



# HOKKAIDO UNIVERSITY

Title	マイクロ・コンピュータ用BASICプログラム II
Author(s)	石田, 完; ISHIDA, Tamotsu
Citation	低温科学. 物理篇. 資料集, 42, 57-71
Issue Date	1984-03-15
Doc URL	<a href="https://hdl.handle.net/2115/18735">https://hdl.handle.net/2115/18735</a>
Type	departmental bulletin paper
File Information	42_p57-71.pdf



## マイクロ・コンピュータ用 BASIC プログラム II\*

石田 完

(低温科学研究所)

(昭和 58 年 12 月受理)

### 1. A-D 変換プログラム : Program 1, Program 1.1

野外観測等でデータをデータレコーダにアナログ記録し、これを A-D 変換してマイコンで解析処理する場合が多い。マイコンにはたいてい、内蔵される A-D 変換ボードが用意されているが、性能が貧弱で使用にたえない。吾々の所では、特に高速の A-D 変換が必要なので、A-D 変換器にデitel社の MDAS-16、小型モジュール・データアキュイジションシステムを用い、そのデジタル出力を、内蔵のデジタル I/O ボードを介して IF-800/20 マイコンに取りこんだ。そのプログラムを示す。

A-D 変換器 MDAS-16 のアナログ入力 は 16 チャンネル・シングルエンド、入力電圧は  $\pm 2.5$  V 以上で、変換性能は 12 bit、50 KHz サンプル時最大誤差はフルスケールの  $\pm 0.025\%$  となっている。吾々の場合、入力は 4 チャンネル、電圧範囲  $\pm 1$  V となっているので、A-D 変換器チャンネル番号 0 ~ 3 に 2.5 倍のオペアンプが 4 個つけてある。

プログラムをランさせると、まず文 2 で ADCON4.MEC なる機械語プログラム (Program 1.1) が E 000 番地から格納され、第 1 図に示すような NOTE が CRT に表示される。マイコンのメモリ容量の関係から、1 チャンネル当りのデータ数は表示の範囲内に制限される。サンプリング間隔は最小 0.5 ms、最大 682 ms となっている。最大間隔は本来無制限なのだが、数 ms というような早いサンプリングには、BASIC では A-D 変換器を制御できなかったのもので、変換の開始、チャンネルの指定、データの取りこみ、サンプル間隔等の制御は機械語で行った。そのプログラムを簡潔にするために上限値ができた。

次に第 1 図に示すようなチャンネル数、1 チャンネル当りのサンプリング数、サンプリング間隔に対するプロンプトがでる (文 30~52)。ここで、夫々の数値をキーインすればよいが、ミスキーインはリターンキーを押す前なら画面操作で修正できる。リターンキーを押すと Sure (Y/N) ? の表示が

\*北海道大学低温科学研究所業績 第 2636 号

```

NOTE: If 4 channels, Sampling No. must be (-512
3 - - - - - (-682
2 - - - - - (-1024
1 - - - - - (-2048
Interval of sampling must be .5+682 msec

```

```

A number of channel      ? 4
A number of sampling     ? 100
Interval of sampling (msec) ? 10

```

If ready , push RETURN key !

Conversion time= 1 sec

Print/Save/Retry/Change Parameters/End ? P

1	2043	2019	2038	2037
2	2014	2027	2037	2040
3	2064	2050	2039	2039
4	2007	2064	2030	2039
5	1991	2041	2036	2040
6	2047	2031	2039	2037
7	2017	2064	2036	2040
8	2032	1984	2040	2041
9	2015	2003	2036	2040
10	2000	2062	2039	2041
11	2033	2064	2030	2041
12	2042	1994	2030	2030
13	1991	2060	2039	2041
14	2000	2052	2039	2039
15	2044	2027	2030	2037
16	2057	2072	2039	2041
17	2031	1997	2037	2040

第1図 Program 1のCRT表示

である。良ければ Y ✓ で次に移る。N ✓ なら再入力となる。この辺の操作を文 500, 以下のサブルーチンで行っている。チャンネル数・サンプリング数・サンプリング間隔の 3 数値をキーインしおえると、If ready, push RETURN key ! の表示がでる(文150)。ここで A-D 変換器の接続、データレコーダの接続、テープスタート等を確認し、リターンキーを押すと、変換・データ格納が開始され、終了すると変換時間の表示がでる(秒単位)。第 1 図の例では 10 ms×100 なので表示は 1 sec となっている。その後、1 データ 12 bit を 2 番地に分けて格納したものを BASIC で扱えるように配列変数に格納しなおす(文 180~270)。データ数により多少の待ち時間があるが、第 1 図の Print/Save/Retry/change parameters/End ? の表示がでる。P/S/R/C/E の一字をキーインすることにより夫々の処理がなされる。P は結果の表示で第 1 図に示す如く左欄からサンプリング番号、第 1 ~ 第 4 チャンネルの数値である。12 bit なので、-1 V は 0, 0 V は 2048, +1 V は 4096 となる。S はフロッピーディスクへの格納で、各チャンネルのファイル名を聞いてくるので夫々をインプットすれば結果がファイルされる。R は今の各パラメータを変えずに A-D 変換しなおす場合で、文 140(I/O ボードの初期化)にもどる。C はパラメータを変更して A-D 変換しなおす場合で RUN 10 で最初にも

どる。RUN 10 としたのは配列変数の再定義をするからである。E はプログラムの終了で、機械語プログラムのために確保した E 000 番地以降を開放し、すべての領域を BASIC で使えるようにして終る (文 480)。

文 55 の式はサンプリング間隔の ms の数値を機械語のむだ時間発生ルーチン (Program 1.1 の E 02 F ~ E 038 番地) の繰返し回数に直す。標準信号発生器の正確な正弦波を適当な繰返し回数で A-D 変換し、波形の周期と繰返し回数との関係を求めて、式を決定した。文 70 ~ 130 は A-D 変換に必要な各パラメータを機械語プログラムに引渡す部分である。

