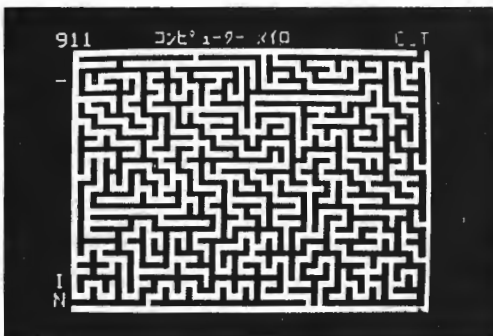


NAKAMOTOZU TINY BASIC

最終回 ファイル入出力ルーチンとプログラム例 山下春生

最近、構造化プログラミングの波がマイコン・ホビーの世界にも押し寄せて来ており、BASICに代ってPASCALのような手続き型言語が注目されています。このような言語のプログラムは、全体の構造がモジュール化されていて見やすく、これらと比較するとBASICのプログラムを見るのが厭になってきます。

今の時点では、望み過ぎの感もありますが、この違いは、インタープリタとコンパイラの違いと思っても良いかもしれません。インタ



ープリタでは、見やすくすればそれだけメモリーを食い実行速度が落ちるからです。

そこで、インタープリタの特徴を生かし、インタープリタでしかできない方向へ進んで来たもののひとつがTINY BASICであると思います。数Kバイトのメモリーで走り、エディタ機能を内蔵し、速度は遅いが一応リアルタイムで動く等。

NTBは、インタープリタでしかできない事は思いつく限り可能に、少しでも構造化言語に近づけようとしてできたものです。

ファイル・カセット 入出力サブルーチン

NTBでデータ・ベース、エディタ、アセンブラそしてコンパイラなどのプログラムを作った時、テキストやデータをテープに出力する必要があります。一般には、機械語ヘリンクして行ないますが、NTBでは、PROGRAM-5のようなサブルーチンでメモリ上（ここでは、1バイト単位）のファイルをかセットに入出力できます。

GRAPHIC-EDITORでは、カセットSAVEにプリント文を用いていました

が、このサブルーチンでは、画面へのエコーバックが不要の場合を考慮してACIAに、直接書き込みおよび読み出します。（9200行目と9250行目）

TAPEのフォーマットは、図3とほぼ同様の形式ですが、STXとCRの間はアスキーコードのテキストとし、行間のディレイはとっていません。ファイルの最後のCRの次にCont-D（\$04）を付けて、ファイルの終わりを表わしています。ファイルをLOADする場合、一行ごとにSTXマークのフラグ待ちを行なっていますので、行間にディレイを置いて録音した場合でも読み取りエラーは起こりません。

メモリ上のファイル形式および録音フォー

●PROGRAM 5

```

X.
9000 REM CASSET LOAD
9010 F=0:P.C.17
9020 DO
9030 DO:GOS.9250
9040 U.(D=2)+(D=4)
9050 IFD=4 F=1:G.9080
9060 DO:GOS.9250:#Z=D:Z=Z+1
9070 U.D=#0D
9080 U.F=1:P.C.19:#Z=4
9090 RET
9100 REM CASSET SAVE
9110 P.C.18;
9120 F.I=0T01000:N.I
9130 DO:D=2:GOS.9200
9140 DO:D=#Z:GOS.9200:Z=Z+1
9150 U.D=#0D
9160 U.#Z=4:D=4:GOS.9200:P.C.20
9170 RET
9200 REM SUB OUT
9210 DO:U.AND(##E010,2):##E011=D
9220 RET
9250 REM SUB IN
9260 DO:U.AND(##E010,1):D=##E011
9270 RET

READY
    
```

