Kassette erhältlich

**iwv**

Programme reichen von stöchiometrischen Berechnungen bis zur Elementedatei. Besonders genutzt werden die grafischen Möglichkeiten. Die Anwendungsbeispiele gehen von Strukturformeln, grafischen Darstellungen von Versuchsanordnungen und Funktionen bis zur 3-D Grafik im Detail.

In Vorb. Juni 1984. Ca. 220 Seiten. **Spiralhb.**  
Ca. DM 48,-/ca. Fr. 48,-/ca. S 432,-  
ISBN 3-88322-049-3

● ○ ○

**J.Elsing H.Sterner A.Wagner**

**SPIELE UND  
SIMULATIONEN  
AUF DEM  
COMMODORE  
64**

Fertige Programme  
und Anregungen zum  
Selberprogrammieren

Programme auf Diskette/Kassette erhältlich

**iwv**

Dieses Buch enthält eine ganze Reihe von sofort lauffähigen Spiel- und Simulationsprogrammen, möchte aber auch dazu anregen, diese Programme zu verändern und weiterzuentwickeln. Besonders reizvoll dürfte es wohl sein, den »lernenden« Programmen noch etwas mehr »Intelligenz« zu verleihen.

1984. 208 Seiten. **Spiralhb.** DM 38,-/  
Fr. 38,-/S 342,-  
ISBN 3-88322-050-7

● ○ ○

**Vera F. Birkenbihl**

**EINSTIEG IN  
SIMON'S BASIC  
FÜR DEN  
COMMODORE  
64**

Schwerpunkt  
Grafik

Programme auf Diskette erhältlich

**iwv**

Grafikprogramme werden »gehirngerecht« aufbereitet, d.h. man sieht, wie Grafikbefehle »gehen«. Neue Art des Formats – man bekommt ein »Bild« des Befehls. Demo-Programme unterstützen das Gedächtnis. Bildschirm-Hardcopies als schnelles Nachschlagewerk. farbige Übersichtskarten zur Programmier-Erleichterung.

1984. 208 S. **Spiralhb.**  
DM 44,-/Fr. 44,-/S 396,-  
ISBN 3-88322-056-6



Der erste Band einer Reihe, die mit zahlreichen Programmen für Spiele und „ernsthafte“ Themen den Computer dem Benutzer näherbringt. Der Autor hat aus seiner Schulpraxis heraus Programme entwickelt, die das Lernen und Spielen mit dem Computer zum Vergnügen machen.

1983. 234 Seiten. **Kart. DM 32,-/Fr. 32,-/S 288,-**  
**ISBN 3-88322-013-2**



Jetzt wird es ernst: Hier wird Ihnen gezeigt, wie Sie mit dem Computer die Lohn- oder Einkommensteuerklärung erledigen, zeigt wohin die Staatsverschuldung geht – oder auch Ihre eigene, berechnet Ihre Zinsen (Soll oder Haben) auf der Bank, oder wie Sie Ihr Haus finanzieren können.

1983. 204 Seiten. **Kart. DM 32,-/Fr. 32,-/S 288,-**  
**ISBN 3-88322-014-0**



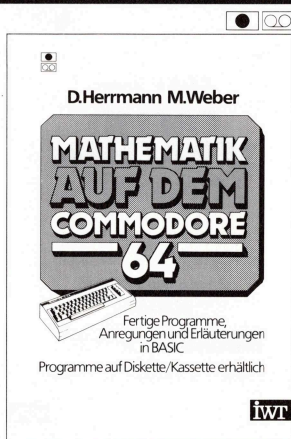
Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlen-theorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

1983. 248 Seiten. **Kart. DM 42,-/Fr. 42,-/S 378,-**. **ISBN 3-88322-016-7**



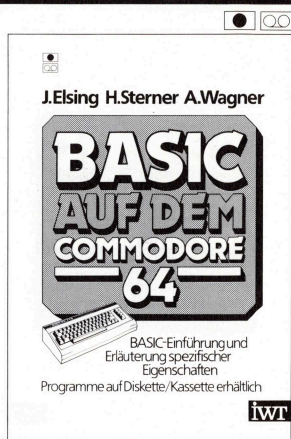
Der C 64 bietet vielseitige grafische Möglichkeiten. Dieses Buch gibt Informationen wie man Grafikfunktionen anwendet – Informationen, die man im Commodore-Handbuch nicht findet. Ausgehend von Grafiken mit den „festen“ Grafik-Zeichen wird systematisch zu den anspruchsvolleren Möglichkeiten, illustriert durch typische Beispiele, geführt.

