ND8080 応用プログラム集

(有)中日電工

目次

〒463-0067　名古屋市守山区守山2－8－14
パレス守山305
有限会社中日電工
℡052-791-6254　Fax052-791-1391
E-mail　thisida@alles.or.jp
Homepage　http://www.tyunitidenko.x0.com/

2016．4．29　Rev.　1．0

第1部　ゲームプログラム

ｉ）はじめに

本編はND8080のためのゲームプログラム集です。

2003年にND80K用の応用プログラム集として編集したものを、2010年にND80ZⅢのために書き直しました。その後2012年にND80Z3．5用に、内容はそのまま語句のみND80ZⅢをND80Z3．5に書き換えました。

今回（2016年4月）、もとはZ80CPU用として作成したプログラムを8080CPU用に書き改めました。

数日間でZ80プログラムを8080用に書き直さなければならなかったため、Z80なればこそメリットがあったプログラムをそのまま8080用に書き直すというかなり強引なプログラムになってしまいました。

結果としてプロが作ったと自慢できるほどのプログラムではなくなってしまいました。

ま、その辺はお許しいただくとしまして、序文にありますように、なにはともあれ、レトロな世界をお楽しみください。

2016年4月　（有)中日電工代表者　菱田照久　記

以下は2003年にND80K用として編集した応用プログラムの序文です。

本編はND80Kのためのゲームプログラム集です。

ND80Kは20年前にNECのTK80とソフトコンパチブルとして発売したND80の後継機種です。

発売当初はTK80と同じくCPUは8080でしたが、その後Z80を採用し（NECは8085を使ったTK85を発売した）、マイコンがパソコンになって、Windows全盛の今日まで、当時に近い機能のまま今日に至っています。

その間参考プログラム集の作成も検討はしてきたのですが、世の動きに合わせて私の関心も機械語からアセンブラへ移り（8080アセンブラを自作し、ついでに逆アセンブラも作りこれは後にZ80アセンブラ、逆アセンブラに発展します）、さらにはBASICへと展開していきました。当然のことながらBASICも整数型のBASICからNECのPC8001に刺激されて浮動小数点演算やカラーグラフィック機能をも搭載したBASICの製作とハードの製作という、およそ零細企業にあるまじき狂気のふるまいの中に埋没してしまい、ND80のZ80版であるND80Zも改良を続けながら販売しては来たのですがいつしか惰性に流され、ND80Z参考プログラム集の製作も着手しないまま今日まで来てしまいました。

ND80KはCPUをZ80から川崎製鉄のKL5C8012に強化するとともに、DOS／Vパソコンと接続してプログラム開発ができる機能（オプション）をも付加して2000年に発売を開始しました。しかし過去からのいきさつからこのボードを作ったものの、今更TK80コンパチでもあるまいし、ということで、勿論あらためて参考プログラムを作る気もありませんでした。

ところがつい先日（2003年4月）、当社のホームページを見た方から電話をいただき、TK80のシミュレーションソフトが販売されていること、そしてそれを購入したけれども、やはりパソコン上での模擬体験ではなく実物がさわってみたい、というお話しを聞きました。

さっそくネットで検索して、はじめてそういうものが売られていることを知りました（いかに自分がTK80から離れていたかの証左です）。

しかし！これはなんなのだ！ふつうシミュレーションとは実物が容易に体験できないものが対象であるはず（たとえば航空機の操縦とか、電車でGOとか）。しかるにたかがワンボードマイコンをバーチャルで体験するとは何たる世の中であるか！ここに実物が、ND80Kがあるではないか！

もーぜんと闘志がわいてきました。

参考プログラム集も作ってやろーではないか！と怒りにまかせてゴールデンウィークを棒に振って作成したのが本書です。

本書中のプログラムのほとんどは、当時出版されていた、岸田孝一著「マイコンゲーム21」（産報出版）からアイデアを拝借しました。この本はいつか参考プログラム集を出すときに参考にしようと思い、本箱の隅に眠っていたものですが、私自身はゲームには興味がなくまたそんなひまもなかったのでプログラムを打ち込んでみることもなくそのままになっていました。

参考プログラム集を作成しようと思い立ったとき真っ先に頭に浮かんだのが「マイコンゲーム21」でした。

TK80ソフトコンパチを売り物にしてきたとはいえ、ND80KはTK80とはクロックもメモリ容量も比較にならぬほど速く、そして大容量です。「マイコンゲーム21」のニーモニックは8080（インテル）であるのに対して、ND80KではZ80（ザイログ）ニーモニックです。またZ80は8080に倍化する新しい命令が使えます。当然「マイコンゲーム21」のプログラムをそのまま使うというわけにはいきません。

またどうせ作るなら、「マイコンゲーム21」で作者が述べているように、こちらもプロのプログラマが作ったらこうなるというところを見せようではないか、と思い至りました。そこでできるだけ「マイコンゲーム21」のゲームプログラムと同じ動作をするプログラムを作ってみることにしました。しかしプロの意地でもあって、参考にしたのは各ゲームの説明部分のみでプログラムリストそのものは読んでいません（説明のみを読んでこの程度のプログラムがすぐに書けないようではプロにはなれません）。

この参考プログラム集は、そういうわけでプロの意地などという不純な動機でもって作成されたもので、良い教科書となりうるものではないかもしれません。かなりアクの強いテクニックを随所で用いています。このようなテクニックはプログラマとして必ずしも必要なものではありませんから、敢えて学ぶ必要は無いとも思います。こういうクセは知らず知らずの

うちに身にしみついてきてしまいます。初心者にはわかりづらいところがあるかもしれません。そこは「意地」に免じてお許しください。できるだけ説明を入れるようにしました。
もしも「マイコンゲーム21」をお持ちの方でしたら、両方のプログラムを比較することで、同じ動きをするプログラムでも作者が違うこととCPUが8080に対してZ80であることで全く異なっている（多分そのはずです）ことに驚かれると思います。
「マイコンゲーム21」にはその名の通り21本のゲームが紹介されています。しかしこの参考プログラム集ではそのうち何本かは割愛しました。当時と違いパソコンやTVゲームがあるのですから、8ケタのLEDだけで2次元的なゲームを表現するのは余りに無理があるように感じたからです。
ND80Kは最近のWindowsパソコンから見たらまるで新幹線と自転車のようですが、しかしながらWindowsパソコンでC＋＋やVisual Basicなどのプログラムを作るのは相当の知識を必要とするのに対し、かえってむずかしいはずの機械語の方が簡単に扱えるように思えます。
なにはともあれ、レトロなマイコンの世界をじっくりお楽しみ下さい。

2003年5月　（有)中日電工代表者　菱田照久　記

ⅱ）プログラムリストについて

プログラムはWindowsパソコンを使い、当社オリジナルの8080アセンブラ（ASM80）で作成しました。
ND8080は機械語で入力しますから、リストのうち必要なのは左端の4桁のアドレスとその次の機械語コードのみです。
TK80は標準ではRAMが8200～83FFまでしか実装されていませんでした（たった512バイト！です）。そのためTK80用に書かれたプログラムは大抵は8200番地から始っています。
ND8080は標準ではRAMとして32KB（キロバイト）の62256を実装していますから、TK80の64倍のメモリエリアがあることになります。したがってTK80と同じように8200番地からプログラムを書くこともできますが、ND8080はプログラムの開始アドレスは8000番地からとなっているため、通常は8000番地から書くようにします。
プログラムによってはCPUレジスタだけでは不足することがあってRAM上にワークエリアを設けます。システムが使わなくて、ユーザープログラムとも重ならないRAMアドレスならばどこでもよいのですが、このゲームプログラム集では8200～に割り当てています。
実はこのゲーム集のプログラムは8080用に新規に作成したものではなくて、Z80用のソースプログラムを8080用にコンバートして、それに加筆訂正したものです。
そのため8080にはない命令には”？”がついていて、その前後にそれに代わる8080の命令文が書かれています。

ⅲ）附属CDROMのプログラム、データの扱い方

ND8080組立キットの附属CDROMにはこの応用プログラム集に記載したゲームプログラム、電子オルゴールプログラム、曲データが全て収録されています。
CDROM（nd8080フォルダ）にはプログラムがTK80形式のバイナリファイルで入っています（ソースプログラムはASM80フォルダにあります）。
nd8080フォルダごとハードディスクにコピーしてください。
プログラムのロードはコマンドプロンプトで操作します。詳細についてはND8080操作説明書、USB接続説明書を参照してください。

1）TK80モードでの操作を前提にしていますから、ND8080はTK80モニタモードで使ってください。
2）USBケーブルでND8080とWindowsパソコンを接続します。
3）ND8080のキーボードの［REG］キー（TK80モードでは［LOAD］キーとして働きます）を押してください。受信スタンバイになります。LED表示は変わりませんが［MON］キー（［RESET］キー）以外は受け付けなくなります。
4）Windowsパソコンのコマンドプロンプト画面からHIDWR8コマンドを使って、ND8080で実行したいプログラム（拡張子は．BTKです）を送信します。
　＞hidwr8　ohayo．btk［Enter］

［注記］プログラムは8000番地からLOADされます。

プログラム№1　おはよー（OHAYO8．BTK）

これはゲームではありません。「マイコンゲーム21」にもありません。雑誌だったのか、どういう本に載っていたのかも思い出せません。実は私自身完全に忘れていました。覚えていたのはうちのカミさんでした。その昔に、私が何かの本を見てこんな動作をするプログラムを入力して動かしてみせたのを覚えているというのです（げに世の女房族の記憶力には恐怖すべきものがあります）。

カミさんの記憶を頼りに作ったのがこのプログラムです。ゲームではありませんが、オープニングの肩ならしにはちょうどよいと思います。

ではさっそくスタートです。

プログラムを入力して、[8][0][0][0][ADRSSET][RUN]とするとコンピュータが起きあがってのそりのそりと歩き始めます（LEDに表示されるのは足跡のみ）。目をパチパチさせて、それからゆっくりと「おはよー」と声をかけます（言葉は話せませんからLEDにそれもローマ字で表示します）。