IF-800 のデジタル I/O ボードは 8 bit パラレルの入出力ポートが 4 個ある。ここでは変換データの入力用に 2 個を入力ポート (C 0 番地と C 1 番地)、A-D 変換器の制御用に 1 個を出力ポート (C 2 番地) としている。C 0 番地ポートには 12 bit データの内、下位 8 bit が、C 1 番地ポートには上位 4 bit と、ビット位置 4 に変換中は 0、変換終了時に 1 がある。C 1 ポートを調べて、その内容が 16 以下なら変換中であり、16 か 16 以上になれば変換終了なので (Program 1.1 の E 011 ~ E 015 番地)、C 1 ポートと C 0 ポートの内容をメモリに格納する (Program 1.1 の E 018 ~ E 01 F 番地)。C 2 出力ポートはビット位置 0 ~ 3 がチャンネルアドレス指定 (Program 1.1 の E 00 E ~ E 00 F 番地)、ビット位置 4 が変換開始、ビット位置 5 が変換器のリセットである (Program 1.1 の E 00 A ~ E 00 C 番地)。

## 2. マイコン間のデータ転送プログラム : Program 2.1, 2.2

処理速度の向上と大きなメモリの活用をはかるために、IF-800/20 に蓄積されたプログラムや数値データを RS-232 C インターフェイスで PC-9801 に転送するプログラムである。Program 2.1 は IF-800 で用い、IF-800 から PC-9801 へデータを送る。データは ASCII 形式でファイルされたものでなければならない。プログラムと数値データは区別している (文 20)。転送の終了を PC-9801 に知らせるために、プログラムの場合は最後に "FFFFF" を、数値データの場合は 99999 を送る (文 90 と文 120)。いずれもその前後に CR のコードである CHR\$ (&HD) をつけなければならない。文 60 は RS-232 C インターフェイスのモード設定で、7 ビット偶数パリティ、ストップビット 1 のデータを 4800 ボーの速度で送ることを指定している。

Program 2.2 は PC-9801 側でデータを受取りファイルするプログラムである。データがプログラムの場合は 20 行、数値データの場合は 100 データ受取ると STOP するので (文 110, 文 220)、CONT をキーインして継続させる。内容が正しければファイル名をインプットすればファイルされる。内容がおかしければ単にリターンキーを押すことによりファイルされずに END となる。

両方のマイコンともプログラムを RUN させ、データがプログラムか数値かにより、T/N をキーインしておいてから、IF-800 側でファイル名をインプットする。

### 3. フロッピーディスクの内容をダンプするプログラム：Program 3.1, 3.2

誤操作により、ファイルの Directory や FAT (file allocation table) をこわしてファイル名では読めなくなる場合がある。その場合でも、あらかじめ Directory と FAT の内容をダンプしておけば、DSKO\$, DSKI\$関数を使ってたいい救済できる。このプログラムは8インチ両面倍密度フロッピーディスク用に、Rrogram 3.1はドライブ番号、サーフェイス番号、トラック番号、セクタ番号により、Program 3.2はファイル名により、その内容をダンプするものである。Directoryはサーフェイス0、トラック35、セクタ1~22に、FATはサーフェイス0、トラック35、セクタ24~26(通常24, 25, 26の内容は全く同じ)に書きこまれている。DirectoryとFATの内容の詳細い説明は計算機マニュアルにゆずり、ここでは一例を示す。第2図はProgram 3.1によりDirectoryとFATをダンプしたものでDirectoryのA0番地(以後、2桁の記号は16進数を表わす)の1行を見るとA8番地まではファイル名であり、WARK□□FILとなっている。AA番地の内容は42であるが、これはこのファイルの先頭クラスタ番号で、これを10進数になおした数(=66)をCとするとサーフェイス番号H、トラック番号Tは次式で計算できる。

```
drive= 1 surface= 0 track= 35 sector= 1

0 1 2 3 4 5 6 7 8 9 A B C D E F
0000 64 61 74 63 72 65 20 20 20 00 47 FF FF FF FF FF datcre .G.....
0010 66 72 61 63 74 61 6C 20 20 80 45 FF FF FF FF FF fractal _E.....
0020 66 72 61 63 74 61 32 20 20 80 48 FF FF FF FF FF fracta2 _H.....
0030 4F 20 37 20 20 20 44 41 54 00 44 FF FF FF FF FF 0 7 DAT.D.....
0040 66 72 61 63 74 61 73 75 62 00 49 FF FF FF FF FF fractasub.I.....
0050 4F 20 38 20 20 20 44 41 54 00 43 FF FF FF FF FF 0 8 DAT.C.....
0060 4F 20 39 20 20 20 44 41 54 00 4A FF FF FF FF FF 0 9 DAT.J.....
0070 73 79 73 67 65 6E 6E 38 38 80 40 FF FF FF FF FF sysgenn88_M.....
0080 73 65 74 69 6E 6E 38 38 80 40 FF FF FF FF FF setinfn88_@.....
0090 66 6F 72 6D 61 74 6E 38 38 80 41 FF FF FF FF FF formatn88_A.....
00A0 57 41 52 4B 20 20 46 49 4C 00 42 FF FF FF FF FF WARK FIL.B.....
00B0 74 65 73 74 20 20 64 61 74 00 3F FF FF FF FF FF test dat.?......
00C0 64 72 64 61 74 31 63 72 65 80 4E FF FF FF FF FF drdat1cre_N.....
00D0 74 65 73 74 31 20 64 61 74 00 3E FF FF FF FF FF test1 dat.>.....
00E0 64 72 64 61 74 32 63 72 65 80 4F FF FF FF FF FF drdat2cre_0.....
00F0 64 72 61 67 6F 6E 32 20 20 80 3D FF FF FF FF FF dragon2 _=.....

drive= 1 surface= 0 track= 35 sector= 24

0 1 2 3 4 5 6 7 8 9 A B C D E F
0000 FE FE FE FE FE FE FF FF FF FF FF FF FF FF FF FF .....
0010 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
0020 FF FF FF FF FF FF CD CD C1 C2 C2 CB C1 C3 C3 C2 .....
0030 C4 C2 D9 32 C2 D6 C5 CC D7 CC D1 C3 C4 C3 C3 C3 トツル2ツヨナフラムテテテテ
0040 C5 D0 33 DA C9 C3 FE C2 C2 4B 4C D9 C6 C3 C3 ナミ3レノテ.ツツツKLルニテテ
0050 C3 C4 C1 C9 D7 CA D0 C5 C4 C6 C6 C3 C4 C1 C2 C4 テトチノラハミナトニニテトチツト
0060 C2 C1 CA C1 C4 CF CD CD FF FF FF FF FF FF FF FF ツチハチトマハハ.....
0070 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
0080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
0090 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00A0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
00F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF .....
```