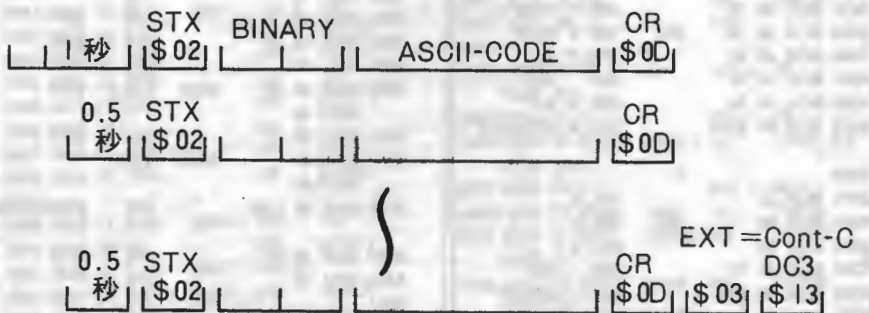
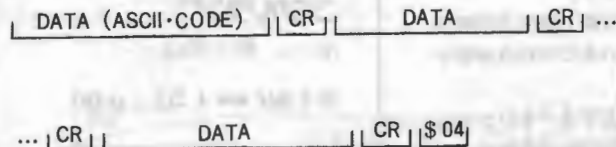


図3 N.T.B TAPE FORMAT

◇ メモリ上



メモリ上のファイルの先頭ADRS をZ にセットして、GOSUB 9100すると、次のように出力され、Z は\$04のADRS を示してRETURN する。

◇ テープ上

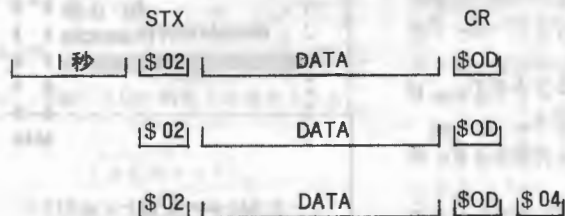
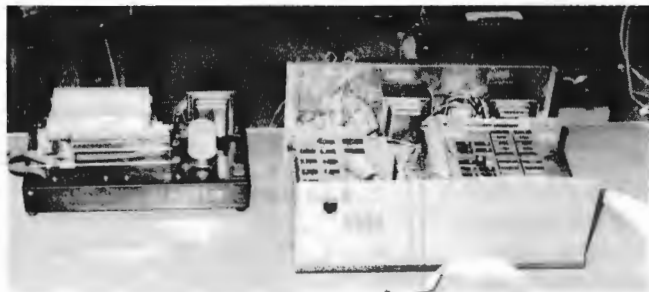


図5 ファイル・カセット入出力サブルーチンのフォーマット



▲筆者のシステム、H68/TRをベースにした自作システムと放電プリンタ。

マットを図5に示します。

このサブルーチンは、ACIAのADRSの変更のみでどの機種でも使用できます。

バリケード・ゲーム

V-RAM上を#とのキャラクタが方向を変えながら何物とも衝突しないようにバリ

ケードを築くゲームで先に衝突した方が負けで相手の得点が1点プラスされ10回終了した時の合計点を争うゲームです。

このゲームの魅力は、何と言ってもリアルタイムのスピード感と様々なテクニックにあります。

H-68用キーボードを2分割してKEY関数を使用し、同時に2人がデータを与え続け

●PROGRAM6

```

READY
XL.
10 *****
20 * N.T.B. バリケード ゲーム *
30 * BY 瀬才 千代子 *
40 * H68-KEYS *
50 * [ □ ] [ # ] *
60 * 1 4 *
70 * 6 8 9 B *
80 * D G *
85 * RESTART : 'S' KEY *
90 * 1 ゲーム 10 回 *
95 *****
100 V=$C000:P=0:Q=0:J=1
110 S=$FF-" ":C=$FF-"□":D=$FF-"#":E=$FF-"::F=$F
F-"■"
200 DO:CLR:T=R.(7)+2
210 F.I=0T015*32 S.32:$(U+I)=E:$(U+I+31)=E
220 N.I
230 R=R.(2):A=$12-2*R:H=U+R.(14)*32+R*31+32
240 R=R.(2):B=$15-2*R:M=U+R.(14)*32+R*31+32:IFH=
M T.240
250 REM LOOP
260 #H=C:#M=D:P.C.6;
270 F.I=0TOT
280 K=K.(0,2):IF(K=$01)+(K=$10)+(K=$12)+(K=$21)=
1 A=K
290 K=K.(3,5):IF(K=$04)+(K=$13)+(K=$15)+(K=$24)=
1 B=K
300 N.I
310 H=((A=$21)-(A=$01))*32+(A=$12)-(A=$10)+H
320 M=((B=$24)-(B=$04))*32+(B=$15)-(B=$13)+M
330 IFH=M #H=F:G.400
340 IF#H<>S T.370
350 IF#M=S T.250
360 P=P+1
370 IF#M=S Q=Q+1
400 CLR(0,1):CU.(3,0):P.J;" ゲーム *** # [";Q;" ] ,
□ [";P;" ]"
410 P.C.7:J=J+1:IFJ=11 U.1:G.450
420 DO:U.KEY=$44
430 P.C.6:C.7;
440 U.0
450 P." AGAIN ? ( Y OR N ) ?";
460 IF GET$="Y" T.100