このプログラムはこれでおしまいです。

プログラムの説明

（以下の説明文に出てくる命令やレジスタの機能などの基本的な内容や詳細は、ND8080に付属の8080命令説明書を参照してください。またND8080のメモリアドレスやシステムサブルーチン、キーの操作などはND8080操作説明書を参照してください）

LEDに0～F以外のパターンを表示するとか、表示を全部クリアして（0を表示するのではなくて）ブランクにしたいときなどは、LED表示アドレス（＄83F8～＄83FF）に直接書き込みます。各アドレスの8ビットのデータのうち1のビットに対応するLEDのセグメントが点灯し0のビットに対応するセグメントは消灯します。以下のプログラムの中でLED1～LED8に対してデータを書き込んでいるところは全部この目的で使われています。

;EYE　close／openやサブルーチンCLRがその例です。;ASIATO　dispと;ohayo　dispでは表示するデータを8バイト分用意しておいて、全部の表示を順次置換えています。

サブルーチンCLRは＄83F8～＄83FFに00を書き込むことでLEDを全消灯しています。

;ASIATO　dispと;ohayo　dispで使っているLDIは8080には無い命令なのでサブルーチンとしました。

本当は機械的にサブルーチン化するのではなくて、目的に合わせて適切なプログラムを書くべきなのですが、時間が無い中での作業のため、安易な方法になってしまいました。

```
2016/4/20  7:39  ohayo8.txt
END=8087
              ; OHAYO for NDZ
              ; 03/04/28 5/9
              ;10/6/1 for ND80Z3
              ;16/4/17 for nd8080
              ;
                   ORG $8000
                   LED1=$83F8;$FFF8
                   LED4=$83FB;$FFFB
                   START1=$0051;$080F
                   TM5M=$02DD
              ;
8000 CD5180        CALL CLR
              ;ASIATO disp
8003 217280        LXI H,ASIDT
8006 11F883        LXI D,LED1
              ;     LXI B,$0008
8009 0E08        MVI C,08
800B CD5E80   ASIDP2:CALL TM1S
              ;?      LDI
800E CD8280      CALL LDI;
              ;     JPE ASIDP2
8011 C20B80      JNZ ASIDP2
8014 CD5E80        CALL TM1S
8017 CD5180        CALL CLR
```

```
801A CD5E80         CALL TM1S
             ;EYE close/open
801D 0603           MVI B,03
801F 211C1C         LXI H,$1C1C
8022 113F3F         LXI D,$3F3F
8025 22FB83    EYE:SHLD LED4
8028 CD6480         CALL TM025
             ;?       LD (LED4),DE
802B EB            XCHG;
802C 22FB83        SHLD LED4;
802F EB            XCHG;
8030 CD6180          CALL TM05
             ;?       DJNZ *EYE
8033 05            DCR B
8034 C22580        JNZ EYE
8037 CD5180          CALL CLR
             ;ohayo disp
803A 217A80          LXI H,OHAYODT
803D 11F883          LXI D,LED1
             ;      LXI B,$0008
8040 0E08          MVI C,08;
8042 CD6180    OHYDP2:CALL TM05
             ;?       LDI
8045 CD8280        CALL LDI;
             ;      JPE OHYDP2
8048 C24280        JNZ OHYDP2;
804B CD5E80        CALL TM1S
804E C35100          JMP START1
             ;LED clear
8051 21F883    CLR:LXI H,LED1
8054 010008          LXI B,$0800
8057 71        CLR2:MOV M,C
8058 23              INX H
             ;?       DJNZ *CLR2
8059 05            DCR B;
805A C25780        JNZ CLR2;
805D C9              RET
             ;1sec timer/0.5sec timer/0.25sec timer
805E CD6180    TM1S:CALL TM05
8061 CD6480    TM05:CALL TM025
8064 D5        TM025:PUSH D
8065 1E32            MVI E,32;=50
8067 D5        TM025_2:PUSH D
8068 CDDD02        CALL TM5M;4.684ms
806B D1            POP D
806C 1D              DCR E
806D C26780          JNZ TM025_2
8070 D1              POP D
8071 C9              RET
             ;asiato data
8072 43        ASIDT:DB 43
8073 4C              DB 4C
8074 43              DB 43
8075 4C              DB 4C
8076 43              DB 43
8077 4C              DB 4C
8078 43              DB 43
8079 4C              DB 4C
```

```
                ;"ohayo---"
807A 3F         OHAYODT:DB 3F;O
807B 76              DB 76;H
807C 77              DB 77;A
807D 6E              DB 6E;y
807E 3F              DB 3F;O
807F 40              DB 40;-
8080 40              DB,40;-
8081 40              DB 40;-
                ;
8082 7E         LDI:MOV A,M
8083 12            STAX D
8084 23            INX H
8085 13            INX D
8086 0D            DCR C
8087 C9            RET
                ;
ASIDP2          =800B  ASIDT        =8072  CLR          =8051
CLR2            =8057  EYE          =8025  LDI          =8082
LED1            =83F8  LED4         =83FB  OHAYODT      =807A
OHYDP2          =8042  START1       =0051  TM025        =8064
TM025_2         =8067  TM05         =8061  TM1S         =805E
TM5M            =02DD
```

プログラム№2　CRAZY　EIGHT（CRAZY88．BTK）

ゲームの説明

8000番地からRUNするとスピーカからピーと音が出て、LEDが全部消え、キー入力待ちになります。

ゲームはLED1～LED8のどこをコンピュータが選ぶか予測してその番号をキー入力するものです。

1～8のいずれかのキーを押すとその数に対応するLEDに”8”の下半分が表示されます。次の瞬間にコンピュータが選んだ位置に”8”の上半分が表示されます。もし位置が一致すれば”8”の字が完成しピッピッという断続音とともにその”8”が点滅したあと点灯して静止します。位置が異なれば下半分と上半分が離れて表示されるだけで何もおこりません。続いて残った消灯しているLEDから同じようにコンピュータの選択を予測して、その位置の数をキー入力します。これを繰り返してブランクが全て無くなればゲーム終了です。約2秒後にLEDの右2桁に獲得した点数が表示されます。点数は”8”ができるごとに加算されます。加算される点数はその”8”の表示直前のLEDが消灯している数になります。ゲーム開始直後は全部消灯していますから、最初にうまくヒットして”8”ができれば8点ゲットです。失敗すると2ヶ所が点灯しますから残りは6個で、6点が次の”8”完成によって加算される得点になります。

数を入力するときに間違えて1～8以外のキーを押したり、すでに表示済みのLEDを示す数を入力しても無視されます。

点数表示状態でどこかキーを押すと最初にもどって再びゲームが開始されます。このとき得点もゼロクリアされます。

プログラムの説明

CPUのレジスタだけでは足りないのでメモリ上にワークエリアを設けます。キー入力位置を記憶するYRADRS（your addressのつもり。名前は自分がわかる適当な名前をつける。これはアセンブラに必要なもので機械語に直してしまえば、ただのアドレスになってしまう。この場合は8003）、LEDの残り消灯数を格納するREST、乱数の作業用にRNDDT、RNDDT2、獲得した点数をカウントするCNTRです。

CNTRは普通のRAMエリアではなくて、システムのデータエリアである83F7を指定しています。

このあたりが機械語プログラムを書くときにアクの強さが出てくるところです。83F7はLEDの表示用のデータエリアで、このプログラムでは最後の点数表示以外では使用しないのではじめから表示用データエリアをカウンタとして使ってしまおうというわけです。なおこのカウンタは点数を加算し、それを10進で表示するため、バイナリではなくて10進で演算します。

さてキー入力ですが、1～8の入力のチェックにCPI　09はよいのですが、ORA　A、JZとしないのはなぜでしょう？

そうする代わりにDCR　A、JMとしています。

こういうところにプログラマの性格が現れます。なんともせっかちな…。

じつはここで入力した数は、そのあとで対応するLEDアドレスを計算するのに使いますが、そこでは1～8ではなくて0～7でないと都合が悪いのです。どうせ後でDCRするなら、ここでDCRを実行して同時に0キー入力も除外してしまおうというささやかなアイデアです。

なおLEDアドレスの算出は本来は、

```
LXI　H，LED1
MOV　E，A
MVI　D，00
DAD　D
```

とするべきです。ここでは、ADD　Lとしています。わかっていてするのはテクニックですが初心者は誤解してしまいます（教科書としては不適切です）。こういうプログラムを書くと学校のテストなら×になるかもしれません。しかし私が先生ならこの設定でこのプログラムは◎をあげます。

上で例示したプログラムはどんな場合でも通用するプログラムです。初心者はこう書けば間違いありません。しかし今回はデータに一定範囲の条件があります。Aの値は0～7で、HL＝83F8という条件です。この場合にはADD　Lでも、ORA　Lでも構いません。

さてMYTURNでは乱数を使って、コンピュータ側の1～8を発生させています。実際には上と同じ理由で0～7を用いています。乱数については後で説明します。

とにかくサブルーチンRND3をCALLすると、00～07の範囲のランダムな数が得られます。プレーヤーが入力した数（YRADRS）と一致すれば”8”になりますが、それ以外ですでに点灯済みのLEDを選択するのは無効ですから、それをチェックして、点灯済みなら乱数発生からやりなおします。

結果が不一致の場合にはLEDの残りが2つ減りますから、（REST）に対して2回DECを実行します。その結果（REST）＝0になったらゲーム終了です。

結果が一致したらヒットです。HIT：でその処理をします。サウンドの出力と、”8”の点滅表示のあと、点数を加算します。（CNTR）に（REST）を加算します。10進数（BCD）として扱うため10進補正命令、DAAを使っています。

ではなぜ10進数なのか？なぜ10進補正が必要なのか？

それは（CNTR）の値を最後にLEDに表示させるためです。通常の2進演算の場合、09＋01は0Aになります。これを（CNTR）に入れてLEDにそのまま表示させると、0Aと表示されてしまいます。09＋01の加算後にDAAを実行するとAレジスタの値は、10になります。これをそのままLEDに表示すれば10進の計算結果として正しく点数が表示されること