1983. 138 S. 1 Folie. **Spiral. DM 38,-/Fr. 38,-/S 342,-**. **ISBN 3-88322-027-2**



Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlen-theorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

1984. 260 Seiten. **Kart. DM 42,-/Fr. 42,-/S 378,-**  
**ISBN 3-88322-048-5**



Dieses Buch bietet eine systematische Einführung in die Programmiersprache BASIC. Außer vielen kleineren Programmen zur Illustrierung der BASIC-Anweisungen gibt es eine umfangreiche Programmsammlung zu den verschiedensten Themenbereichen. Die besonderen Fähigkeiten des C 64 werden mit vielen Programmbeispielen erläutert.

1983. 356 Seiten. **Spiral. DM 56,-/Fr. 56,-/S 498,-**  
**ISBN 3-88322-029-9**



David Possin

# CBASIC

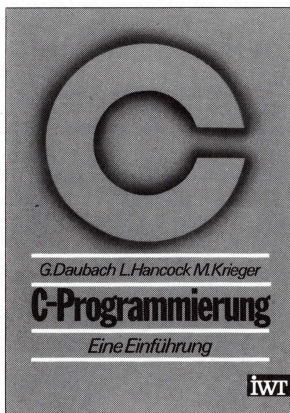
GSX C-CP/M DR-C

Einführung in die Verwendung von CBASIC für Grafik, Multitasking und Funktionsbibliotheken

iwt

Die Besonderheiten der 16-Bit Version von CBASIC werden erläutert. Spezielle Grafikbefehle sprechen das Betriebssystem GSX von DR direkt an. Weiter werden »C«-Funktionen mit CBASIC Programmen verknüpft, um so die Anwendung von Assembler zu umgehen. Vorteile von CP/M86, CCP/M86 und PC-DOS werden behandelt.

In Vorb. Juli 1984. Ca. 250 Seiten. Spiralh. Ca. DM 68,-/ca. Fr. 68,-/ca. S 612,- ISBN 3-88322-071-X



Die Programmiersprache »C« ist besonders zur Erstellung schneller, maschinennaher Programme geeignet, weist jedoch gegenüber der Assemblerprogrammierung wesentliche Vorteile auf, wie z. B. lokale und globale Variable, Prozeduren, Funktionen usw. Diese Einführung setzt beim Leser keine umfassenden Kenntnisse voraus.

In Vorb. 1984. Ca. 250 Seiten. Geb. ca. DM 56,-/ca. Fr. 56,-/ca. S 498,- ISBN 3-88322-041-8



Wer hat nicht bereits verzweifelt versucht, das »Computerchinesisch« zu verstehen? Hier hilft das Wörterbuch der Computerei mit seinen über tausend Begriffen. Außerdem sind die wichtigsten Begriffe erklärt. Ein handliches Nachschlagewerk für jeden, der sich mit Computerei beschäftigt.

1983. 2., erw. Aufl. 144 Seiten. Kart. DM 32,-/Fr. 32,-/S 288,- ISBN 3-88322-026-4

David Possin

# CBASIC

Access-Manager Display-Manager

Professionelle Programmierwerkzeuge für die effektive Programmerstellung

iwt

Zwei leistungsfähige Programme, die unter CP/M mit CBASIC professionell verknüpft werden: Access Manager für schnelle Dateiverwaltung, Display Manager als Maskengenerator. Diese und andere Einsatzmöglichkeiten werden beschrieben. Häufige Programmteile werden in Standardbibliotheken gespeichert und bei Bedarf neu geladen.

In Vorb. Juli 1984. Ca. 220 Seiten. Spiralh. Ca. DM 58,-/ca. Fr. 58,-/ca. S 522,- ISBN 3-88322-070-1



Das Buch behandelt den Einsatz von Mikrocomputern im Fremdsprachenlernprozess. Vom Inhalt: Vokabel-Trainingsprogramm, breiter Multiple-Choice Trainingsblock, Übungen zur Erweiterung des Wortschatzes, für Alltagssituationen etc. Zahlreiche Beispiele, engl. und franz., ausführlich kommentierte Programme.