```

READY

る事ができます。(他のキーボードの人は無理だと思う。)

毎回の2つのバリケードの出発点とスピードは、乱数で変化し、同時に衝突した場合と同時に同じ場所へ行こうとした場合は、引き分けの処理をしています。

使用するキーは、プログラム先頭のコメント中で説明してあり、一度ある方向のキーを

目次ダイジェスト

インターフェイス・エイジ誌

78年12月号

GENERAL FEATURES

1978 INDEX TO MICROCOMPUTER BOOKS 54
FIRST ANNUAL LISTING FOR BOTH THE HOME AND THE ADVANCED MICRO USER

WORDPROCESSING: A MAJOR FACTOR IN SMALL BUSINESS COMPUTING? 54
AN OVERVIEW OF THE ROLE OF WORD PROCESSING FOR THE BUSINESSMAN
by Carl Hansen, Senior Editor

IT'S NOT A BIG MIRACLE 70
A FATHER USES A MICRO TO MONITOR HIS QUARTIC SON
by Matthew Turinsky

THE INFOTEC IMP: AN ALTERNATIVE MACHINE 74
A BUSINESS SYSTEM STRUCTURED AROUND EXISTING SOFTWARE
by Terry Costlow, Assistant Editor

BUSINESS FEATURES

BUSINESS EDITORIAL — WHY IS THERE SO LITTLE GOOD BUSINESS SOFTWARE? 77
by Keith Parsons, Structural Systems Group

SOME BASICS OF ACCOUNTING 78
by James C. Hamilton, Structural Systems

SEQUENTIAL FILES — WEDGE 84
by Martin Fildrig

THE TECHNIQUES OF DATA ENTRY 87
by William S. Ombler

HARDWARE FEATURES

CARD OF THE MONTH: VECTOR GRAPHIC INC. PROGRAM BOARD 82
by Roger W. George, Hardware Editor

BUILDING AN INEXPENSIVE 2708 EPROM PROGRAMMER 94
by Daniel J. Van Auer

A REVIEW OF THE HEATH H11 MICROCOMPUTER 106
by Barry J. Andrews

SOFTWARE FEATURES

PEEK AND POKE A TOTAL 129
by Douglas L. Jones

CP/M: AN 8080Z-80 DISK OPERATING SYSTEM — PART 2 190
by Alan R. Miller, Contributing Editor

MS800 DATA OR PROGRAM ADDRESS LOCATOR 136
by G. Gordon White

PICO-FUMI 138
by Kenneth D. Stanger

79年2月号

GENERAL FEATURES

MULTITASKING THE IBM FABOS, AN ON-LINE SYSTEM 54
by Alan R. Miller, Contributing Editor

A SIMPLE FINANCIAL REPORT WRITER 64
by Alan R. Miller, Contributing Editor

BUSINESS AND COMPUTERS: THE PERFECT MARKET 76
by Alan R. Miller, Contributing Editor

TECH AND THE COMPUTER 78
by Alan R. Miller, Contributing Editor

BUSINESS FEATURES

BUSINESS EDITORIAL — MICROCOMPUTING AND ACCOUNTING 80
by Alan R. Miller, Contributing Editor

NOTARY 81
by Alan R. Miller, Contributing Editor

NEW TERMINAL, NEW IDEA 84
by Alan R. Miller, Contributing Editor

THE REAL WORLD OF MICROCOMPUTING 88
by Alan R. Miller, Contributing Editor

MEASURING BUSINESS ACCOUNTS 89
by Alan R. Miller, Contributing Editor

HARDWARE FEATURES

CARD OF THE MONTH: MICROGNOS INC. — BETTER BUG TRAP 93
by Alan R. Miller, Contributing Editor