になります。
結果が一致したときは、LEDの残りは1つしか減りません。(REST)に対してDCRを1回だけ実行します。その結果(REST)が奇数の場合と偶数の場合が出てきます。その両方の場合でゲーム終了の判断が異なります。
END:がゲーム終了での処理です。ここでやっとCNTRとしてLEDの表示用データレジスタをそのまま使用したわけが明らかになります(なんとずる賢い)。
データ表示ルーチン01C0をCALLすれば何もしなくても点数がLEDの右2桁に表示されます。残りの6桁には無関係な値が表示されてしまいますからLEDレジスタに00を書き込んでLED1～LED6の表示を消してしまいます。

乱数について

ゲームではよくサイコロを使います。ところがコンピュータでサイコロを作るのはなかなか大変なのです。本格的な乱数発生の計算方法もいろいろ考案されているらしいのですが、たかがワンボードのゲームのために複雑な計算をするのはちょっと考え物です。といっていいかげんではゲームになりません。一番よいのは乱数表をメモリに入れておくことなのですが、仮にそうしたとしても、そのどこを読むかが問題です。毎回同じアドレスを選択したり、その選択する順序が毎回同じであったりすれば乱数にはなりません。
ここでは擬似的な乱数表として、ROMに書かれたプログラムの命令コードを利用しています。コードの並びはプログラムとしては意味がありますが、数としてみれば大小順序はばらばらで乱数的と言えます。しかし完全ではなくて、よく使われる命令コードやアドレスは頻繁に現れます。
乱数発生サブルーチンRND3では連続して同じ数が出ないように工夫しています。また次のアドレスを算出するときも毎回同じ流れにならないようにするため、キー入力のときにコールされるチャタリングタイマーに乱数用の初期値カウンタR(＝FFD2)を組み込んで、タイマーがコールされるたびに＋1されるようにして、それを利用しています。
RNDの実行結果はまあまあですが、ときとしてちょっとできの悪いサイコロになることもあります。とりあえずは簡単なゲーム用ですから、こんなものでしょう。

```
2016/4/20  11:50  crazy88.txt
END=8136
              ; CRAZY EIGHT for NDZ
              ; 03/05/07 5/9
              ;10/6/1 for ND80Z3
              ;16/4/16 for nd8080
              ;4/18 sound,timer
              ;4/19 rnd
              ;4/20
              ;
                   ORG $8000
              ;     YRADRS=$E800
              ;     REST=$E801
              ;     RNDDT=$E810;$E811
                   CNTR=$83F7;$FFF7;=DP4
                   LED1=$83F8;$FFF8
                 R=$FFD2
              ;
                   DTDPS=$01C0;$05C0;DATA DISP
                   KEYIN1=$0216;$0616
                   D1=$02DD;4.684msec
              ;
              ;data set
8000 C30880      JMP START
8003 00        YRADRS:NOP
8004 00        REST:NOP
8005 00        RNDDT:NOP
8006 00          NOP
8007 00        RNDDT2:NOP
8008 3AD2FF    START:LDA R
800B 210580      LXI H,RNDDT
800E 77          MOV M,A
800F 23          INX H
8010 77          MOV M,A
8011 CDD880      CALL CLR
```

```
8014 AF            XRA A
8015 32F783        STA CNTR;DECIMAL COUNTER clear
8018 3E08          MVI A,08
801A 320480        STA REST
                ;GAME START
801D 3E10          MVI A,10;puuuu
801F 060A          MVI B,0A
8021 CDE580     START1:CALL SOUND
                ;?      DJNZ START1
8024 05            DCR B;
8025 C22180        JNZ START1;
8028 CD1602     KEY:CALL KEYIN1
802B FE09          CPI 09
802D D22880        JNC KEY
8030 3D            DCR A
8031 FA2880        JM KEY
                ;keyin position check
8034 320380        STA YRADRS;position save
8037 21F883        LXI H,LED1
803A 85            ADD L
803B 6F            MOV L,A
803C 7E            MOV A,M
803D B7            ORA A
803E C22880        JNZ KEY;already used
8041 365C          MVI M,5C;"low" disp
                ;my turn
8043 CDBA80     MYTURN:CALL RND3;0-7 select
                ;     ANI 07
8046 57            MOV D,A
8047 3A0380        LDA YRADRS
804A BA            CMP D
804B CA7180        JZ HIT
804E 7A            MOV A,D
804F 21F883        LXI H,LED1
8052 85            ADD L
8053 6F            MOV L,A
8054 7E            MOV A,M
8055 B7            ORA A
8056 CA6480        JZ MYTN2;not used
8059 3A0480        LDA REST
805C FE01          CPI 01
805E CA9F80        JZ END;no rest
8061 C34380        JMP MYTURN
8064 3663       MYTN2:MVI M,63;"high" disp
8066 210480        LXI H,REST
8069 35            DCR M
806A 35            DCR M
806B C22880        JNZ KEY
806E C39F80        JMP END
                ;
8071 21F883     HIT:LXI H,LED1
8074 85            ADD L
8075 6F            MOV L,A
8076 0E05          MVI C,05
8078 367F       HIT1:MVI M,7F;"8" disp
807A 3E17          MVI A,17;piii
807C 0603          MVI B,03
807E CDE580     HIT2:CALL SOUND
```

```
                ;?      DJNZ HIT2
8081 05           DCR B;
8082 C27E80       JNZ HIT2;
8085 3600           MVI M,00
8087 CD2681         CALL TM02S
808A 0D             DCR C
808B C27880         JNZ HIT1
808E 367F           MVI M,7F
8090 210480         LXI H,REST
8093 3AF783         LDA CNTR;point up
8096 86             ADD M
8097 27             DAA
8098 32F783         STA CNTR
809B 35             DCR M
809C C22880         JNZ KEY
809F 0614       END:MVI B,14;=20 ,2sec wait
80A1 CD2981     WAIT:CALL TM01S
                ;?      DJNZ WAIT
80A4 05           DCR B;
80A5 C2A180       JNZ WAIT;
80A8 CDC001         CALL DTDPS
80AB 21F883         LXI H,LED1
80AE 0606           MVI B,06
80B0 AF             XRA A
80B1 CDDE80     END2:CALL CLR1
80B4 CD1602         CALL KEYIN1
80B7 C30880         JMP START
                ;
                ;ransu
80BA E5         RND3:PUSH H
80BB D5              PUSH D
80BC 2A0580     RND2:LHLD RNDDT
80BF 23           INX H
80C0 7C           MOV A,H
80C1 E603         ANI 03
80C3 67           MOV H,A
80C4 220580       SHLD RNDDT
80C7 3A0780       LDA RNDDT2
80CA 57           MOV D,A
80CB 7E             MOV A,M
80CC E607         ANI 07
80CE BA             CMP D
80CF CABC80       JZ RND2
80D2 320780       STA RNDDT2
80D5 D1             POP D
80D6 E1             POP H
80D7 C9             RET
                ;
                ;LED all clear
80D8 21F883     CLR:LXI H,LED1
80DB AF              XRA A
80DC 0608            MVI B,08
80DE 77         CLR1:MOV M,A
80DF 23              INX H
                ;?      DJNZ *CLR1
80E0 05           DCR B;
80E1 C2DE80       JNZ CLR1;
80E4 C9             RET
```

```
                ;
                ;
80E5 F5         SOUND:PUSH PSW
80E6 E5            PUSH H
80E7 D5            PUSH D
80E8 C5            PUSH B
80E9 210E81        LXI H,SNDTBL
80EC 85            ADD L
80ED 6F            MOV L,A
80EE 46            MOV B,M
80EF 1E1A          MVI E,1A
80F1 50         SNDS1:MOV D,B
80F2 3EFF          MVI A,FF;SP OUT=H
80F4 D398          OUT 98
80F6 7F         SNDS2:MOV A,A;5---------
80F7 15            DCR D;5          | 5+5+10=20  20/2=10microsec
80F8 C2F680        JNZ SNDS2;10----
80FB 50            MOV D,B
80FC 3EDF          MVI A,DF;sp out=L
80FE D398          OUT 98
8100 7F         SNDS3:MOV A,A;5---------
8101 15            DCR D;5          | 5+5+10=20  20/2=10microsec
8102 C20081        JNZ SNDS3;10----
8105 1D            DCR E
8106 C2F180        JNZ SNDS1
8109 C1            POP B
810A D1            POP D
810B E1            POP H
810C F1            POP PSW
810D C9            RET
                ; SOUND TABLE
810E 7F         SNDTBL:DB 7F;so4
810F 77            DB 77;so#4
8110 71            DB 71;ra4
8111 6A            DB 6A;ra#4
8112 5F            DB 5F;do5
8113 59            DB 59;do#5
8114 54            DB 54;re5
8115 4F            DB 4F;re#
8116 47            DB 47;fA5
8117 43            DB 43;fa#5
8118 3F            DB 3F;so5
8119 3B            DB 3B;so5#
811A 35            DB 35;ra#5
811B 32            DB 32;si5
811C 2F            DB 2F;do6
811D 2C            DB 2C;do#6
811E 25            DB 25;mi6
811F 27            DB 27;re#6
8120 2A            DB 2A;re6
8121 4B            DB 4B;mi5
8122 38            DB 38;ra5
8123 64            DB 64;si4
8124 23            DB 23;fa6
8125 21            DB 21;fa#6
                ;
                ;0.2sec timer
8126 CD2981     TM02S:CALL TM01S
```

```
                ;0.1sec timer
8129 D5         TM01S:PUSH D
812A 1E15          MVI E,15;=21
812C D5         TM01S2:PUSH D
812D CDDD02        CALL D1;4.684ms
8130 D1            POP D
8131 1D              DCR E
8132 C22C81          JNZ TM01S2
8135 D1              POP D
8136 C9              RET
                ;
CLR           =80D8  CLR1          =80DE  CNTR          =83F7
D1            =02DD  DTDPS         =01C0  END           =809F
END2          =80B1  HIT           =8071  HIT1          =8078
HIT2          =807E  KEY           =8028  KEYIN1        =0216
LED1          =83F8  MYTN2         =8064  MYTURN        =8043
R             =FFD2  REST          =8004  RND2          =80BC
RND3          =80BA  RNDDT         =8005  RNDDT2        =8007
SNDS1         =80F1  SNDS2         =80F6  SNDS3         =8100
SNDTBL        =810E  SOUND         =80E5  START         =8008
START1        =8021  TM01S         =8129  TM01S2        =812C
TM02S         =8126  WAIT          =80A1  YRADRS        =8003
```