In Vorb. Juli 1984. Ca. 220 Seiten. Geb. Ca. DM 58,-/ca. Fr. 58,-/ca. S 522,- ISBN 3-88322-055-8

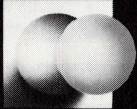


Das Buch ist sowohl für Programmierer als auch für Fans. Sie bekommen Einblick in das Zusammenwirken von zwei 6502-Prozessoren und den Datenverkehr über PIAs mit dem Rechner, dieses System arbeitet selbständig parallel. Ein Leckerbissen, wenn man den Computer noch besser nutzen will.

1982. 120 Seiten. Mit zahlr. Programmen. Ringb. DM 104,-/Fr. 104,-/S 936,- ISBN 3-88322-015-9

# Bernd Pol Vom Umgang mit CP/M

Eine allgemein-verständliche Einführung



CP/M für die Praxis 1

iwt

Das Buch ist in drei Teile gegliedert: Teil 1 führt in die Eigenschaften von Computern und CP/M im besonderen ein. Aufbauend auf diesen Kenntnissen werden im 2. Teil die zentralen CP/M-Hilfsprogramme vorgestellt. Der 3. Teil geht – nach einer Einführung in die Funktionsweise des 8080-Prozessors – auf die CP/M-Systembesonderheiten ein.

1982. 386 S. Mit zahlr. prakt. Beispielen. Geb. DM 48,-/Fr. 48,-/S. 432,- ISBN 3-88322-004-3

# Bernd Pol CP/M im Einsatz

Tips und Tricks für die Programmierung



CP/M für die Praxis 2

Band I

iwt

Dieser Band beschreibt wichtige Details des BDOS-Kerns und der CBIOS-Schnittstelle sowie Hinweise zur Fehlerverhütung. Weiter: Aufbau eines CBIOS-Systems, Fehlerbehandlung – Erweiterungsmöglichkeiten – Kompatibilitätsfragen zu MP/M und CP/Mplus.

In Vorb. 1984. Ca. 300 Seiten. Mit zahlreichen praktischen Beispielen.

Geb. Ca. DM 56,-/ca. Fr. 56,-/ca. S. 498,- ISBN 3-88322-006-X

W.Kitza

# dBase II Eine Einführung

Von der Idee zur Konzeption

iwt

Dieses Buch beschreibt an Beispielen aus der Praxis das Arbeiten mit dem wohl meistbenutzten Datenbanksystem der Welt. Es werden keine Vorkenntnisse vorausgesetzt, sondern es wird Schritt für Schritt das komplette Wissen vermittelt, das man zum Arbeiten mit einer Datenbank braucht.

1984. 226 Seiten. Geb. DM 56,-/Fr. 56,-/S. 498,- ISBN 3-88322-038-8

# Bernd Pol CP/M im Einsatz

Programme zur praktischen Arbeit



CP/M für die Praxis 2

Band II

iwt

Inhalt: Macrobibliothek zur strukturierten ASM-Programmierung, Standard-Prozeduren, System-Schnittst. zum Abfangen aller BDOS-Fehler unter CP/M 2.2, ähnlich CP/M+, MP/M II. Standardroutinen zur zeichenorientierten I/O. Programme in strukturierter Zwischensprache setzen relocierenden Makro-ASM RMAC von D.R. voraus.

In Vorb. 1984. Ca. 360 S. Mit zahlr. ausgearb. Progr. Geb. Ca. DM 56,-/ca. Fr. 56,-/S. 498,-. ISBN 3-88322-032-9

# Bernd Pol CP/M im Einsatz

Sammlung von System-Programmen



CP/M für die Praxis 2

Band III

iwt

Zentrale Programme für CP/M, Programme zur Nutzung der BDOS-Eigenschaften, Zentralen Datenzugriff, CBIOS-Schnittstelle, Dienstprogramme zur Erweiterung und direkten Diskettenzugriff, Routinen für Dateien. Programme kommentiert in strukturierter Zwischensprache. Relocierender Makro-ASM RMAC von DR nötig.