2-80: IS IT REALLY BETTER? 94
by Alan R. Miller, Contributing Editor

NEW MINI SERIES OF BASIC ELECTRONICS 98
by Alan R. Miller, Contributing Editor

SOFTWARE FEATURES

PROGRAMMING TECHNIQUE — NIGHT 2 130
by Alan R. Miller, Contributing Editor

BURN YOUR OWN PROMS: BUILDING AND RESOLVING A PROM PROGRAMMER 138
by Alan R. Miller, Contributing Editor

MACROPROCESSING BASES THE SOFTWARE DEVELOPMENT TASK 136
by Alan R. Miller, Contributing Editor

ON-LINE MACHINE LANGUAGE SUBROUTINES FOR BASIC INTERPRETERS 140
by Alan R. Miller, Contributing Editor

FREE RUNNING

BUSINESS SOFTWARE REVIEW 18
by Alan R. Miller, Contributing Editor

THE COLUMN 4
by Alan R. Miller, Contributing Editor

EDITOR'S NOTEBOOK 4
by Alan R. Miller, Contributing Editor

FROM THE FOUNTAINHEAD 4
by Alan R. Miller, Contributing Editor

INVENTOR'S SKETCHPAD 4
by Alan R. Miller, Contributing Editor

JURISPRUDENT COMPUTERIST 4
by Alan R. Miller, Contributing Editor

LETTERS TO THE EDITOR 4
by Alan R. Miller, Contributing Editor

MICRO MATHEMATICAL 4
by Alan R. Miller, Contributing Editor

MICRO MEDICINE 4
by Alan R. Miller, Contributing Editor

MIND REVOLUTION 4
by Alan R. Miller, Contributing Editor

WHITE DOLLAR MICROCOMPUTER 4
by Alan R. Miller, Contributing Editor

DEPARTMENTS

ADVERTISER INDEX 160
by Alan R. Miller, Contributing Editor

CALENDAR 161
by Alan R. Miller, Contributing Editor

PIFO 162
by Alan R. Miller, Contributing Editor

MICRO MARKET 163
by Alan R. Miller, Contributing Editor

NEW PRODUCTS 164
by Alan R. Miller, Contributing Editor

UPDATE 165
by Alan R. Miller, Contributing Editor

79年4月号

GENERAL FEATURES

A ROBOT IN A HELL 58
by Alan R. Miller, Contributing Editor

A ROVER ROBOT SCOPES SON AND TESTS WATER 60
by Alan R. Miller, Contributing Editor

REACHING FOR THE FUTURE 62
by Alan R. Miller, Contributing Editor

THE BEGINNING OF A NEW ERA: THE WORKING ROBOTS 64
by Alan R. Miller, Contributing Editor

A ROBOT THAT DOES NOTHING UNTIL CONTROLS ARE ADDED 66
by Alan R. Miller, Contributing Editor

BUSINESS FEATURES

GESTATION CALCULATIONS 70
by Alan R. Miller, Contributing Editor

INDUSTRIAL ROBOTS 72
by Alan R. Miller, Contributing Editor

GA: I HEAR YOU 76
by Alan R. Miller, Contributing Editor

PROTOTYPE A LOW-COST HIGH QUALITY WORD PROCESSOR 80
by Alan R. Miller, Contributing Editor

TUTORIALS

NEW MINI SERIES — UNIT #2 82
by Alan R. Miller, Contributing Editor

PROGRAMMING TECHNIQUE — NIGHT 4 82
by Alan R. Miller, Contributing Editor

HARDWARE FEATURES

CARD OF THE MONTH: ONKO SCIENTIFIC SINGLE BOARD COMPUTER 96
by Alan R. Miller, Contributing Editor

HIGH PERFORMANCE, LOW COST NEW PRINTER 102
by Alan R. Miller, Contributing Editor

SOFTWARE FEATURES

THE 80A DROPPER FOR CP/M 136
by Alan R. Miller, Contributing Editor

ASBE COMPREHENSIVE MEMORY TEST PROGRAM 140
by Alan R. Miller, Contributing Editor