プログラム№3　Hi－Lo（HILO8．BTK）

ゲームの説明

8000番地からRUNするとLEDの左2桁に、－－が表示され、右4桁に0000が表示されます。このときコンピュータは乱数を利用して4桁の数0000～9999を算出して隠しています。それを当てるゲームです。

キーから適当な数、たとえば4567と入力してWRITE　INCキーを押すと、コンピュータは入力された数と、自分が隠している数とを比較して、入力された数が隠した数よりも大きければLEDの左2桁に、Hi　と表示します。小さければ、Lo　と表示します。この表示を参考にして、なるべく少ない回数で正解を得るのが目的です。

キーからの数値の入力はWRITE　INCキーを押さない限りは何回でも入れなおしができます。

正解が出ると、LEDの左2桁にそれまでの入力回数が表示され、LED全体が約2秒間点滅したあと、最初に戻って新しい数が選択されてゲームが再開されます。

プログラムの説明

最初にRND4サブルーチンを4回CALLして4桁のBCD数を求めて、HIDNDTに格納します（hidden　dataのつもり）。試行回数をカウントするCNTRは先回のCRAZY　EIGHTと同じ手ですが、今回はLEDの左2桁に結果を表示するため、アドレスは83EFに割り当てています。先回は表示用データレジスタ（83F4～83F7）をカウンタに利用したのですが、今回はアドレス、データレジスタ（83EC～83EF）を利用しています。今回のプログラムではキー入力された数値をデータレジスタに一度格納してからそれを表示するので、カウンタもアドレスレジスタに置かなければなりません。キー入力された数値はLED表示用データレジスタに置くこともできるのですが、そうするとHとLの位置が逆になるので、プログラムが少し面倒になります。

（参考）

```
HLの数値をデータレジスタに入れる場合
SHLD　DTRG
HLの数値をLED表示用データレジスタに入れる場合
XCHG
LXI　H, DP3;DP3＝＄83F6
MOV　M, D
INX　H
MOV　M, E
```

;compare　DATA　でデータの比較を行っています。HL、DEともにBCD数なので、SBC　HL, DE（この命令は8080にはないのでサブルーチンSBCHLDEをコール）を行っても正しい結果は得られません。しかしここでは差を計算することが目的ではなくて大小がわかればよいので、比較命令としてつかっているのでこれでよいのです。

ALBLNKサブルーチンではND8080のDMAによるLEDの表示機能を利用しています。

LEDの一部だけを点滅させる場合には、表示データを一定時間毎にブランクと入れ替えて表示します。

しかしここでは全部を点灯、消灯すればよいのですから、LED表示回路が行っているDMAによる表示を禁止したり、許可したりすればよいことになります。LEDONはLED表示のDMA許可ルーチンでLEDOFFはDMA禁止ルーチンです。

```
2016/4/20　11:49　hilo8.txt
END=8102
              ; Hi-Lo for NDZ
              ; 03/04/29 4/30 5/1 5/3 5/5 5/9 5/10
              ;10/6/1 for ND80Z3
              ;16/4/16 for ND8080
              ;4/18
              ;4/19 rnd
              ;4/20
              ;
                   ORG $8000
              ;     HIDNDT=$E800;$E801
              ;     RNDDT=$E810;$E811
                   DTRG=$83EC;$FFEC
                   ADRG=$83EE;$FFEE
                   CNTR=$83EF;$FFEF;=ADRGH
                   LED1=$83F8;$FFF8
```

```
                    LED3=$83FA;$FFFA
                  R=$FFD2
               ;

                    ADDPS=$01A1;$05A1
                    KEYIN1=$0216;$0616
                  D1=$02DD;4.684msec
               ;
8000 C30880       JMP START
8003 00        HIDNDT:NOP
8004 00           NOP
8005 00        RNDDT:NOP
8006 00           NOP
8007 00        RNDDT2:NOP
               ;data(hidden) set
8008 3AD2FF    START:LDA R
800B 210580       LXI H,RNDDT
800E 77           MOV M,A
800F 23           INX H
8010 77           MOV M,A
8011 0604         MVI B,04
8013 CD8480    DTST1:CALL RND4
               ;     ANI 0F
8016 FE0A           CPI 0A
8018 D21380         JNC DTST1
801B 29             DAD H
801C 29             DAD H
801D 29             DAD H
801E 29             DAD H
801F B5             ORA L
8020 6F             MOV L,A
               ;?       DJNZ *DTST1
8021 05           DCR B;
8022 C21380       JNZ DTST1;
8025 220380         SHLD HIDNDT
8028 210000         LXI H,$0000
802B 22EE83         SHLD ADRG;DECIMAL COUNTER clear
802E 22EC83    REENT:SHLD DTRG
8031 E5             PUSH H
8032 CDA101         CALL ADDPS
8035 21CF80         LXI H,STDT
8038 CDC580    REENT1:CALL H4DP
803B E1             POP H
               ;key entry(0000-9999)
803C CD1602    KEY:CALL KEYIN1
803F FE15           CPI 15;WRITE INC
8041 CA5280         JZ COMP
8044 FE0A           CPI 0A
8046 D23C80         JNC KEY
8049 29             DAD H
804A 29             DAD H
804B 29             DAD H
804C 29             DAD H
804D B5             ORA L
804E 6F             MOV L,A
804F C32E80         JMP REENT
               ;compare DATA
8052 3AEF83    COMP:LDA CNTR;DECIMAL COUNTER up
```

```
8055 C601          ADI 01
8057 27            DAA
8058 32EF83        STA CNTR
805B B7            ORA A;clear CARRY
                ;?      LD DE,(HIDNDT)
805C EB           XCHG;
805D 2A0380       LHLD HIDNDT;
8060 EB           XCHG;
8061 E5             PUSH H
                ;?      SBC HL,DE
8062 CDDB80       CALL SBCHLDE;
8065 CA7480         JZ SAME
8068 21D780         LXI H,LODT
806B DA3880         JC REENT1
806E 21D380         LXI H,HIDT
8071 C33880         JMP REENT1
8074 E1         SAME:POP H
8075 CDA101         CALL ADDPS
8078 210000         LXI H,$0000
807B 22FA83         SHLD LED3
                ;OK! ALL LED blink
807E CDA280         CALL ALBLNK
8081 C30880         JMP START
                ;
                ;ransu
8084 E5         RND4:PUSH H
8085 D5              PUSH D
8086 2A0580     RND2:LHLD RNDDT
8089 23            INX H
808A 7C            MOV A,H
808B E603          ANI 03
808D 67            MOV H,A
808E 220580        SHLD RNDDT
8091 3A0780        LDA RNDDT2
8094 57            MOV D,A
8095 7E              MOV A,M
8096 E60F          ANI 0F
8098 BA              CMP D
8099 CA8680        JZ RND2
809C 320780        STA RNDDT2
809F D1              POP D
80A0 E1              POP H
80A1 C9              RET
                ;
                ;LED all blink
80A2 060A       ALBLNK:MVI B,0A;=10
80A4 3EEF       ALBLNK1:MVI A,EF
80A6 D398          OUT 98;LED OFF
80A8 CDB780          CALL TM01S
80AB 3EFF            MVI A,FF
80AD D398          OUT 98;LED ON
80AF CDB780          CALL TM01S
                ;?      DJNZ *ALBLNK1
80B2 05            DCR B;
80B3 C2A480        JNZ ALBLNK1;
80B6 C9              RET
                ;
                ;0.1sec timer
```

```
80B7 D5       TM01S:PUSH D
80B8 1E15        MVI E,15;=21
80BA D5       TM01S2:PUSH D
80BB CDDD02      CALL D1;4.684ms
80BE D1          POP D
80BF 1D            DCR E
80C0 C2BA80        JNZ TM01S2
80C3 D1            POP D
80C4 C9            RET
              ;
              ;High 4 disp
80C5 11F883   H4DP:LXI D,LED1
80C8 010400        LXI B,$0004
              ;?      LDIR
80CB CDF680      CALL LDIR;
80CE C9            RET
              ;
80CF 40       STDT:DB 40
80D0 40            DB 40
80D1 00            DB 00
80D2 00            DB 00
80D3 76       HIDT:DB 76;H
80D4 06            DB 06;i
80D5 00            DB 00
80D6 00            DB 00
80D7 38       LODT:DB 38;L
80D8 3F            DB 3F;0
80D9 00            DB 00
80DA 00            DB 00
              ;
80DB C5       SBCHLDE:PUSH B
80DC 47          MOV B,A
80DD 7D          MOV A,L
80DE 9B          SBB E
80DF 6F          MOV L,A
80E0 7C          MOV A,H
80E1 9A          SBB D
80E2 67          MOV H,A
80E3 DAED80      JC SBCHLDE1
80E6 FAED80      JM SBCHLDE1
80E9 B5          ORA L
80EA C2F080      JNZ SBCHLDE2
80ED 78       SBCHLDE1:MOV A,B
80EE C1          POP B
80EF C9          RET
80F0 3E01     SBCHLDE2:MVI A,01
80F2 B7          ORA A;clear sf
80F3 78          MOV A,B
80F4 C1          POP B
80F5 C9          RET
              ;
80F6 F5       LDIR:PUSH PSW
80F7 7E       LDIR2:MOV A,M
80F8 12          STAX D
80F9 23          INX H
80FA 13          INX D
80FB 0B          DCX B
80FC 78          MOV A,B
```

```
80FD B1          ORA C
80FE C2F780      JNZ LDIR2
8101 F1          POP PSW
8102 C9          RET
                 ;
ADDPS       =01A1  ADRG        =83EE  ALBLNK      =80A2
ALBLNK1     =80A4  CNTR        =83EF  COMP        =8052
D1          =02DD  DTRG        =83EC  DTST1       =8013
H4DP        =80C5  HIDNDT      =8003  HIDT        =80D3
KEY         =803C  KEYIN1      =0216  LDIR        =80F6
LDIR2       =80F7  LED1        =83F8  LED3        =83FA
LODT        =80D7  R           =FFD2  REENT       =802E
REENT1      =8038  RND2        =8086  RND4        =8084
RNDDT       =8005  RNDDT2      =8007  SAME        =8074
SBCHLDE     =80DB  SBCHLDE1    =80ED  SBCHLDE2    =80F0
START       =8008  STDT        =80CF  TM01S       =80B7
TM01S2      =80BA
```