In Vorb. 1984. 350 Seiten. Geb. Ca. DM 56,-/ca. Fr. 56,-/ca. S. 498,-. ISBN 3-88322-033-7

W.Kitza

# dBase II Programmierung

Von der Konzeption zum Programm

Programme auf Diskette erhältlich

iwt

Das Buch, auch als Einstieg in das Programmieren geeignet, zeigt dBase II als mächtige Programmiersprache für die Datenbankanwendung. Inhalt: Was ist eine Datenbank? Warum dBase II? Wie wird programmiert? Einführung: Autokostenverwaltung; für Fortgeschrittene: Literaturverwaltung, kommentierte Listings, Tips und Tricks.

In Vorb. Juli 1984. Ca. 240 Seiten. Geb. Ca. DM 56,-/ca. Fr. 56,-/ca. S. 498,- ISBN 3-88322-039-6





mit Anwendungen für die Praxis.  
Programme auf Diskette erhältlich.

**iwt**

Dieses Buch führt Sie zur sicheren Handhabung der Tabellenkalkulation. Zahlreiche Anwendungen mit ausführlicher Beschreibung schließen sich an. Unter anderem: Werbeplanung, Reisekosten, Kapazitätsauslastung, Baufinanzierung, Reaktionszeiten. – Auch für die englische Version nutzbar.

In Vorb. Juni 1984. Ca. 250 Seiten. **Spiralh.**  
Ca. DM 56,-/ca. Fr. 56,-/ca. S 498,-  
ISBN 3-88322-074-4



mit Programmen und Masken  
für den Anwender.  
Auch auf Diskette lieferbar.

**iwt**

Dieses Buch wendet sich an SuperCalc-Anwender: Anfänger führt es schnell zur sicheren Handhabung, Fortgeschrittene finden ausgereifte Anwendungen. Beispiele mit ausführlicher Beschreibung machen das Buch auch für VisiCalc-Anwender interessant. Die Unterschiede zwischen VisiCalc und SuperCalc sind in einem Kapitel dargestellt.

1984. 248 Seiten. **Spiralh.** DM 56,-/  
Fr. 56,-/S 498,-  
ISBN 3-88322-040-X



mit Anwendungen für die Praxis.  
Teil 1: Kalkulation und Grafik  
Programme auf Diskette erhältlich.

**iwt**

Die Verbindung von Kalkulation, Grafik und Datenbank verhalf Lotus zu schnellem Erfolg. In diesem Buch erlernen Sie zunächst die sichere Nutzung von Kalkulation und Grafik. Zahlreiche Beispiele zeigen die Leistungsfähigkeit von 1-2-3: Umsatzprognose, Investitionsrechnung, Tilgungsplan, Private Ausgaben u.a.

1984. Ca. 250 Seiten. **Geb.** Ca. DM 58,-/  
ca. Fr. 58,-/ca. S 522,-  
ISBN 3-88322-085-X

G. Daubach D. Herrmann

Programme in  
**MICRO  
SOFT  
BASIC**

Band 1

**MATHE  
MATIK**

Fertige Programme, Anregungen  
und Erläuterungen in BASIC

Programme auf Diskette erhältlich

**iwt**

Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlentheorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

In Vorb. Juni 1984. 250 Seiten. **Spiralh.**  
Ca. DM 48,-/ca. Fr. 48,-/ca. S 432,-  
ISBN 3-88322-077-9

G. Daubach D. Herrmann

Programme in  
**MICRO  
SOFT  
BASIC**

Band 2

**WIRT  
SCHAFT**

BASIC-Programme für den Anwender  
mit grafischer Darstellung

Programme auf Diskette erhältlich

**iwt**

Eine Hilfestellung für wirtschaftliche Entscheidungen sind Programmsammlungen, die die guten Grafik- und Farbmöglichkeiten des Computers nutzen. Diagramme, Sprites, optische Darstellungen von Simulationen werden eingesetzt, die die Ergebnisse verdeutlichen. Die finanzmathematischen Grundlagen sind zu jedem Programm beschrieben.

In Vorb. Juni 1984. Ca. 200 Seiten. **Spiralh.**  
Ca. DM 48,-/ca. Fr. 48,-/ca. S 432,-  
ISBN 3-88322-076-0

G. Daubach H. Dürkopff H. Sterner

Programme in  
**MICRO  
SOFT  
BASIC**

Band 3

**CHEMIE**

Anregungen und Programm-  
beispiele aus verschiedenen Bereichen  
Programme auf Diskette erhältlich

**iwt**

Programme reichen von stöchiometrischen Berechnungen bis zur Elementedatei. Besonders genutzt werden die grafischen Möglichkeiten. Die Anwendungsbeispiele gehen von Strukturformeln, grafischen Darstellungen von Versuchsanordnungen und Funktionen bis zur 3-D Grafik im Detail.