第2図 Directory と FAT の内容を Program 3.1 によりダンプした結果

$$H=C \text{ MOD } 2, T=C \div 2$$

FAT の 42 番地を見ると 33 であり、これはクラスタ番号 33 に続いていることを示している。更に FAT の 33 番地は 32 であり、クラスタ番号 32 に続く。32 番地の内容は D 9 であるが、2 桁目が C か D の場合は後続するクラスタがないことを示し、その内容から  $C0$  を引いたもの、 $D9 - C0 = 19$  は 32 クラスタで使用しているセクタ数 (10 進数の 25) を表わしている。結局、WARK.FIL と云うファイルは 3 クラスタを使用しているが、実際に使われているセクタ数は 1 クラスタ = 26 セクタなので  $26 \times 2 + 25 = 77$  セクタと云うことになる。以上のように大きなファイルは必ずしも連続したクラスタに納められていないので FAT がこわれると後続のクラスタを探しようがない。なお FAT の内容で FE はシステムで使用していることを示し、FF は空きで使用可能であることを示している。

Program 3.2 ではファイル名を与えることにより文 60 で先頭のクラスタ番号  $C(0)$  を求め、文 130, 140 で FAT を調べ、後続クラスタがあれば文 150~180 でその番号を  $C(1)$ ,  $C(2)$ , ... に求めている。このプログラムは N<sub>88</sub>-BASIC で書かれており、ダンプした結果を CRT に表示するか、プリンタに印字させるかを選択できる (文 70)。また文 110~140 はファイル・バッファの先頭番地を求める N<sub>88</sub>-BASIC の常套手段である。

#### 4. ファイルの効率よい使用法

##### 4.1 データの構造

例として電話帳を考える。一つの電話番号について、名前・職業・住所がある。これらを項目 (NK) とよび、項目番号で表わす。

NK = 1 : 電話番号

NK = 2 : 名 前

NK = 3 : 職 業

NK = 4 : 住 所

今の場合、項目数は 4 であるが、一般に項目数を KK で表わす。一組の項目に対しデータ番号を与える。それを ND とすると、電話番号が 100 個あれば、 $ND = 1, 2, \dots, 100$  となる。これをファイルするのであるが、ファイルの I/O は 1 レコード (256 バイト) 単位である。今の場合、1 レコードに 1 データだけでは余裕がありすぎるので 1 レコードに 3 データ宛書きこむことにする。一般に 1 データの長さは可変で 1 レコードに納まるデータ数もかわるから、これを KD とする。今の場合  $KD = 3$  である。

ここでレコード番号を NR, 入力した項目の続き番号を N としておく。今の場合、最大の  $N = (\text{データ番号 } ND) \times (\text{項目数 } KK) = 100 \times 4 = 400$  である。普通この NR と N は必要でない。吾々は ND と NK を与えて、ファイルを書きかえたり、追加したりする時、計算機内で NR や N を計算させればよい。次表にレコード番号・データ番号・項目番号・続き番号の関係を示す。

項目数 KK と 1 レコード内のデータ数 KD は既知である。

レコード番号 NR	1												2											
データ番号 ND	1				2				3				4				5				6			
項目番号 NK	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
続き番号 N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

データ番号 ND からレコード番号 NR を知るには

$$NR = (ND + KD - 1) \div KK \quad (1)$$

データ番号 ND と項目番号 NK から一連番号 N を知るには

$$N = (ND - 1) * KK + NK \quad (2)$$

一連番号 N からデータ番号 ND を知るには

$$ND = (N \div KK + 1) * KK \quad (3)$$

となる。

#### 4.2 データの入力：Program 4.1

各項目の文字数を以下のように定める。

電話番号：12 文字

名 前：23 文字

1 データ 85 文字

職 業：20 文字

住 所：30 文字

1 データは 85 文字となるから、1 レコードに 3 データを入れると、 $85 \times 3 = 255$  バイトとなり 1 バイト残る。これはレコードの先頭の目印とする。ランダム・ファイルとして入力すれば任意の項目を読みだす時、便利であるが、シーケンシャル・ファイルとして入力しても、ランダム・ファイルと同様に扱うことができる。Program 4.1 はシーケンシャル・ファイルとして入力している。

文 1020 で new file か old file かをきき、OUTPUT か APPEND かでファイルをオープンする。FIELD 文で行う文字型変数の領域割当てを文 1070~1100 で行ない、文 1110 の PRINT USING 文で書きこむ。文 1110 で最初のデータ番号をきき (new file なら当然 1)、これを 3 で割った余りが 1 ならレコードの先頭だから目印の CHR\$(0) を書きこむ。

この例ではすべて文字型のデータを入力しているが、例えば気象要素のデータで

日	時	気 温	気 圧	風 向	風 速	雲 量
83, 12/25,	15:30	-5.6	1012	15	3.4	2

の如きものであれば、日時は 14 バイトの文字、他はすべて数値である。気温・気圧・風速は単精度実数とすれば MKS\$関数により 4 バイトの文字として、風向は 16 方位、雲量は 10 段階とすれば MKI\$関数により 2 バイト文字として書きこむ。この例では 1 データ 30 バイトとなり、1 レコードに 8 データ書きこむことができる。

### 4.3 データファイルの読みだしと訂正：Program 4.2

Program 4.1 で作ったデータファイルの一部訂正を行うプログラムである。データ番号 ND の項目番号 NK の内容を読みだす。例えば 7 番データの住所であれば、ND=7, NK=4 である。文 2080, 2130 で夫々をインプットする。文 2050 の FIELD 文は 1 レコードに先頭 1 バイトの目印, 85 バイト宛 3 データ分がはいっていることを示す。文 2060, 2070 の FIELD 文はファイルをオープンしない作業用バッファで 85 バイトの 1 データを 4 項目に分けて取りだすものである。指定された項目は文 2140 により表示される。訂正する時は → 印がでたら文字データをキーインする。単精度実数値データなら、この部分は INPUT C : C\$=MK\$(C) としなければならない。訂正する必要がない時は → 印がでた後、単にリターンキーを押せばよい。そのデータ番号で、他の項目を読みだす必要がなければ項目番号 0 を、また他のデータ番号の内容も読みださないのなら、データ番号 0 をキーインすればプログラムは終了する。