プログラム№4　FLIPFLOP（FLIPFLOP8．BTK）

ゲームの説明

8000番地からRUNするとLEDに”8”の上半分が8個表示されます。各桁を示す1～8の数をキーから入力すると、その位置のLEDが点滅します。それでよければWRITE　INCキーを押します。WRITE　INCキーを押す前なら別の数を入れなおして位置の指定を変更することができます。WRITE　INCを押すと指定位置の今点滅しているLEDの表示が上下反転します。上半分の表示は下半分の表示になり、下半分の表示は逆に上半分の表示になります。ゲーム開始時点でコンピュータは乱数によって各LEDに別のLEDを結びつけています。そのため、指定位置のLEDが反転するとともに関係つけられた別のLEDも一緒に反転してしまいます。関係つけのリンクは1本のみで、自分自身を指定しているときもあります。この場合には他のLEDは反転しません。関係つけの方向は一方向のみです。たとえばLED1を指定したときにLED3が同時に反転したとしても、次にLED3を指定したときにLED1も同時に反転するとは限りません。逆にLED3はLED6からも関係つけられているかもしれません。その場合はLED1を指定しても、LED6を指定してもそれぞれ同時にLED3が反転することになります。

首尾良く全部の表示が下半分になると、全部の表示が点滅し2秒後に、それまでの試行回数が表示されます。さらに2秒たつと最初からゲームが再スタートします。関係つけによってはエンドレスになってしまってゲームが終了しない場合もあります。そんなときはリセットしてもう一度8000番地からRUNしてください。

プログラムの説明

このプログラムでは各LEDにリンクされているLEDの番号を格納する場所としてLINKBF（8バイト）を用意しています。はじめに各LED位置毎にRND3サブルーチンによって1～8（実際には0～7）を求めて対応するLINKBF以降の8バイトに順に数を割り当てて行きます。

KEY:ではLED表示静止の状態でキー入力を待つため、KEYIN1（＄0216）をCAllしていますが、BLNK0:以下の部分では表示をブリンクさせながらキーの入力をチェックしなければならないため、KEYIN2（＄0223）をCALLしています。

```
2016/4/20  11:51  flipflp8.txt
END=8104
              ; FLIP-FLOP for NDZ
              ; 03/04/29 4/30 5/1 5/3 5/5 5/10
              ;10/6/1 for ND80Z3
              ;16/4/16 for ND8080
              ;4/17
              ;4/19
              ;4/20
              ;
                   ORG $8000
              ;     LINKBF=$E800
              ;     RNDDT=$E810;$E811
                   LED1=$83F8;$FFF8
                   DP1=$83F4;$FFF4
                   DP3=$83F6;$FFF6
                   CNTR=$83F7;$FFF7;=DP4
                R=$FFD2
              ;
                   DTDPS=$01C0;$05C0
                   KEYIN1=$0216;$0616
                   KEYIN2=$0223;$0623
                   D1=$02DD;4.684msec
              ;
8000 C30E80      JMP START
8003 00       LINKBF:NOP
8004 00          NOP
8005 00          NOP
8006 00          NOP
8007 00          NOP
8008 00          NOP
8009 00          NOP
```

```
800A 00          NOP
800B 00       RNDDT:NOP
800C 00          NOP
800D 00       RNDDT2:NOP
              ;all"UP" disp
800E 3AD2FF   START:LDA R
8011 210B80      LXI H,RNDDT
8014 77          MOV M,A
8015 23          INX H
8016 77          MOV M,A
8017 3E63        MVI A,63;"UP"
8019 0608          MVI B,08
801B 21F883        LXI H,LED1
801E 77       UPDP1:MOV M,A
801F 23            INX H
              ;?       DJNZ UPDP1
8020 05          DCR B;
8021 C21E80      JNZ UPDP1;
8024 210000        LXI H,$0000
8027 22F683        SHLD DP3;DECIMAL COUNTER clear
802A 22F483        SHLD DP1
              ;LINK set
802D 0608          MVI B,08
802F 210380        LXI H,LINKBF
8032 CDC480   LNKST1:CALL RND3
              ;     ANI 07
8035 77            MOV M,A
8036 23            INX H
              ;?       DJNZ LNKST1
8037 05          DCR B;
8038 C23280      JNZ LNKST1;
              ;key entry
803B CD1602   KEY:CALL KEYIN1
803E FE09          CPI 09
8040 D23B80        JNC KEY
8043 3D            DCR A
8044 FA3B80        JM KEY
              ;LED blink and wait WINC in
8047 21F883   BLNK0:LXI H,LED1
804A 85            ADD L
804B 6F            MOV L,A
804C 56            MOV D,M;LED data save
804D 42            MOV B,D
804E 0E00          MVI C,00
8050 71       BLNK1:MOV M,C
8051 CDF780        CALL TM01S
8054 D5          PUSH D
8055 C5          PUSH B
8056 CD2302        CALL KEYIN2
8059 C1          POP B
805A D1          POP D
805B FE09          CPI 09
805D D26880        JNC BLNK2
8060 3D            DCR A
8061 FA6D80        JM BLNK3
8064 72            MOV M,D
8065 C34780        JMP BLNK0
8068 FE15     BLNK2:CPI 15;WRITE INC
```

```
806A CA7380        JZ REVRS
806D 79        BLNK3:MOV A,C
806E 48             MOV C,B
806F 47             MOV B,A
8070 C35080         JMP BLNK1
               ;LED reverse
8073 72        REVRS:MOV M,D
8074 3E63           MVI A,63;"UP"
8076 BE             CMP M
8077 C27C80         JNZ REVRS1
807A 3E5C           MVI A,5C;"DOWN"
807C 77        REVRS1:MOV M,A
               ;link LED reverse
807D 7D             MOV A,L
807E E607           ANI 07
8080 57             MOV D,A; No. save
8081 210380         LXI H,LINKBF
8084 85             ADD L
8085 6F             MOV L,A
8086 7E             MOV A,M
8087 BA             CMP D
8088 CA9980         JZ REVRS3;same No.(no LINK)
808B 21F883         LXI H,LED1
808E 85             ADD L
808F 6F             MOV L,A
8090 3E63           MVI A,63;"UP"
8092 BE             CMP M
8093 C29880         JNZ REVRS2
8096 3E5C           MVI A,5C;"DOWN"
8098 77        REVRS2:MOV M,A
8099 3AF783    REVRS3:LDA CNTR;DECIMAL COUNTER up
809C C601           ADI 01
809E 27             DAA
809F 32F783         STA CNTR
               ;check
80A2 21F883         LXI H,LED1
80A5 0608           MVI B,08
80A7 3E5C           MVI A,5C;"DOWN"
80A9 BE        CK1:CMP M
80AA C23B80         JNZ KEY;not success
80AD 23             INX H
               ;?      DJNZ *CK1
80AE 05           DCR B;
80AF C2A980       JNZ CK1;
               ;OK!  ALL LED blink and COUNTER disp
80B2 CDE280         CALL ALBLNK
80B5 CDC001         CALL DTDPS
80B8 0614           MVI B,14;=20 ,2sec wait
80BA CDF780    WAIT:CALL TM01S
               ;?      DJNZ WAIT
80BD 05           DCR B;
80BE C2BA80       JNZ WAIT;
80C1 C30E80         JMP START
               ;
               ;ransu
80C4 E5        RND3:PUSH H
80C5 D5             PUSH D
80C6 2A0B80    RND2:LHLD RNDDT
```

```
80C9 23          INX H
80CA 7C          MOV A,H
80CB E603        ANI 03
80CD 67          MOV H,A
80CE 220B80      SHLD RNDDT
80D1 3A0D80      LDA RNDDT2
80D4 57          MOV D,A
80D5 7E            MOV A,M
80D6 E607        ANI 07
80D8 BA            CMP D
80D9 CAC680      JZ RND2
80DC 320D80      STA RNDDT2
80DF D1            POP D
80E0 E1            POP H
80E1 C9            RET
                 ;
                 ;LED all blink
80E2 060A      ALBLNK:MVI B,0A;=10
80E4 3EEF      ALBLNK1:MVI A,EF;LED OFF
80E6 D398        OUT 98
80E8 CDF780        CALL TM01S
80EB 3EFF          MVI A,FF;LED ON
80ED D398        OUT 98
80EF CDF780        CALL TM01S
                 ;?      DJNZ *ALBLNK1
80F2 05          DCR B;
80F3 C2E480      JNZ ALBLNK1;
80F6 C9            RET
                 ;
                 ;0.1sec timer
80F7 D5        TM01S:PUSH D
80F8 C5          PUSH B
80F9 0615        MVI B,15;=21
                 TM01S2:;CALL TM1M
80FB CDDD02      CALL D1;4.684ms
80FE 05            DCR B
80FF C2FB80        JNZ TM01S2
8102 C1          POP B
8103 D1            POP D
8104 C9            RET
ALBLNK       =80E2  ALBLNK1      =80E4  BLNK0        =8047
BLNK1        =8050  BLNK2        =8068  BLNK3        =806D
CK1          =80A9  CNTR         =83F7  D1           =02DD
DP1          =83F4  DP3          =83F6  DTDPS        =01C0
KEY          =803B  KEYIN1       =0216  KEYIN2       =0223
LED1         =83F8  LINKBF       =8003  LNKST1       =8032
R            =FFD2  REVRS        =8073  REVRS1       =807C
REVRS2       =8098  REVRS3       =8099  RND2         =80C6
RND3         =80C4  RNDDT        =800B  RNDDT2       =800D
START        =800E  TM01S        =80F7  TM01S2       =80FB
UPDP1        =801E  WAIT         =80BA
```