ASTROLOGICAL HOROSCOPE PROGRAM 146
by Alan R. Miller, Contributing Editor

FREE RUNNING

ASMR JOURNAL 18
by Alan R. Miller, Contributing Editor

BUSINESS SOFTWARE REVIEW 47
by Alan R. Miller, Contributing Editor

THE COLUMN 29
by Alan R. Miller, Contributing Editor

EDITOR'S NOTEBOOK 4
by Alan R. Miller, Contributing Editor

FROM THE FOUNTAINHEAD 4
by Alan R. Miller, Contributing Editor

INDUSTRY NEWS 4
by Alan R. Miller, Contributing Editor

INVENTOR'S SKETCHPAD 4
by Alan R. Miller, Contributing Editor

JURISPRUDENT COMPUTERIST 4
by Alan R. Miller, Contributing Editor

LETTERS TO THE EDITOR 4
by Alan R. Miller, Contributing Editor

MICRO MATHEMATICAL 4
by Alan R. Miller, Contributing Editor

MICRO MEDICINE 4
by Alan R. Miller, Contributing Editor

R.H.D. 4
by Alan R. Miller, Contributing Editor

DEPARTMENTS

ADVERTISER INDEX 162
by Alan R. Miller, Contributing Editor

BOOK REVIEWS 163
by Alan R. Miller, Contributing Editor

CALENDAR 164
by Alan R. Miller, Contributing Editor

CONNECTIONS 165
by Alan R. Miller, Contributing Editor

MICRO MARKET 166
by Alan R. Miller, Contributing Editor

NEW PRODUCTS 167
by Alan R. Miller, Contributing Editor

UPDATE 168
by Alan R. Miller, Contributing Editor

これらの本は、アスキー出版にて販売しています。
☎ 03-407-4910にて在庫をお問合わせください。

押すその他の方向のキーを押すまでその方向を維持するようになっています。

ゲーム中は、両端を除く30×16マスが有効で、各回終了と同時にそれまでの得点が表示されます。

私のV-RAMは、DATA-BUSが反転のため、PEEKやPOKEで直接V-RAMを操作する時は、反転コードを使用していますが、使用するコードは、すべて行番号110で変数に代入して使っていますので、この行の変更だけで済みます。(V-RAM先端ADDRESSは、行番号100.)

行番号310と320は、ひとつの式で、4方向のKEY-CODEをデコードして、ポインタを動かしています。

毎回Sを押すごとにRESTARTし、10回終了後は、Yで再ゲームです。一般に、動きのあるゲームを作るとき、絵を動かす方法には、

- (1) CURSORを設定して、プリント文で絵を書く。(大きなパターンを一瞬で出力できる。)
- (2) CURSORを設定しCHR\$を用いたプリント文でグラフィック記号を出力する。(DATAにより、多種のパターンを出力できる。)
- (3) POKEを用いる。(小さなパターンを高速で動かすのに向き、PEEKでパターンを読むことが可能。)
- (4) IX配列を用いる。(特に横方向2バイトのパターンのとき有利であり、配列の添え字を変化させると2バイトごとに動き、対応するIX変数の方を変化させると2バイトのパターンが1バイトごとに動く。(4)と同様に、読むことができる。)
- (5) グラフィック・ステートメントを用いる。(1ドット単位で点を動かしたり、数式で表わされる曲線図型を描くのに適する。!P関数を用いてドットを読むことができる。)

NTBのグラフィック改造を行なうと以上のどの方法でグラフィック・ディスプレイしても、同じ絵であれば結果的には同じものができるという利点があり、これをうまく利用したものがGRAPHIC-EDITORだと言えます。

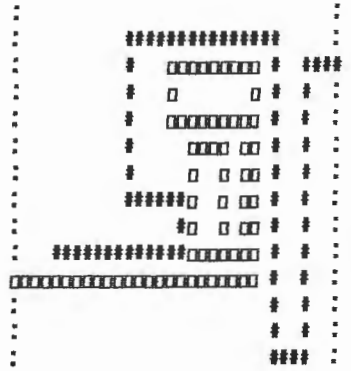
π の計算

π (円周率) の計算は、前回行なったe(自然対数の底)の計算よりずっと複雑ですが、配列を任意有効桁数のアキュムレータとして使用する考え方は同じであり、昔から電子計算機の歴史のようなものですので、NTBでも行なってみました。