以上二つのプログラムは汎用でないが、例をとって基本的な所を示した。夫々のデータ構造により一部を書きかえれば夫々の場合に応用しうる。

### 4.4 非 ASCII で格納した数値データの読みだし：Subprogram 4.3

N 個の数値データを

```
100 OPEN F$ FOR OUTPUT AS #1
110 FOR I=1 TO N:PRINT #1, A(I);:NEXT
```

で格納するよりも

```
110 FOR I=1 TO N:PRINT #1, MK$(A(I));:NEXT
```

と 4 バイト文字として格納すれば、領域が約半分ですむ。第 3 図は 50 個の乱数を PRINT #1, RND; でファイルしたものを Program 3.2 によりダンプした結果である。第 4 図は同じ 50 個の乱数を PRINT #1, MK\$(RND); でファイルした内容である。これを読みだすプログラムを Subprogram 4.3 として示す。

ファイルをランダム・ファイルとして扱い、1 レコード分を 128 バイト宛 A\$ と B\$ に分けて読みだす (文 1050, 1070)。これを文 1010 の FNC 関数で更に 4 バイト宛に分割して、CVS 関数により数値に直し、A(1), ……A(N) に取りだす。A\$ と B\$ のスイッチングは K が 32 増す毎に行い、K が 64 増す毎にレコード番号を 1 宛増してゆく。

## 5. ソーティング：Program 5

ソーティングのプログラムは非常に沢山あるが、ここに示すものは、先頭文字がアルファベットに限られた文字列を辞書式に配列し直すもので、かなり高速である。

まず、先頭の文字が A なら 0 番の箱 (B\$(0,JJ), B なら 1 番の箱 (B\$(1,JJ)) に区分けして (文 1000~1030), 次に各箱の中の文字列をソートする (文 1060~1130)。各箱にはいった文字列の個数は K(J) の配列にある。J は先頭文字の番号で、J=0 は A, J=1 は B を表わす。K(J)=0 なら J 番

```

10 OPEN 'ascii.dat' FOR OUTPUT AS #1
20 FOR I=1 TO 50 :PRINT #1,RND; :NEXT
30 END

```

File : ascii.dat

drive= 1 ,surface= 1 ,track= 50 ,sector= 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	20	2E	30	32	38	32	34	36	34	20	20	2E	36	31	30	37	.0282464 .6107
0010	36	39	20	20	2E	39	36	34	37	36	39	20	20	2E	39	37	69 .964769 .97
0020	32	30	38	31	20	20	2E	33	38	39	35	34	32	20	20	2E	2081 .389542 .
0030	38	34	38	39	38	37	20	20	2E	38	35	37	32	35	33	20	848987 .857253
0040	20	2E	37	39	36	31	37	36	20	20	2E	39	32	32	35	39	.796176 .92259
0050	20	20	2E	33	36	38	33	33	33	20	20	2E	31	34	30	32	.368333 .1402
0060	34	31	20	20	2E	31	32	30	31	34	38	20	20	2E	30	36	41 .120148 .06
0070	34	38	39	31	36	20	20	2E	36	30	36	33	30	37	20	20	48916 .606307
0080	2E	32	35	31	32	33	31	20	20	2E	33	38	31	34	39	38	.251231 .381498
0090	20	20	2E	32	35	30	30	34	20	20	2E	34	39	32	35	39	.25004 .49259
00A0	37	20	2E	31	31	39	39	31	32	20	20	2E	35	30	39	39	7 .119912 .509
00B0	39	31	34	20	20	2E	34	31	39	34	34	31	20	20	2E	34	914 .419441 .4
00C0	38	30	33	32	36	20	20	2E	31	39	35	35	30	31	20	20	80326 .195501
00D0	2E	34	35	34	36	31	34	20	20	2E	30	31	30	35	39	34	.454614 .010594
00E0	34	20	20	2E	39	39	35	32	37	38	20	20	2E	34	31	35	4 .995278 .415
00F0	35	30	31	20	20	2E	31	35	33	30	39	38	20	20	2E	39	501 .153098 .9

Residue: 1 sct

drive= 1 ,surface= 1 ,track= 50 ,sector= 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0100	36	31	30	30	31	20	20	2E	39	37	38	38	35	37	20	20	61001 .978857
0110	2E	32	30	39	35	39	36	20	20	2E	30	33	35	30	35	34	.209596 .035054
0120	33	20	20	2E	37	31	32	30	36	37	20	20	2E	33	37	32	3 .712067 .372
0130	34	37	31	20	20	2E	30	32	33	31	30	31	32	20	20	2E	471 .0231012 .
0140	35	34	35	37	39	34	20	20	2E	36	39	37	33	38	36	20	545794 .697386
0150	20	2E	31	30	35	38	30	36	20	20	2E	37	38	31	37	39	.105806 .78179
0160	37	20	20	2E	31	31	31	31	20	20	2E	38	34	36	36	34	7 .1111 .84664
0170	35	20	20	2E	31	32	30	32	36	39	20	20	2E	39	33	38	5 .120269 .938
0180	38	30	37	20	20	2E	31	38	30	31	38	39	20	20	2E	31	807 .180189 .1
0190	30	39	30	36	34	20	20	2E	38	35	33	33	36	20	20	2E	09064 .85336 .
01A0	39	31	39	39	31	35	20	20	2E	36	39	30	35	36	33	20	919915 .690563
01B0	20	2E	34	31	38	32	33	35	20	20	2E	37	33	38	36	37	.418235 .73867
01C0	33	20	1A	00	00	00	00	00	00	00	00	00	00	00	00	00	3 .....
01D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
01E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
01F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Residue: 0 sct

第 3 図

```

10 OPEN 'nonasc.dat' FOR OUTPUT AS #1
20 FOR I=1 TO 50 :PRINT #1,MKS$(RND); :NEXT
30 END