プログラム№5　SWAP（SWAP8．BTK）

ゲームの説明

　8000番地からRUNするとLEDの左3桁に”8”の下半分が表示されその右に1桁のブランクがあってさらに右3桁には”8”の上半分が表示されます。右端の1桁はブランクでこの桁は使用しません。

　ルールにしたがって表示を移動し、左３桁の表示と右３桁の表示を上下入れ替えるのがゲームの目的です。ルールは次の通りです。

1）ブランクの隣の表示はブランクと位置を交代することができます。

2）隣に異なる表示があるとき、その表示をはさんでさらにその隣がブランクならば、間の表示を飛び越してブランクと表示位置を交代することができます。たとえば表示が[下][上][ブランク]とならんでいる時、[下]のLED番号をKEY入力すると、[ブランク][上][下]に表示が入れ替わります。ルールに合わない表示を指定しても無視されます。

　入れ替えたいLED位置を示す数をKEY入力するとその位置のLEDが点滅します。それでよければWRITE　INCキーを押します。WRITE　INCキーを押す前なら別の数を入れなおして位置の指定を変更することができます。WRITE　INCを押すとルールにしたがって表示とブランクが入れ替わります。ルールに合わない場合には点滅は続きますが入れ替えは行われません。

　うまく全部の入れ替えに成功すると2秒間表示が点滅し、さらに2秒間試行回数が表示されたあと、最初に戻って再びゲームが開始されます。

プログラムの説明

　このプログラムではシステムのワークエリア以外は使用しません。LED表示はLED表示用のアドレスをそのまま使用しています。また乱数も使いません。その割りには結構ややこしいプログラムです。こういうプログラムが得意な人と不得意な人がでてきます。いかにもコンピュータ的なプログラムです。ややこしいのは、場合分けが何通りもあるからです。こういう作業が苦手なプログラマは何か別のうまい方法を考えようとします。それはそれでよいので、こう書かなければならないというきまりはありません。こういうところでプログラマの個性が出て来ます。

　私はというと割りと得意というか、苦にならない方なので、こんな感じに書いてしまいます。

　場合分けはまずキー入力されたLED位置の左の状態を確認して、それから右を確認します。右から見ていっても構いません（たまたま私は左利きなのでどうしても左が先になってしまいます）。

　順に確認します。CHECK：のルーチンです。

①入力した番号のLEDは左端か？これはLEDアドレスの下位3ビットが0であることで確認できます（LED1＝＄FFF8）。このときは右側のチェックに行きます。

②1 つ左はブランクか？左のLEDアドレスのデータが00ならブランクです。置換えルーチンへ行きます。

③1つ左は自分（KEY入力したLEDアドレス）と同じ表示データか？同じ場合は右側のチェックに行きます。

④1つ左のLEDは左端か？これは①と同じ方法でチェックできます。この場合は右側のチェックに行きます。

⑤もう１つ左はブランクか？この場合は置き換えルーチンへ行きます。ブランクでない場合は左側のチェックは終了です。右側のチェックに行きます。

　右側のチェックの説明は省略しますが、やっていることは同じことです。

```
2016/4/18  14:8  swap8.txt
END=810D
                ; SWAP for NDZ
                ; 03/04/29 4/30 5/2 5/3 5/5 5/10
                ;10/6/2 for ND80Z3
                ;16/4/18 for nd8080
                ;
                     ORG $8000
                     LED1=$83F8;$FFF8
                     DP1=$83F4;$FFF4
                     DP3=$83F6;$FFF6
                     CNTR=$83F7;$FFF7;=DP4
                     DTDPS=$01C0;$05C0
                     KEYIN1=$0216;$0616
                     KEYIN2=$0223;$0623
                   D1=$02DD;4.684sec
                ;
                ;start data disp
8000 21F280     START:LXI H,STDT
8003 11F883           LXI D,LED1
```

```
8006 010800         LXI B,$0008
                    ;?      LDIR
8009 CD0181       CALL LDIR;
800C 210000         LXI H,$0000
800F 22F683         SHLD DP3;DECIMAL COUNTER clear
8012 22F483         SHLD DP1
                    ;key entry
8015 CDBF80     KEY:CALL KEYIN1S;1-7 input
8018 FE08           CPI 08
801A D21580         JNC KEY
801D 3D             DCR A
801E FA1580         JM KEY
                    ;LED blink and wait WINC in
8021 21F883     BLNK0:LXI H,LED1
8024 85             ADD L
8025 6F             MOV L,A
8026 7E             MOV A,M
8027 B7             ORA A
8028 CA1580         JZ KEY;space position
802B 57             MOV D,A;LED data save
802C 47             MOV B,A
802D 0E00           MVI C,00
802F 71         BLNK1:MOV M,C
8030 CDE480         CALL TM01S
8033 CDC780         CALL KEYIN2S;1-7,15 input
8036 FE08           CPI 08
8038 D24380         JNC BLNK2
803B 3D             DCR A
803C FA4880         JM BLNK3
803F 72             MOV M,D
8040 C32180         JMP BLNK0
8043 FE15       BLNK2:CPI 15;WRITE INC
8045 CA4E80         JZ CHECK
8048 79         BLNK3:MOV A,C
8049 48             MOV C,B
804A 47             MOV B,A
804B C32F80         JMP BLNK1
                    ;check
804E 7D         CHECK:MOV A,L
804F E5             PUSH H;save HL(current position)
8050 E607           ANI 07
8052 CA6B80         JZ CHECKR;position=LED1
8055 2B             DCX H
8056 7E             MOV A,M
8057 B7             ORA A
8058 CA8980         JZ REP;left=space
805B BA             CMP D
805C CA6B80         JZ CHECKR;left=same
805F 7D             MOV A,L
8060 E607           ANI 07
8062 CA6B80         JZ CHECKR;left=LED1
8065 2B             DCX H
8066 7E             MOV A,M
8067 B7             ORA A
8068 CA8980         JZ REP;left of left=space
806B E1         CHECKR:POP H
806C E5             PUSH H
806D 7D             MOV A,L
```

```
806E FEFE          CPI FE
8070 CABB80        JZ CANT;position=LED7,cannot replace
8073 23            INX H
8074 7E            MOV A,M
8075 B7            ORA A
8076 CA8980        JZ REP;right=space
8079 BA            CMP D
807A CABB80        JZ CANT;right=same,cannot replace
807D 7D            MOV A,L
807E FEFE          CPI FE
8080 CABB80        JZ CANT;right=LED7,cannot replace
8083 23            INX H
8084 7E            MOV A,M
8085 B7            ORA A
8086 C2BB80        JNZ CANT;right of right is not space,cannot replace
                  ;replace
8089 72        REP:MOV M,D;set current DATA to space position
808A E1            POP H
808B 3600          MVI M,00;set space to current position
808D 3AF783        LDA CNTR;DECIMAL COUNTER up
8090 C601          ADI 01
8092 27            DAA
8093 32F783        STA CNTR
                  ;all check
8096 21F883        LXI H,LED1
8099 11FA80        LXI D,ENDDT
809C 0607          MVI B,07
809E 1A        ALCHK1:LDAX D
809F BE            CMP M
80A0 C21580        JNZ KEY;not success
80A3 23            INX H
80A4 13            INX D
                  ;?      DJNZ *ALCHK1
80A5 05            DCR B;
80A6 C29E80        JNZ ALCHK1;
                  ;OK! ALL LED blink and COUNTER disp
80A9 CDCF80        CALL ALBLNK
80AC CDC001        CALL DTDPS
80AF 0614          MVI B,14;=20 ,2sec wait
80B1 CDE480    WAIT:CALL TM01S
                  ;?      DJNZ WAIT
80B4 05            DCR B;
80B5 C2B180        JNZ WAIT;
80B8 C30080        JMP START
                  ;cannot replace
80BB E1        CANT:POP H
80BC C34880        JMP BLNK3
                  ;
                  ;KEYIN with DE,BC save
80BF C5        KEYIN1S:PUSH B
80C0 D5            PUSH D
80C1 CD1602        CALL KEYIN1
80C4 D1            POP D
80C5 C1            POP B
80C6 C9            RET
80C7 C5        KEYIN2S:PUSH B
80C8 D5            PUSH D
80C9 CD2302        CALL KEYIN2
```

```
80CC D1          POP D
80CD C1          POP B
80CE C9          RET
                 ;
                 ;LED all blink
80CF 060A        ALBLNK:MVI B,0A;=10
80D1 3EEF        ALBLNK1:MVI A,EF
80D3 D398          OUT 98
80D5 CDE480          CALL TM01S
80D8 3EFF          MVI A,FF
80DA D398          OUT 98
80DC CDE480          CALL TM01S
                 ;?      DJNZ *ALBLNK1
80DF 05            DCR B;
80E0 C2D180        JNZ ALBLNK1;
80E3 C9              RET
                 ;
                 ;0.1sec timer
80E4 D5          TM01S:PUSH D
80E5 1E15          MVI E,15;=21
80E7 D5          TM01S2:PUSH D
80E8 CDDD02        CALL D1;4.684ms
80EB D1            POP D
80EC 1D              DCR E
80ED C2E780          JNZ TM01S2
80F0 D1              POP D
80F1 C9              RET
                 ;
80F2 5C          STDT:DB 5C
80F3 5C               DB 5C
80F4 5C               DB 5C
80F5 00               DB 00
80F6 63               DB 63
80F7 63               DB 63
80F8 63               DB 63
80F9 00               DB 00
                 ;
80FA 63          ENDDT:DB 63
80FB 63               DB 63
80FC 63               DB 63
80FD 00               DB 00
80FE 5C               DB 5C
80FF 5C               DB 5C
8100 5C               DB 5C
                 ;
8101 F5          LDIR:PUSH PSW
8102 7E          LDIR2:MOV A,M
8103 12            STAX D
8104 23            INX H
8105 13            INX D
8106 0B            DCX B
8107 78            MOV A,B
8108 B1            ORA C
8109 C20281        JNZ LDIR2
810C F1            POP PSW
810D C9            RET
                 ;
ALBLNK       =80CF  ALBLNK1      =80D1  ALCHK1       =809E
```