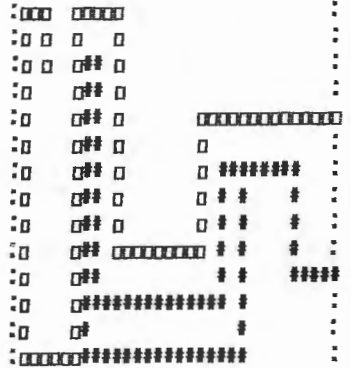
ここでは、一般に多く用いられているマチンの公式を用いることにします。

●パルクードゲーム・サンプルラン

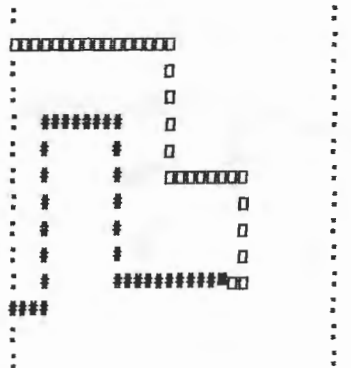
1 カメ *** # [1] , □ [0]



2 カメ *** # [1] , □ [1]



3 カメ *** # [1] , □ [1]



$$\pi = 16 \cdot \tan^{-1} \frac{1}{5} - 4 \cdot \tan^{-1} \frac{1}{239}$$

$$= 16 \left[\frac{1}{5} - \frac{1}{3 \cdot 5^3} + \frac{1}{5 \cdot 5^5} - \frac{1}{5 \cdot 5^7} + \dots \right]$$

$$- 4 \left[\frac{1}{239} - \frac{1}{3 \cdot 239^3} + \frac{1}{5 \cdot 239^5} - \frac{1}{5 \cdot 239^7} + \dots \right]$$

この公式の特長は、収束が速いことであり
次の変形をした式を忠実にアルゴリズムとします。

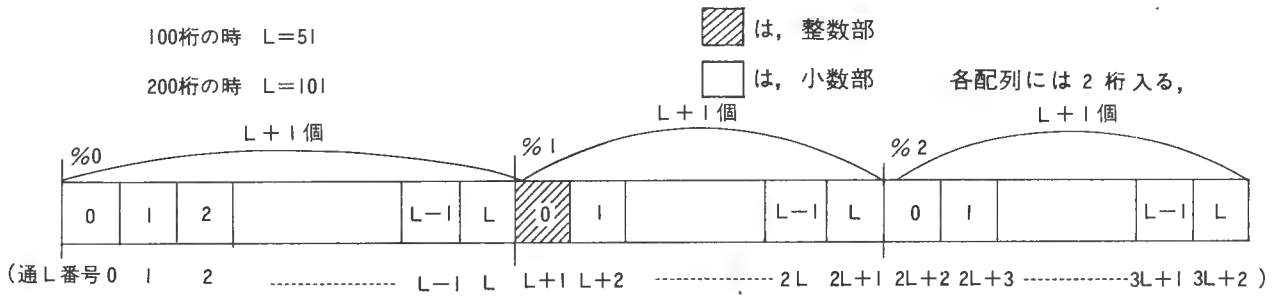


図6 πの計算のためのアキュムレータ設定

$$\pi = 16 * 5 \left\{ \frac{1}{5^2} - \frac{1}{3 \cdot 5^4} + \frac{1}{5 \cdot 5^6} - \frac{1}{7 \cdot 5^8} + \dots \right\}$$

$$+ 4 * 239 \left\{ \frac{1}{239^2} - \frac{1}{3 \cdot 239^4} + \frac{1}{7 \cdot 239^6} - \dots \right\}$$

例えば、小数以下100桁まで計算するとき、

整数部2桁と小数部102桁(打ち切り誤差を除くため.)の計104桁(2進化100進数で表わすので配列を52使う.)のアキュムレータを3個用意します。(%0, %1および%2)

学生時代最後の良い思い出を作ってください
ASCII誌に感謝致します。

〈4月号NTBの訂正とお詫び〉

ASCII 4月号発表のNAKAMOZU TINY BASIC (山下春生氏)について表現方法、記載もれの御指摘がありましたので訂正させていただきます。

1. P75のサムチェック・プログラムはアセンブラで書かれたものですが誌面の都合で省略されています。P89日下田氏作のプログラムを参考にサムチェックを行って下さい。
2. P76のサムチェック・プログラムはオブジェクトダンプ用プログラムとして投稿いただいたものです。又製版上のミスで行番号10が欠落してしまいましたので以上のように訂正させていただきます。