```

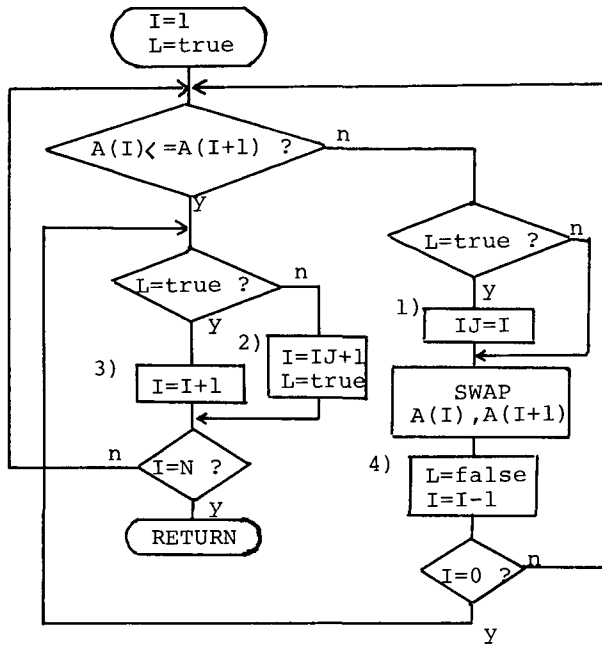
File : nonasc.dat

drive= 1 ,surface= 0 ,track= 51 ,sector= 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	00	65	67	7B	5F	5B	1C	80	16	FB	76	80	4D	DA	78	80	.eg( [ _ . . . v Mlvx _
0010	08	72	47	7F	38	57	59	80	F2	74	5B	80	29	D2	48	80	.rG ;WY_#t[_)xK_
0020	E0	2E	6C	80	2E	96	3C	7F	38	9B	0F	7E	28	10	76	7D	=.1_  K 8J.(.v)
0030	E0	E5	04	7D	F3	36	1B	80	54	A1	00	7F	C2	53	43	7F	=.)#6._T_.. ySC
0040	30	05	00	7F	9E	35	7C	7F	30	94	75	7D	8D	89	02	80	0.. 5; 0u)7I _
0050	E8	C0	56	7F	56	ED	75	7F	88	31	48	7E	32	C3	68	7F	4V VQ_ l1H^27h
0060	00	94	2D	7A	87	CA	7E	80	7C	BC	54	7F	D4	C5	1C	7E	.-z#~_i3T ++.~
0070	2C	04	76	80	63	96	7A	80	68	A0	56	7E	10	95	0F	7C	..vclz_h V_-.!
0080	08	4A	36	80	7E	84	3E	7F	C0	3E	3D	7B	2D	89	0B	80	.J6_~I> 7>=(~7_
0090	E4	87	32	80	D8	B0	58	7D	D2	23	48	80	48	88	63	7D	2J-X)#H_HI c)
00A0	C0	BD	58	80	88	4F	76	7D	AE	55	70	80	94	83	38	7E	7SX_70v)3Up_~8~
00B0	ED	5C	5F	7D	D3	75	5A	80	8A	7F	6B	80	C1	C8	30	80	=>)FuZ_ k_7*0_
00C0	F0	22	56	7F	AF	19	3D	80	1A	00	00	00	00	00	00	00	X^V 7_=_.....
00D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Residue: 0 sct

第 4 図



```

1060 I=1 :L=-1
1070 IF A(I)<=A(I+1) THEN 1110
1080 IF L THEN IJ=I
1090 SWAP A(I), A(I+1) :L=0 :I=I-1
1100 IF I<>0 THEN 1070
1110 IF L THEN I=I+1 ELSE I=IJ+1 :L=-1
1120 IF I<>N THEN 1070
1130 RETURN
    
```

第5図 ソーティング・サブルーチンプログラムとフローチャート

- 1) もどり始める時の位置を記憶
- 2) もどり始めた位置の右隣りへ jump
- 3) 一つ右隣りへ
- 4) Lを false にして、一つ左隣りへ

の文字はないことになるし、 $K(J) = 1$  なら1文字列しかないのでソートする必要がなく、文 1050 で次の番号の箱のソートに移る。J 番の箱に2文字列以上ある場合は、文 1060～1130 により第5図に示すフローチャートの方法でその箱内をソートする。

Program 5 は文字列のソーティングであるが、N 個の数列  $A(I)$  を小さい方から順に並べかえる場合なら文 1070 の比較文を  $A(I) <= A(I+1)$ 、文 1090 の交換文を  $SWAP A(I), A(I+1)$ 、文 1120 の比較文を  $I <> N$  とし、サブルーチン化するなら文 1130 を RETURN 文とすればよい。第5図のフローチャートとサブルーチンプログラムはこの場合を示している。

元の数列  $A(1), \dots, A(N)$  が左から右へ並んでいるとする。右隣りが常に大ならフローチャートの左側の列を真すぐ下って RETURN となるが、右隣りが小ならチャートの右側の列に移り、数列を左へもどって比較をしてゆくことになる。この時、もどり初めた数列の位置を記憶しておいて (IJ)、次には IJ 番の右隣りから初め、すでに比較しおえた所は二度くりかえさないようにしている。これでも数列が長い場合は、行きつ、もどりの回数が増え時間を要することとなる。長い数列の

場合は数列を幾つかに分割し、分割したグループ内を夫々ソートし、次にソートされたグループ同志をソートする方法をとる。その時のサブルーチンとして使える。

Program 5 は頭文字がアルファベットの 2 ないし 6 文字のランダムな文字列を発生し (文 60 とサブルーチン 3000~3080)、これを並べかえるもので、50 個の文字列の場合の実行例をプログラムの後に示した。PC-9801 でソーティング実行時間は約 1.5 秒であった。