BLNK0	=8021	BLNK1	=802F	BLNK2	=8043
BLNK3	=8048	CANT	=80BB	CHECK	=804E
CHECKR	=806B	CNTR	=83F7	D1	=02DD
DP1	=83F4	DP3	=83F6	DTDPS	=01C0
ENDDT	=80FA	KEY	=8015	KEYIN1	=0216
KEYIN1S	=80BF	KEYIN2	=0223	KEYIN2S	=80C7
LDIR	=8101	LDIR2	=8102	LED1	=83F8
REP	=8089	START	=8000	STDT	=80F2
TM01S	=80E4	TM01S2	=80E7	WAIT	=80B1

プログラム№6　CATCH（CATCH8．BTK）

ゲームの説明

これはコンピュータ相手の鬼ごっこ（正確には追いかけっこ）です。

8000番地からRUNするとLEDのどこかに上向きのマーク（CUP）と下向きのマーク（CAP）がそれぞれ1個表示されます。CUPがプレーヤーでCAPがコンピュータです。最初の表示位置は乱数で決定します。CUPが点滅してKEY入力待ちであることを示します。プレーヤーはCUPを右に動かすか、左に動かすかを決めてKEY入力します。左に移動するなら[0]を、右に移動するなら[WRITE　INC]を押します。移動量はコンピュータが乱数で求めます。CUPが移動して、CAPと同じ位置で止まれば、プレーヤーの勝ちです。ピーと音がしてCAPが消滅します。CAPと同じ位置で止まらなかったときはCAPが移動する番です。コンピュータがCAPをどちらにいくつ移動するかを乱数で求めてそのとおりに移動します。移動の結果CUPと同じ位置で止まるとコンピュータの勝ちです。ブーと音がしてCUPが消滅します。どちらかが消滅するとゲームは終りです。2秒たつと始めに戻って再びゲームが開始されます。

なおゲーム開始時にCUPとCAPが同じ位置に表示されることがあります。この場合プレーヤーはまずCAPから逃げることから開始することになります。またLEDの左端と右端はリング状につながっていることとしています。左端からさらに左に移動したマークは右端から出現します。逆に右端からさらに右に移動したマークは左端から現れます。

プログラムの説明

作業用のメモリとしてMYADRS（my　address，コンピュータ側の表示位置を記憶）とYRADRS（your　address，プレーヤーの表示位置を記憶）の各1バイトとRNDDT、RNDDT2（乱数用）を使います。それだけで済むのですから簡単かというと処理はかなり複雑です。動作のポイントはプレーヤーもコンピュータも左か右に指定しただけ移動することと、移動後に相手の位置を確認して、キャッチできたかどうかを判定することです。

ここで、うーむと考えると、どうもコンピュータとプレーヤーは表示マークが違うだけで、同じプログラムで動かせるのではないか、という気がします。ここが一番重要なポイントです。

そう考えた結果2つのサブルーチンを作りました。DTSETとSTEPです。

DTSETはゲーム開始時のCUPとCAPの位置を乱数で選び、その位置情報をメモリに入れるとともに、それぞれのマークを表示します。Eレジスタに表示マークを、BCレジスタに位置情報を格納するメモリアドレスを入れてDTSETをCALLするようにできています。このようにすることで、サブルーチンは与えられた情報の内容に関係無くまさに「機械的」に処理をすることで目的の動作をしてしまいます。

こういうサブルーチン（場合によってはプログラム全体がこういうつくりになっているものもある）の特徴はそのサブルーチンだけを見ると、何をやっているのかさっぱりわからないということにあります。CLRとDTSETを比べてみるとそのことがよくわかると思います。ではDTSETは悪いプログラムなのかというと、むしろDTSETの方がよりコンピュータ的で、より望ましいプログラムなのだと言えます（まああまりこだわって無理にしなくても、基本的には好きなように書けばよいわけですけれど）。

たとえば良く似た処理を10回個々にプログラムするよりは何とか工夫して同じサブルーチンをCALLするようにした方がすっきりまとまります。これが100回となるとこれはもう絶対にそうすべきです。こうすることの最大のメリットはとにかくプログラムが短くまとまる（ということはプログラム自体がフローチャート風になって、分かりやすくなる）ことです。ことのついでですが、プロは滅多にフローチャートは書きません（私だけか？！）。たいていはひとりで始めから終わりまでひとつのプログラムを書くことになるので、その場合、ここで紹介している程度のプログラムなら、フローチャートは必要ありません。やるべきことを、たとえば上で説明したゲームのルールをざっと読んで、理解すると、あとは頭の中であああやってこうやってと自然にプログラムの流れが出来てきます。あとはいきなりパソコンに向かってアセンブラのニーモニックを打ち込みはじめます（何と原始的な！）。

ちょっと長いプログラムなら使いたいワークエリアのアドレスとか名前をメモしたり、処理の概略チャートをメモ風にかくこともあります。しかし、教科書にのっているようなチャートを書いているよりも直接プログラムを書いた方が速いのです！

プログラムに慣れてくると、少なくとも自分で書いたプログラムならば、チャートなどなくても、プログラムリストを見れば何をやっているのか分かります。あとで分からなくなるかなあと思ったときは、必要と思われるコメントを書き込んでおけば十分です。

これはあくまで個人で仕事をする場合の話で、グループで作業をする場合には、チャートが必要なこともあると思います（何事もケースバイケースです）。ともあれ私個人としてはチャートは面倒ですから、書きません。このプログラム集のプログラムに書き込んでいるコメントも大半は読者向けのもので自分用には殆ど不要です（実際、たった少しのコメントをプログラム中に書きこむのさえ、怠惰なプログラマにとっては面倒なのです）。

さて余計なことを書きすぎました。もうひとつサブルーチン化のメリットがあります。それはチェックや修正が容易な点です。プログラムは書いていきなり完成ということはまずありません。こんな短いプログラムでもそうです。ここで紹介したプログラムも一度では終っていません。リストの最初に日付が覚えとして書いてあります。何回も訂正していることが分かります。こういうときにサブルーチンになっていると、そこだけ直せば済んでしまいます。同じような処理を何回もだらだら書いたプログラムでは、バグ修正が大変です。

サブルーチン化にはデメリットもあります。はじめに書いたように、そこだけ読んでもなにをやっているのかよくわからないという点です。この点についてはたとえばプリントアウトしたリストに処理されるデータの１つを実際に書きこむなどして、プログラムをあらためてながめてみると、流れが見えてきます。

と無理矢理納得していただいたところでもうひとつのサブルーチンSTEPについて説明します。こちらはちょっと複雑です。サブルーチンに与えるべきデータ(これをパラメータとか引数とかといいます)がちょっと多くて分かりづらいかもしれません。

現在自分のマークが表示されているLEDアドレス(HLレジスタ)、右へ行くのか左へ行くのか(Cレジスタ)、移動後に現在の表示を消すためのマスク(同じ場所に相手も表示している場合を考えて、自分だけを消すようにする。Dレジスタ)、移動後に表示するマーク(Eレジスタ)、移動後の位置情報を格納するメモリアドレス(もとはIXレジスタを使っていたが８０８０にはないためそれに代わるワークエリアとしてADRSWKを使う)、とこれだけのデータをサブルーチンに与えます。データが沢山あってなんだかややこしそうですがこれらのデータをサブルーチンのリストの各レジスタのところにメモなどして、あらためてながめてみると、それほど複雑ではないことがわかると思います。

このサブルーチンで特にコメントしたいのがCレジスタの値です。これがテクニックです。左に行くか右に行くかをFFか０１で示しています。どこがテクニックか？

左に行くということは現在表示しているLEDのアドレス(８３F８～８３FFのいずれか)から－１することです。右に行くのは逆にアドレスを＋１することです。－１はDCR、＋１はINRを普通は使います。あるいはSUI　０１かADI　０１を使います。ここから先はわからない人にはわかりません。

突然無関係のように登場したFFですが、FFを１０進に直すと２５５です。これではわかりません。FFにはもうひとつの意味があります。FFを符号付の数として１０進に直すと－１なのです！これは約束事なのですがでたらめに決めたのではなくて計算上の根拠があっての話なのです。SUI　０１の代わりにADI　FFとしても不思議なことに結果は同じになってしまうのです(フラグは異なります)。どーしても納得ができない人は試してみてください。

このことを利用して本来加算と減算にしなければならないはずのところを同じADD　Cで切りぬけているのです。もうひとつ、加算後にORI　F８としている部分です。これによって加算後に結果がオーバフロー(下位アドレスがF８～FFの範囲を超えてしまう)してもリング状に左端と右端がつながったように結果が計算されます。これも納得できない人は試してみてください。