サムチェック・プログラム
→オブジェクトダンプ・プログラム
10 P. "*** メモリー ダンプ
***"

3. 山下春生氏作のI/Oルーチン掲載が誌面の都合により6月号になってしまい本文の記載と食い違う部分が出てしまいました。
4. 日下田氏の記事中のNTBのSAVE及びLOADルーチンは、まだ多少安定性に欠ける…… (P88)の記載は、H68TVが、VRAMに対して同期書き込みを行っているためでH68のハードウェア構成に依存するもので、ソフトウェアに一切問題はありせん。誤解を招く表現をお詫びすると共に、P88②を削除させて頂きます。

図7は、そのアルゴリズムであり、上半分と下半分はそれぞれ前と後の級数に対応しています。

前の級数では、M=5²として[%0]を割っていますが、後の級数では、M=239として2回[%0]を割るアルゴリズムになっています。これは、239²=57121となり、NTBの変数に入らないからで他意はありません。

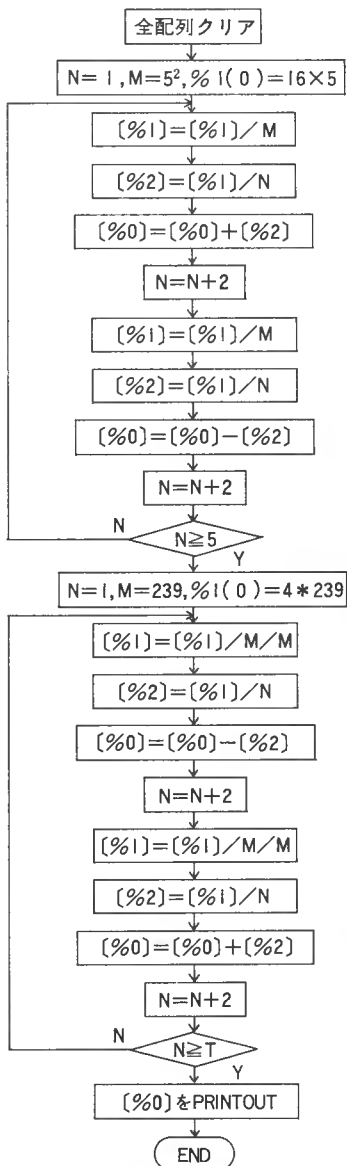
配列をアキュムレータとした除算、加算そして減算は、それぞれ4つのサブルーチンにしてあり、フローチャート通り構成しています。

実行時間100桁で2分50秒、200桁のときは、11分20秒になりました。他のTINY-BASICを使用の人も挑戦してください。

最後に

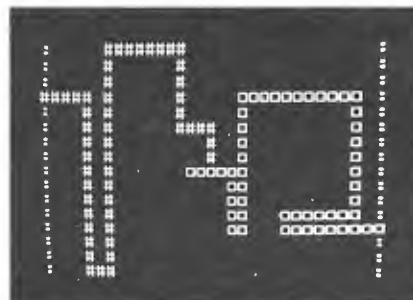
NTBは、電大版を基にして私の思い付くままに拡張と高速化を進めていくうちに、完成した(一応)ものであり、自分のシステムを最大限に活用しようとしたので、すべての人が命令を使用できるとは限らないと思います。

システム依存性が高いリストを発表する点は気が引けたのですが、少なくとも既存のTINYでは、不可能な事も実現できる機能を持っていると考えて発表することにしました。移植に成功された方は、良いプログラムをどんどん発表して戴けば幸いです。

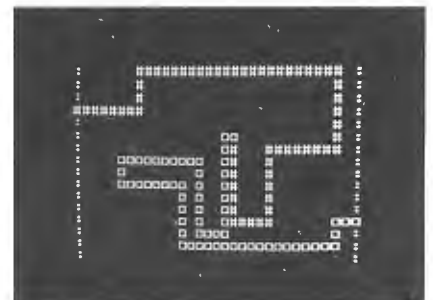


注: [%0]は、%0配列を用いたアキュムレータを表わす

図7 πの計算のアルゴリズム



▲バリエードゲーム第1回戦口の勝ちノ



▲第2回戦またも口の勝ちノ

5. 大西氏製作のGAME IIIを参考にしては
いない由の御指摘が山下氏よりありました。
同氏よりの別のお手紙より“目標を、VTL
・GAME・PALOALTO・電大版の全
ての機能を含むことに置いたからで、……”
(原文そのまま)の一文が欄外の表現となつ