### Program 1

```

1  '*** A-D CONVERSION ***
2  CLEAR ,%HDFFF : BLOAD "ADCON4.MEC"
10 CLS
11 PRINT "
12 PRINT "      | NOTE: If 4 channels, Sampling No. must be <=512 |"
13 PRINT "      |           3   - - - - - - - - - - - - - - - - |"
14 PRINT "      |           2   - - - - - - - - - - - - - - - - |"
15 PRINT "      |           1   - - - - - - - - - - - - - - - - |"
16 PRINT "      |           Interval of sampling must be .5→682 msec |"
17 PRINT "
18 PRINT
20 DEFINT A-D, I-M : DEF FNC(A,B)=(A-16)*256+B
30 PRINT "A number of channel      ?"
35 LN=8:GOSUB 500:L=VAL(I$)
40 PRINT "A number of sampling    ?"
45 LN=9:GOSUB 500:K=VAL(I$)
50 PRINT "Interval of sampling (msec) ?"
52 LN=10:GOSUB 500:T=VAL(I$)
55 T=95.89*T-46.46
56 M1=INT(T/256) : M2=INT(T-M1*256+.5)
60 ON L GOTO 65,66,67,68
65 DIM A(K) : GOTO 70
66 DIM A(K),B(K) : GOTO 70
67 DIM A(K),B(K),C(K) : GOTO 70
68 DIM A(K),B(K),C(K),D(K)
70 X=&HE000 : Y=&HE040 : X1=&HE001 : X2=&HE002
80 K1=K*256 : K2=K-K1*256
120 POKE X2,K1 : POKE X1,K2
130 POKE &HE007,L : POKE &HE031,M1 : POKE &HE030,M2
140 OUT 194,0 : J=0
150 PRINT :PRINT "If ready , Push RETURN key !"
151 I=CSRLIN-1
152 I$=INKEY$
153 IF I$="" OR I$<>CHR$(13) THEN 154 ELSE 157
154 LOCATE 16,I:COLOR 0,1:PRINT "RETURN":PAUSE .5
155 LOCATE 16,I:COLOR 1,0:PRINT "RETURN":PAUSE .5
156 GOTO 152
157 CLS I,1
160 BEEP : T1=TIME : CALL X : T2=TIME : BEEP
170 LOCATE 0,I : PRINT "Conversion time=";T2-T1;"sec" : PRINT
180 FOR I=1 TO K
190 A1=PEEK(Y+J) : A2=PEEK(Y+J+1) : IF L=1 THEN 230
200 B1=PEEK(Y+J+2) : B2=PEEK(Y+J+3) : IF L=2 THEN 230
210 C1=PEEK(Y+J+4) : C2=PEEK(Y+J+5) : IF L=3 THEN 230
220 D1=PEEK(Y+J+6) : D2=PEEK(Y+J+7)
230 A(I)=FNC(A1,A2) : IF L=1 THEN 270
240 B(I)=FNC(B1,B2) : IF L=2 THEN 270
250 C(I)=FNC(C1,C2) : IF L=3 THEN 270
260 D(I)=FNC(D1,D2)
270 J=J+L*2 : NEXT I
280 INPUT "Print/Save/Retry/Change Parameters/End ";P$ : PRINT
290 IF P$="P" THEN 300 ELSE IF P$="S" THEN 360 ELSE IF P$="R" THEN 140 ELSE IF P$="C" THEN RUN 10 ELSE IF P$="E" THEN 480 ELSE 280
300 FOR I=1 TO K : ON L GOTO 310,320,330,340
310 PRINT I,A(I) : GOTO 350
320 PRINT I,A(I),B(I) : GOTO 350
330 PRINT I,A(I),B(I),C(I) : GOTO 350
340 PRINT I,A(I),B(I),C(I),D(I)
350 NEXT I : PRINT : GOTO 280
360 INPUT "File name 1 ";F1$ : IF L=1 THEN 400
370 INPUT "File name 2 ";F2$ : IF L=2 THEN 400

```

## Program 1の続き

```

380 INPUT "File name 3 ";F3$ : IF L=3 THEN 400
390 INPUT "File name 4 ";F4$
400 OPEN F1$ FOR OUTPUT AS #1
410 FOR I=1 TO K : PRINT #1,A(I); : NEXT : CLOSE #1 : IF L=1 THEN 280
420 OPEN F2$ FOR OUTPUT AS #1
430 FOR I=1 TO K : PRINT #1,B(I); : NEXT : CLOSE #1 : IF L=2 THEN 280
440 OPEN F3$ FOR OUTPUT AS #1
450 FOR I=1 TO K : PRINT #1,C(I); : NEXT : CLOSE #1 : IF L=3 THEN 280
460 OPEN F4$ FOR OUTPUT AS #1
470 FOR I=1 TO K : PRINT #1,D(I); : NEXT : CLOSE #1 : GOTO 280
480 CLEAR ,&HFFFF : END
500 '*** SUB ***
510 I$="" :I=0
520 J$=INKEY$ :IF J$="" THEN 600 ELSE IF J$=CHR$(13) THEN 700
530 I$=I$+J$:LOCATE 30+I,LN:PRINT J$;:I=I+1:GOTO 520
600 LOCATE 7,1:COLOR 0,1:PRINT "NOTE":PAUSE .5
610 LOCATE 7,1:COLOR 1,0:PRINT "NOTE":PAUSE .5 :GOTO 520
700 LOCATE 40,LN:COLOR 0,1:INPUT "Sure (Y/N) ";Y$
710 IF Y$="Y" THEN COLOR 1,0:RETURN
720 COLOR 1,0:LOCATE 30,LN:PRINT SPC(30):GOTO 510

```

## Program 1.1

```

E000 01 F4 01 LD BC,01 F4
E003 21 40 E0 LD HL,E0 40
E006 16 01 LD D,01
E008 1E 10 LD E,10
E00A 3E 20 LD A,20
E00C 03 C2 OUT (C2 ),A
E00E 7B LD A,E
E00F 03 C2 OUT (C2 ),A
E011 0B C1 IN A,(C1 )
E013 E6 10 AND 10
E015 CA 11 E0 JP Z,E0 11
E018 0B C1 IN A,(C1 )
E01A 77 LD (HL),A
E01B 23 INC HL
E01C 0B C0 IN A,(C0 )
E01E 77 LD (HL),A
E01F 23 INC HL
E020 15 DEC D
E021 CA 20 E0 JP Z,E0 20
E024 1C INC E
E025 C3 0A E0 JP E0 0A
E028 0B DEC BC
E029 7B LD A,B
E02A 01 OR C
E02B C2 2F E0 JP NZ,E0 2F
E02E C9 RET
E02F 11 00 02 LD DE,02 00
E032 1B DEC DE
E033 7A LD A,D
E034 B3 OR E
E035 C2 32 E0 JP NZ,E0 32
E038 C3 06 E0 JP E0 06

```

## Program 2.1

```

10 '*** Send data to PC-9801 ***
15 D$=CHR$(&HD)
20 INPUT 'Text or Numerical data (T/N) ';C$
30 IF NOT(C$='T' OR C$='N') THEN 20
40 INPUT 'File name ';F$
50 OPEN F$ FOR INPUT AS #2
60 OPEN 'COM1:7E71' AS #1
70 IF C$='T' THEN 80 ELSE 110
80 WHILE NOT EOF(2):LINE INPUT #2,A$:PRINT #1,A$
90 WEND:PRINT #1,D$;'FFFFF';D$
100 CLOSE #1,#2:END
110 WHILE NOT EOF(2):INPUT #2,X:PRINT #1,X;
120 WEND:PRINT #1,D$;99999!;D$
130 CLOSE #1,#2:END