In Vorb. Aug. 1984. Ca. 220 Seiten. **Spiralh.**  
Ca. DM 56,-/ca. Fr. 56,-/ca. S 498,-  
ISBN 3-88322-078-7

# IBM PC

## Wirtschaft

von G. Daubach  
und D. Herrmann

BASIC-Programme  
und grafische Darstellungen  
für den Anwender

Programme auf Diskette erhältlich

iwt

# IBM PC

## Mathematik

von G. Daubach  
und D. Herrmann

Fertige Programme,  
Anregungen und Erläuterungen  
in BASIC

Programme auf Diskette erhältlich

iwt

# IBM PC

## Chemie

von G. Daubach,  
H. Dürkopf  
und H. Sterner

Anregungen und  
Programmbeispiele aus  
verschiedenen Bereichen

Programme auf Diskette erhältlich

iwt

Eine Hilfestellung für wirtschaftliche Entscheidungen sind Programmsammlungen, die die guten Grafiken und Farbmöglichkeiten des Computers nutzen. Diagramme, Sprites, optische Darstellungen von Simulationen werden eingesetzt, die die Ergebnisse verdeutlichen. Die finanzmathematischen Grundlagen sind zu jedem Programm beschrieben.

In Vorb. Juli 1984. Ca. 200 Seiten. **Spiralhb.** Ca. DM 48,-/ca. Fr. 48,-/ca. \$ 432,- ISBN 3-88322-060-4

Dieses Buch enthält 40 mathematische Programme aus den Bereichen: Mehrregister-Arithmetik – Zahlen-theorie – Kombinatorik – Algebra – Geometrie – numerische Mathematik. Neu ist die Langzahl-Arithmetik. Sie gestattet die Grundrechenarten für Zahlen bis 255 Stellen.

In Vorb. Juli 1984. 250 Seiten. **Spiralhb.** Ca. DM 56,-/ca. Fr. 56,-/ca. \$ 498,- ISBN 3-88322-061-2

Programme reichen von stöchiometrischen Berechnungen bis zur Elementedatei. Besonders genutzt werden die grafischen Möglichkeiten. Die Anwendungsbeispiele gehen von Strukturformeln, grafischen Darstellungen von Versuchsanordnungen und Funktionen bis zur 3-D Grafik im Detail.

In Vorb. Aug. 1984. Ca. 220 Seiten. **Spiralhb.** Ca. DM 56,-/ca. Fr. 56,-/ca. \$ 498,- ISBN 3-88322-062-0

Guido Pahlberg



TI-BASIC  
Computer-Programme zum Spielen und  
Erweitern mit Tips, Kniffen und Tricks.  
Auch auf Kassette lieferbar.

iwt

Die Programme sind in TI-BASIC geschrieben, verwenden die Grundkonfiguration des Computers und machen ausgiebig Gebrauch von den Farbgrafikmöglichkeiten sowie dem Tongenerator. Sie sind so angelegt, daß der Einsteiger schnell Erfolge erzielt, der Fortgeschrittene aber die Spielprogramme nach Belieben ergänzen, erweitern oder variieren kann.

1983. 220 Seiten. **Kart.** DM 38,-/Fr. 38,-/ \$ 342,-. ISBN 3-88322-045-0

Günther Daubach

# TRS-80

## Assembler- Programmierung

Eine Einführung  
für Model I/III

iwt

Assembler-Programmierung mit Z80, dadurch breit einsetzbar. Nach Einführung eine Erklärung des Befehlssatzes mit Beispielen als Bausteine für eigene Programme. Weiteres: Ein- und Ausgabe über Tastatur und Bildschirm, Verwendung eingebauter Routinen, eines Editors/-Assemblers. Hilfreiches Nachschlagewerk.