たわけです。

ASCII誌は原則として、スタッフ・ライ
ティング制度を採用しており、御投稿いただ
いた原稿はASCIIスタッフにより加筆修正
を加えています。一枚のプリンタ用紙の裏に
書いた走り書きから一本の記事が生まれるこ

とさえあります。加筆の際にも筆者の意図を
そのまま伝えるべく努力し、良りよい誌面を
お届けしたいと考えます。(文責編集部)

●PROGRAM 7

```
>
READY
>L.
10 *****
20 *   ノイノケイケン   *
30 *   N.T.B. H. YAMASHITA   *
40 *****
60 G.400
100 REM DIVC%1=%1/M]
110 F.I=0/1 TOL
120 %1(I)=(%1(I)+K*MOD)/M
130 N.I
140 RET
150 REM DIVC%2=%1/N]
160 F.I=0/1 TOL
170 %2(I)=(%1(I)+K*MOD)/N
180 N.I
190 RET
200 REM ADDC%0=%0+%2]
210 C=0
220 F.I=L T00 S.-1
240 W=%0(I)+%2(I)+C:C=0:IFW<K W=W-K:C=1
250 %0(I)=W
260 N.I
270 RET
300 REM SUBC%0=%0-%2]
310 C=0
320 F.I=L T00 S.-1
340 W=%0(I)-%2(I)-C:C=0:IFW<0 W=W+K:C=1
350 %0(I)=W
360 N.I
370 RET
400 REM MAIN PROGRAM
410 IN." 79ズウ =100*D   D = "D
420 L=50*D+1:S=145*D:T=45*D
430 K=100:Z=2*L+2:%0=7+1:%1=%0+Z:%2=%1+Z
500 F.I=0T03*L+2:%0(I)=0
510 N.I
520 N=1:M=5*5:%1(0)=16*5
530 DO
```

```
540 REM 1 [+ ]
550 GOS.100:GOS.150:GOS.200:N=N+2
570 REM 1 [- ]
580 GOS.100:GOS.150:GOS.300:N=N+2
590 U.N)=S
600 F.I=0T02*L+1:%1(I)=0
610 N.I
620 N=1:M=239:%1(0)=4*239
630 DO
640 REM 2 [- ]
650 GOS.100:GOS.100:GOS.150:GOS.300:N=N+2
670 REM 2 [+ ]
680 GOS.100:GOS.100:GOS.150:GOS.200:N=N+2
690 U.N)=T
700 REM TYPE OUT
710 CLR:CU.(0,0):P."   ノイ = ";%0(0)";"
720 F.I=1TOL-1
730 P.%0(I)/10;" ";MOD;" ";
740 IFMOD(I,5)=0 P." ";
750 IFMOD(I,10)=0 P.
760 N.I
770 P."   ノイ   ";100*D;"   ノイ"
```

```
READY
>R.
79ズウ =100*D   D = ? 2
ノイ = 3.
1 4 1 5 9 2 6 5 3 5   8 9 7 9 3 2 3 8 4 6
2 6 4 3 3 8 3 2 7 9   5 0 2 8 8 4 1 9 7 1
6 9 3 9 9 3 7 5 1 0   5 8 2 0 9 7 4 9 4 4
5 9 2 3 0 7 8 1 6 4   0 6 2 8 6 2 0 8 9 9
8 6 2 8 0 3 4 8 2 5   3 4 2 1 1 7 0 6 7 9
8 2 1 4 8 0 8 6 5 1   3 2 8 2 3 0 6 6 4 7
0 9 3 8 4 4 6 0 9 5   5 0 5 8 2 2 3 1 7 2
5 3 5 9 4 0 8 1 2 8   4 8 1 1 1 7 4 5 0 2
8 4 1 0 2 7 0 1 9 3   8 5 2 1 1 0 5 5 5 9
6 4 4 6 2 2 9 4 8 9   5 4 9 3 0 3 8 1 9 6
ノイ   200   ノイ
```

READY