```

## Program 2.2

```

10 '*** Receive data from IF-800 ***
20 INPUT 'Text or Numerical data (T/N) ';C$
30 IF NOT(C$='T' OR C$='N') THEN 20
40 PRINT 'Ok !'
50 OPEN 'COM:E71XS' FOR INPUT AS #1
60 IF C$='T' THEN 70 ELSE 180
70 DIM A$(1000):I=0
80 LINE INPUT #1,A$(I):IF A$(I)='FFFFF' THEN 100
90 I=I+1 :GOTO 80
100 PRINT A$(0)
110 FOR J=1 TO I-1 :PRINT A$(J) :IF J MOD 20=0 THEN STOP
120 NEXT J
130 INPUT 'File name ';F$
140 IF F$="" THEN CLOSE #1 :END
150 OPEN F$ FOR OUTPUT AS #2
160 FOR J=0 TO I-1:PRINT #2,A$(J):NEXT
170 CLOSE #1,#2:END
180 DIM X(10000):I=0
190 INPUT #1,X(I):IF X(I)=99999! THEN 210
200 I=I+1:GOTO 190
210 PRINT X(0),:FOR J=1 TO I-1:PRINT X(J),
220 IF J MOD 100 THEN STOP ELSE NEXT
230 PRINT :INPUT 'File name ';F$
240 IF F$="" THEN CLOSE #1:END
250 OPEN F$ FOR OUTPUT AS #2
260 FOR J=0 TO I-1:PRINT #2,X(J);:NEXT
270 CLOSE #1,#2:END

```

## Program 3.1

```

5 DEFINT A-Z
10 FIELD #0,255 AS A$,1 AS B$ :S=VARPTR(#0,1):P=VARPTR(#0)
20 CLS:INPUT"Drive (1/2) ";UD
30 INPUT"Surface (0/1) ";UH
40 INPUT"Track (0-76) ";UT
50 INPUT"Sector (1-26) ";US
60 CLS:PRINT"drive=";UD"surface=";UH"track=";UT"sector=";US:PRINT
70 PRINT"      0 1 2 3 4 5 6 7 8 9 A B C D E F"
80 D$=DSKI$(UD,UH,UT,US)
82 DEF SEG=S
83 K1=PEEK(P+32):K2=PEEK(P+33):UA=K1+K2*256
90 FOR DI=UA TO UA+255 STEP 16
100 GOSUB 300
110 PRINT RIGHT$("000"+HEX$(DI-UA),4)+" "+A$+" "+B$
120 NEXT DI
130 PRINT:PRINT:PRINT"Hit SPACE-BAR for the next sector,or RET-KEY for END !"
140 Z$=INKEY$:IF Z$="" THEN 140
150 IF Z$="" THEN 160 ELSE END
160 US=US+1:IF US>26 THEN END ELSE 60
300 A$="" :B$=""
310 FOR J=DI TO DI+15 :PJ=PEEK(J)
320 A$=A$+RIGHT$("0"+HEX$(PJ),2)+" "
330 IF PJ<&H20 OR PJ>&HF7 THEN B$=B$+"." ELSE B$=B$+CHR$(PJ)
340 NEXT J
350 RETURN

```

## Program 3.2

```

10 ***** File dump *****
20 DEFINT A-Z
30 INPUT"File name ";F$:F1$=LEFT$(F$,1)
40 IF F1$='2' THEN UD=2 ELSE UD=1
50 OPEN F$ FOR INPUT AS #1
60 Q=FPOS(1):C(0)=Q%26 :CLOSE #1
70 PRINT:INPUT "Print (Y/N) ";W$
80 IF W$='Y' THEN 90 ELSE 100
90 OPEN "LPT: " FOR OUTPUT AS #1 :GOTO 110
100 OPEN "SCRN:" FOR OUTPUT AS #1
110 FIELD #0,255 AS A$,1 AS B$ :S=VARPTR(#0,1):P=VARPTR(#0)
120 DEF SEG=S
130 D$=DSKI$(UD,0,35,24)
140 K1=PEEK(P+32):K2=PEEK(P+33):UA=K1+K2*256
150 I=0
160 I1=I+1:C(I1)=PEEK(UA+C(I))
170 IF C(I1)>191 AND C(I1)<219 THEN SE=C(I1)-192 :GOTO 190
180 I=I1 :GOTO 160
190 TS=I*26+SE :PP=0
195 CLS: PRINT #1,"File : ";F$ :PRINT #1,""
200 FOR K=0 TO I :UH=C(K) MOD 2 :UT=C(K)%2
210 IF K=I THEN KE=SE ELSE KE=26
220 FOR US=1 TO KE :AD$=RIGHT$("0"+HEX$(PP),2)
230 PRINT #1,"drive=";UD",surface=";UH",track=";UT",sector=";US:PRINT #1,""
240 PRINT #1,"      0 1 2 3 4 5 6 7 8 9 A B C D E F"
250 D$=DSKI$(UD,UH,UT,US)
260 K1=PEEK(P+32):K2=PEEK(P+33):UA=K1+K2*256
270 FOR DI=UA TO UA+255 STEP 16
280 GOSUB 370
290 PRINT #1,AD$+RIGHT$("0"+HEX$(DI-UA),2)+" "+A$+" "+B$
300 NEXT DI :PRINT #1,"" :TS=TS-1:PP=PP+1
310 PRINT #1,TAB(57):"Residue:";TS;"sct"
320 PRINT #1,"":PRINT #1,""
330 PRINT"Hit SPACE-BAR for the next sector,or RET-KEY for END !"
340 Z$=INKEY$:IF Z$="" THEN 340
350 IF Z$="" THEN 360 ELSE END
360 NEXT US,K :END
370 A$="" :B$=""
380 FOR J=DI TO DI+15 :PJ=PEEK(J)
390 A$=A$+RIGHT$("0"+HEX$(PJ),2)+" "
400 IF PJ<&H20 OR PJ>&HF7 THEN B$=B$+"." ELSE B$=B$+CHR$(PJ)
410 NEXT J
420 RETURN