In Vorb. Juli 1984. Ca. 250 Seiten. **Kart.** Ca. DM 52,-/ca. Fr. 52,-/ca. \$ 468,- ISBN 3-88322-017-5

Günther Daubach

# TRS-80

## BASIC- Trick-Buch

für Model I/III

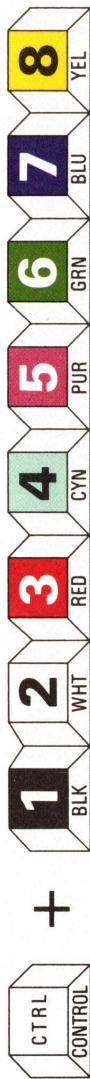
iwt

Für alle gedacht, die ihren TRS 80 besser nutzen wollen. Anfänger wie »alte Hasen« finden eine Vielzahl nützlicher Tips und Tricks. Schwerpunkte: wie Basic-Programme schneller werden; wie man Basic-Programme strukturiert; die Verbindung zwischen Basic- und Maschinenprogrammen, universelle Masken u.a.m.

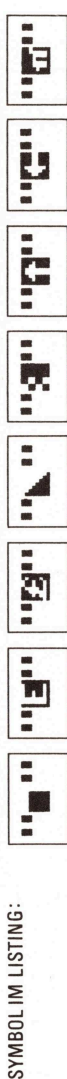
In Vorb. 1984. Ca. 300 Seiten. **Kart.** Ca. DM 52,-/ca. Fr. 52,-/ca. \$ 468,- ISBN 3-88322-018-3



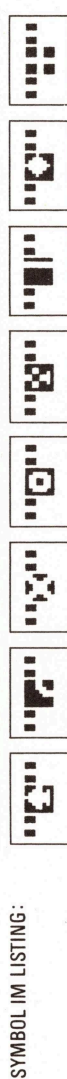




CTRL CONTROL	+	1	2	3	4	5	6	7	8
z.B. POKE 53280,2 Rahmen,rot		0	1	2	3	4	5	6	7
PRINT CHR\$(...)		144	5	28	159	156	30	31	158



C=	+	1	2	3	4	5	6	7	8
z.B. POKE 53281,10 Bildsch.,hellrot		8	9	10	11	12	13	14	15
PRINT CHR\$(...)		129	149	150	151	152	153	154	155



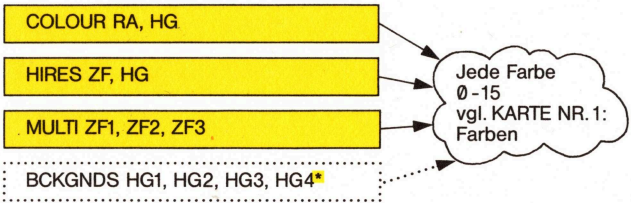






## Anhang E: Karte 2

### Farb- und Grafik-Modus-Befehle des Simon's BASIC



FARB-WECHSEL	MODUS-WECHSEL	
<p>alte Farben</p> <p>neue Farben</p> <p>HI COL</p> <p>LOW COL ZF1,ZF2,ZF3</p> <p>wie bei MULTI</p>	CSET 0 oder NRM	GG GROSS- SCHRIFT + Grafik; NoRMal
	CSET 1	GK GROSS- SCHRIFT + Klein- buchstaben
	CSET 2	holt alte HIRES- Grafik zurück

wie HIRES  
aber ohne Löschen

zum längeren  
Anschauen sollte ein  
PAUSE- oder WAIT-  
Befehl folgen

\* BCKGNDS= BaCKGrouNDS=Hintergründe: Dieser Befehl wurde in diesem Buch nicht besprochen, da er nur bei LO RES funktioniert: 4 verschiedene Hintergrundfarben (sog. extended color modus). Fehlt in der *sibas*-Anleitung; ausführliche Diskussion im Data Becker-Buch (Seiten 135-146).





## Anhang E: Karte 3

Alle HIRES/MULTI-Befehle: Formate

### ALLE HIRES/MULTI-BEFEHLE: FORMATE

1. REC X, Y, X1, Y1, ZT (RELATIV)
  2. BLOCK X, Y, X1, Y1, ZT (ABSOLUT)
  3. LINE X, Y, X1, Y1, ZT (ABSOLUT)
  4. PAINT X, Y, ZT (FLÄCHE FÜLLEN)
  5. TEXT X, Y, 'STRING', ZT, TI, AB
  6. CIRCLE X, Y, X1, Y1, ZT (ELLIPSE!)
  7. ARC X, Y, SW, EW, AB, X1, Y1, ZT
  8. ANGL X, Y, WINKEL, X1, Y1, ZT
  9. PLOT X, Y, ZT (NUR 1 PUNKT)
  10. VARIABLE = TEST (X, Y)
  11. CHAR X, Y, POKE, ZT, TI
  12. ROT DREHUNG, GROESSE
  13. DRAW 'nnn...' , X, Y, ZT
- ROT+DRAW IMMER ZUSAMMEN!