```

## Program 4.1

```

1000 '*** Random filing of data ***
1010 INPUT "A name of file ";FIL$
1020 INPUT "New or old file (N/O) ";Z$
1030 IF Z$="N" THEN 1050 ELSE IF Z$="O" THEN 1040 ELSE 1020
1040 OPEN FIL$ FOR APPEND AS #1 :GOTO 1060
1050 OPEN FIL$ FOR OUTPUT AS #1
1060 T$=""
1070 F$(1)=T$+SPACE$(10)+T$
1080 F$(2)=T$+SPACE$(21)+T$
1090 F$(3)=T$+SPACE$(18)+T$
1100 F$(4)=T$+SPACE$(28)+T$
1110 INPUT "A number of the first data ";K
1120 IF (K MOD 3)<>1 THEN 1140
1130 PRINT #1,CHR$(0);
1140 PRINT " No. ";K
1150 LINE INPUT "Tel :",A$(1) :IF A$(1)="000" THEN END
1160 LINE INPUT "Nam :",A$(2)
1170 LINE INPUT "Job :",A$(3)
1180 LINE INPUT "Add :",A$(4)
1190 FOR I=1 TO 4 :PRINT #1,USING F$(I);A$(I); :NEXT
1200 K=K+1 :IF (K MOD 3)=1 THEN 1130 ELSE 1140

```

## Program 4.2

```

2000 '*** File correction ***
2010 INPUT "A name of file ";FIL$
2020 DEFINT I,J,K,N
2030 OPEN FIL$ AS #1
2040 KD=3 :KK=4
2050 FIELD #1,1 AS P$,85 AS A$(1),85 AS A$(2),85 AS A$(3)
2060 FIELD #2,85 AS A$
2070 FIELD #2,12 AS B$(1),23 AS B$(2),20 AS B$(3),30 AS B$(4)
2080 INPUT "A rewriting data number ";ND :IF ND=0 THEN END
2090 I=ND MOD KD :IF I=0 THEN I=KD
2100 NR=(ND+KD-1)/KD
2110 GET #1,NR :IF P$<>CHR$(0) THEN PRINT "Filing error !":END
2120 LSET A$=A$(I)
2130 INPUT "A rewriting item number ";NK :IF NK=0 THEN 2170
2140 PRINT B$(NK);"->";:LINE INPUT C$
2150 IF C$="" THEN C$=B$(NK)
2160 LSET B$(NK)=C$ :GOTO 2130
2170 LSET A$(I)=A$
2180 PUT #1,NR :GOTO 2080

```

## Subprogram 4.3

```

1000 '*** Subprogram for input of non-ASCII data file ***
1010 DEFINT I-N: DEF FNC(A$)=CVS(MID$(A$,J*4-3,4))
1020 INPUT "File name ";F$
1030 INPUT "A number of data ";N
1040 OPEN F$ AS #1 : DIM A(N)
1050 FIELD #1,128 AS A$,128 AS B$
1060 I=1 : KS=0
1070 GET #1,I
1080 FOR J=1 TO 32 :K=J+KS :IF K>N THEN RETURN
1090 A(K)=FNC(A$) :NEXT
1100 FOR J=1 TO 32 :K=J+KS+32 :IF K>N THEN RETURN
1110 A(K)=FNC(B$) :NEXT :IF K=N THEN RETURN
1120 KS=KS+64 :I=I+1 :GOTO 1070

```

Program 5

```

10 ^*** Sort of character array ***
20 DEFINT I-N
30 INPUT "Number of data ";N
40 DIM A$(N),B$(25,30),K(25)
50 RANDOMIZE
60 FOR I=1 TO N :GOSUB 3000 :A$(I)=W$ :NEXT :PRINT"Soruce :":PRINT
70 FOR I=1 TO N :PRINT A$(I)+SPACE$(8-LEN(A$(I)));:NEXT:PRINT:PRINT
1000 FOR I=1 TO N
1010 J=ASC(A$(I))-65 :K(J)=K(J)+1 :JJ=K(J)
1020 B$(J,JJ)=A$(I)
1030 NEXT
1040 FOR J=0 TO 25 :JJ=K(J)
1050 IF JJ<=1 THEN 1130
1060 I=1 :L=-1
1070 IF RIGHT$(B$(J,I),LEN(B$(J,I))-1)<=RIGHT$(B$(J,I+1),LEN(B$(J,I+1))-1) THEN
1110
1080 IF L THEN IJ=I
1090 SWAP B$(J,I),B$(J,I+1) :L=0 :I=I-1
1100 IF I<>0 THEN 1070
1110 IF L THEN I=I+1 ELSE I=IJ+1 :L=-1
1120 IF I<>JJ THEN 1070
1130 NEXT
2000 PRINT"Result :":PRINT:FOR J=0 TO 25 :IF K(J)=0 THEN 2020
2010 FOR I=1 TO K(J):PRINT B$(J,I)+SPACE$(8-LEN(B$(J,I)));:NEXT:PRINT
2020 NEXT
2030 END
3000 IZ=INT(RND*100) :IF IZ<65 OR IZ>90 THEN 3000
3010 W$=CHR$(IZ)
3020 IR=INT(RND*10) :IF IR<1 OR IR>5 THEN 3020
3030 FOR J=1 TO IR
3040 IZ(J)=INT(RND*100)
3050 IF IZ(J)>47 AND IZ(J)<58 THEN 3070
3060 IF IZ(J)>64 AND IZ(J)<91 THEN 3070 ELSE 3040
3070 W$=W$+CHR$(IZ(J))
3080 NEXT :RETURN
    
```

Soruce :

C00C2	LP6WY	DW	CH	WG6E2	TTNYHU	Q7	J6X	NN9K4R	SE
CIF0QD	D7	X9M	YT	LT	NO	OC4	BNMG	GOZ8I	PJZ
ZSV2	M8G	SYWF01	VEH	O1	MOGA1	POTLZN	YSM	A8	RW6
SU4C	CEWAS	NA	LE	WYB	JD6B	A2C3IG	KJ5	QU50Y3	WNL
GC	Z0MXN	GSRE	G0K4U	OYGJR	NG	QBGNZ8	H1NKF3	MEBG3Z	NV

Result :

A2C3IG	A8								
BNMG									
C00C2	CEWAS	CH	CIF0QD						
D7	DW								
GOZ8I	GC	G0K4U	GSRE						
H1NKF3									
J6X	JD6B								
KJ5									
LE	LP6WY	LT							
M8G	MEBG3Z	MOGA1							
NO	NA	NG	NN9K4R	NV					
O1	OC4	OYGJR							
PJZ	POTLZN								
Q7	QBGNZ8	QU50Y3							
RW6									
SE	SU4C	SYWF01							
TTNYHU									
VEH									
WG6E2	WNL	WYB							
X9M									
YSM	YT								
Z0MXN	ZSV2								