## Anhang E: Karte 4

Tabelle – Zeichentyp (HIRES + MULTI)

<b>Zeichentyp: HIRES+MULTI</b>	
<b>0 = IMMER HINTERGRUNDFARBE</b>	
<b>HIRES</b>	<b>MULTI</b>
<b>1: Zeichenfarbe</b>	<b>1: Zch. farbe NR.1</b>
<b>2: invertiert</b>	<b>2: Zch. farbe NR.2</b>
<b>NICHTS</b>	<b>3: Zch. farbe NR.3</b>
<b>NICHTS</b>	<b>4: invertiert:</b> <b>Farben 0 und 3</b> <b>Farben 1 und 2</b>

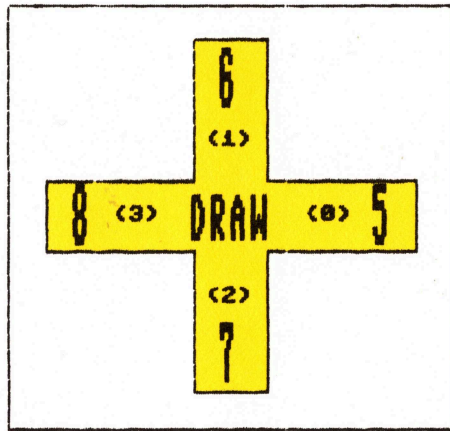






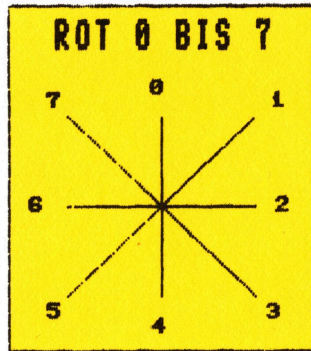
# Anhang E: Karte 5

## DRAW + ROT - Übersichten



$\emptyset$  = Keine ROTation  
 1 = 45°  
 2 = 90°  
 3 = 135°  
 4 = 180°  
 5 = 225°  
 6 = 270°  
 7 = 315°

Dies-selben  $\neq$  haben wir bei ARC kennengelernt  
 Hier leider nur als Code-ziffer!




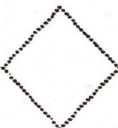

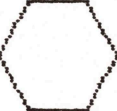

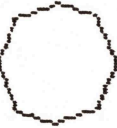








# Anhang E: Karte 6

## Figuren per ARC-Befehl

	<b>ABSTAND 120</b>		<b>ABSTAND 90</b>		<b>ABSTAND 72</b>
	<b>ABSTAND 60</b>		<b>ABSTAND 51</b>		<b>ABSTAND 45</b>
	<b>ABSTAND 40</b>	<b>FIGUREN PER ARC-BEFEHL</b>			<b>ABSTAND 36</b>









Vera F. Birkenbihl

# Einstieg in Simon's BASIC für den Commodore 64

## Schwerpunkt Grafik

Dieses Buch geht einen ungewöhnlichen Weg:  
Von der Gehirnforschung zur Computerei gekommen,  
berücksichtigt die Autorin die Arbeitsweise des Gehirns.  
Es werden Grafikbefehle wie HIRES/MULTI so erläutert,  
daß jeder Einsteiger

- kapiert, wie der Befehl funktioniert
- durch gehirn-gerechtes Vorgehen nachvollziehen kann,  
wie die Befehle »gehen«.

Spezielle Vorteile des Buches:

- gehirn-gerechte Demoprogramme (man liest am Bildschirm eine Information zu einem Befehl und sieht ihn anschließend ausgeführt);
- im Anhang befinden sich Übersichtskarten, so daß das ewige Herumblättern im Buch entfällt;
- zahlreiche quizartige Aufgabenstellungen, die schon durch die Fragestellung ein Problem erläutern;
- zahlreiche Abbildungen (Hardcopies) illustrieren das Geschriebene;
- die umfangreichen Programme sind auch auf Diskette erhältlich, das mühsame Eintippen entfällt.

ISBN 3-88322-056-6