ことのついでにMYTURN:のところでRNDの結果から、右と左を示す０１とFFをさらりと作り出しているところに注目してください。自画自賛ですがこのようにきれいに決まるとヤッターという気になって心が踊ります。

```
2016/4/20  11:51  catch8.txt
END=8169
                ; CATCH for NDZ
                ; 03/05/06 5/10
                ;10/6/2 for ND80Z3
                ;16/4/15 for ND8080
                ;16/4/15 timer
                ;4/18
                ;4/19
                ;4/20
                ;
                     ORG $8000
                ;     MYADRS=$E800
                ;     YRADRS=$E801
                ;     RNDDT=$E810:$E811
                     LED1=$83F8;$FFF8
                   R=$FFD2
                ;
                     KEYIN2=$0223;$0623
                     D1=$02DD;4.684msec
                ;
8000 C30A80        JMP START
8003 00         MYADRS:NOP
8004 00         YRADRS:NOP
8005 00         ADRSWK:NOP
8006 00            NOP
8007 00         RNDDT:NOP
8008 00            NOP
8009 00         RNDDT2:NOP
```

```
            ;data set
800A 3AD2FF START:LDA R
800D 210780    LXI H,RNDDT
8010 77        MOV M,A
8011 23        INX H
8012 77        MOV M,A
8013 CDD080    CALL CLR
            ;my position select
8016 1E23       MVI E,23;cap
8018 010380     LXI B,MYADRS
801B CDDD80     CALL DTSET
            ;your posision select
801E 1E1C       MVI E,1C;cup
8020 03         INX B
8021 CDDD80     CALL DTSET
            ;GAME START
            ;your turn
8024 3A0480 REENT:LDA YRADRS
8027 21F883     LXI H,LED1
802A 85         ADD L
802B 6F         MOV L,A
802C 56         MOV D,M;LED data save
802D 42         MOV B,D
802E 0E00       MVI C,00
            ;blink and keyin
8030 71     KEY:MOV M,C
8031 CD5C81     CALL TM01S
8034 CD1081     CALL KEYIN2S
8037 B7         ORA A
8038 CA4680     JZ LEFT
803B FE15       CPI 15;W INC
803D CA4B80     JZ RIGHT
8040 59         MOV E,C
8041 48         MOV C,B
8042 43         MOV B,E
8043 C33080     JMP KEY
8046 0EFF   LEFT:MVI C,FF
8048 C34D80     JMP RIGHT2
804B 0E01   RIGHT:MVI C,01
804D 111CE3 RIGHT2:LXI D,$E31C;see STEP comment line
            ;?      LD IX,YRADRS
8050 E5        PUSH H;
8051 210480    LXI H,YRADRS;
8054 220580    SHLD ADRSWK;
8057 E1        POP H;
8058 CDEA80      CALL STEP
            ;catch?
            ;?      DEC IX;MYADRS
            ;?      CP (IX+00)
805B E5        PUSH H;
805C 2A0580    LHLD ADRSWK;
805F 2B        DCX H;
8060 220580    SHLD ADRSWK;
8063 BE        CMP M;
8064 E1        POP H;
8065 C27180      JNZ MYTURN
            ;catch!
8068 7E         MOV A,M
```

```
8069 E6DC          ANI DC;cap clear
806B 77            MOV M,A
806C 3E17          MVI A,17;piiii
806E C39D80        JMP END
               ;my turn
8071 CDB280    MYTURN:CALL RND3;LEFT(FF) or RIGHT(01) select
8074 E601          ANI 01
8076 C27A80        JNZ MYTURN1;=01
8079 3D            DCR A;=FF
807A 4F        MYTURN1:MOV C,A
807B 2A0580      LHLD ADRSWK
807E 7E            MOV A,M
807F 21F883        LXI H,LED1
8082 85            ADD L
8083 6F            MOV L,A
8084 1123DC        LXI D,$DC23;see STEP comment line
8087 CDEA80        CALL STEP
               ;catch?
               ;?      INC IX;YRADRS
               ;?      CP (IX+00)
808A E5          PUSH H;
808B 2A0580      LHLD ADRSWK;
808E 23          INX H;
808F 220580      SHLD ADRSWK;
8092 BE          CMP M;
8093 E1          POP H;
8094 C22480        JNZ REENT
               ;catch!
8097 7E            MOV A,M
8098 E6E3          ANI E3;cup clear
809A 77            MOV M,A
809B 3E00          MVI A,00;buuuu
809D 0605      END:MVI B,05
809F CD1881    END1:CALL SOUND
               ;?      DJNZ *END1
80A2 05          DCR B;
80A3 C29F80      JNZ END1;
80A6 0614          MVI B,14;=20 ,2sec wait
80A8 CD5C81    WAIT:CALL TM01S
               ;?      DJNZ WAIT
80AB 05          DCR B
80AC C2A880      JNZ WAIT
80AF C30A80        JMP START
               ;
               ;ransu
80B2 E5        RND3:PUSH H
80B3 D5            PUSH D
80B4 2A0780    RND2:LHLD RNDDT
80B7 23          INX H
80B8 7C          MOV A,H
80B9 E603        ANI 03
80BB 67          MOV H,A
80BC 220780      SHLD RNDDT
80BF 3A0980      LDA RNDDT2
80C2 57          MOV D,A
80C3 7E            MOV A,M
80C4 E607        ANI 07
80C6 BA            CMP D
```

```
80C7 CAB480      JZ RND2
80CA 320980      STA RNDDT2
80CD D1            POP D
80CE E1            POP H
80CF C9            RET
                 ;
                 ;LED all clear
80D0 21F883    CLR:LXI H,LED1
80D3 AF            XRA A
80D4 0608          MVI B,08
80D6 77        CLR1:MOV M,A
80D7 23            INX H
                 ;?      DJNZ *CLR1
80D8 05          DCR B;
80D9 C2D680      JNZ CLR1;
80DC C9            RET
                 ;
                 ;start data set;BC=MYADRS or YRADRS;E=disp data 23(cap) or 1C(cup)
80DD CDB280    DTSET:CALL RND3
                 ;     ANI 07
80E0 02            STAX B
80E1 21F883        LXI H,LED1
80E4 85            ADD L
80E5 6F            MOV L,A
80E6 7B            MOV A,E
80E7 B6            ORA M
80E8 77            MOV M,A;cap or cup or both disp
80E9 C9            RET
                 ;
                 ;step left or right    C=FF(left) C=01(right),HL=current position
                 ;                      D=clear mask DC(cap) or E3(cup)
                 ;                      E=disp data 23(cap) or 1C(cup)
                 ;                      IX=MYADRS or YRADRS
80EA CDB280    STEP:CALL RND3
                 ;     ANI 07
80ED 3C            INR A
80EE FE07          CPI 07
80F0 D2EA80        JNC STEP
80F3 47            MOV B,A
80F4 CD5981    STEP1:CALL TM02S
80F7 7E            MOV A,M
80F8 A2            ANA D;clear
80F9 77            MOV M,A
80FA 7D            MOV A,L
80FB 81            ADD C
80FC F6F8          ORI F8
80FE 6F            MOV L,A
80FF 7E            MOV A,M
8100 B3            ORA E;disp
8101 77            MOV M,A
                 ;?      DJNZ STEP1
8102 05          DCR B;
8103 C2F480      JNZ STEP1;
8106 7D            MOV A,L
8107 E607          ANI 07
                 ;     STA IX+00
8109 E5          PUSH H;
810A 2A0580      LHLD ADRSWK;
```

```
810D 77          MOV M,A;
810E E1          POP H;
810F C9            RET
                 ;
                 ;KEYIN with DE,BC save
8110 C5          KEYIN2S:PUSH B
8111 D5             PUSH D
8112 CD2302         CALL KEYIN2
8115 D1             POP D
8116 C1             POP B
8117 C9             RET
                 ;
                 ;
8118 F5          SOUND:PUSH PSW
8119 E5             PUSH H
811A D5             PUSH D
811B C5             PUSH B
811C 214181         LXI H,SNDTBL
811F 85             ADD L
8120 6F             MOV L,A
8121 46             MOV B,M
8122 1E1A           MVI E,1A
8124 50          SNDS1:MOV D,B
8125 3EFF           MVI A,FF;SP OUT=H
8127 D398           OUT 98
8129 7F          SNDS2:MOV A,A;5---------
812A 15             DCR D;5          | 5+5+10=20  20/2=10microsec
812B C22981         JNZ SNDS2;10----
812E 50             MOV D,B
812F 3EDF           MVI A,DF;sp out=L
8131 D398           OUT 98
8133 7F          SNDS3:MOV A,A;5---------
8134 15             DCR D;5          | 5+5+10=20  20/2=10microsec
8135 C23381         JNZ SNDS3;10----
8138 1D             DCR E
8139 C22481         JNZ SNDS1
813C C1             POP B
813D D1             POP D
813E E1             POP H
813F F1             POP PSW
8140 C9             RET
                 ; SOUND TABLE
8141 7F          SNDTBL:DB 7F;so4
8142 77             DB 77;so#4
8143 71             DB 71;ra4
8144 6A             DB 6A;ra#4
8145 5F             DB 5F;do5
8146 59             DB 59;do#5
8147 54             DB 54;re5
8148 4F             DB 4F;re#
8149 47             DB 47;fA5
814A 43             DB 43;fa#5
814B 3F             DB 3F;so5
814C 3B             DB 3B;so5#
814D 35             DB 35;ra#5
814E 32             DB 32;si5
814F 2F             DB 2F;do6
8150 2C             DB 2C;do#6
```

```
8151 25          DB 25;mi6
8152 27          DB 27;re#6
8153 2A          DB 2A;re6
8154 4B          DB 4B;mi5
8155 38          DB 38;ra5
8156 64          DB 64;si4
8157 23          DB 23;fa6
8158 21          DB 21;fa#6
                 ;
                 ;0.2sec timer
8159 CD5C81      TM02S:CALL TM01S
                 ;0.1sec timer
815C D5          TM01S:PUSH D
815D 1E15          MVI E,15;=21
815F D5          TM01S2:PUSH D
8160 CDDD02        CALL D1;4.684ms
8163 D1            POP D
8164 1D              DCR E
8165 C25F81          JNZ TM01S2
8168 D1              POP D
8169 C9              RET
                 ;
ADRSWK       =8005  CLR          =80D0  CLR1         =80D6
D1           =02DD  DTSET        =80DD  END          =809D
END1         =809F  KEY          =8030  KEYIN2       =0223
KEYIN2S      =8110  LED1         =83F8  LEFT         =8046
MYADRS       =8003  MYTURN       =8071  MYTURN1      =807A
R            =FFD2  REENT        =8024  RIGHT        =804B
RIGHT2       =804D  RND2         =80B4  RND3         =80B2
RNDDT        =8007  RNDDT2       =8009  SNDS1        =8124
SNDS2        =8129  SNDS3        =8133  SNDTBL       =8141
SOUND        =8118  START        =800A  STEP         =80EA
STEP1        =80F4  TM01S        =815C  TM01S2       =815F
TM02S        =8159  WAIT         =80A8  YRADRS       =8004
```

プログラム№7　もぐらたたき（MOGURA8．BTK）

ゲームの説明

LEDにランダムに現れるもぐらをたたくゲームです。

8000番地からRUNするとLEDの全部の表示が”8”の一番下の”__”になります。2秒ごとにLEDのどこかの桁にモグラ（下向きのコの字のパターン）が現れます。1秒たつとモグラは引っ込んでしまいます。その前にうまくモグラをたたくと（表示されているLED位置を示す1～8の数字キーを押すと）ピッと音がしてモグラがつぶれます。1～8のキーを押すと対応するLEDにハンマー（”8”の字の上のバー）が表示され、0．5秒後に下に落とされます。ヒットするのが遅れたり、違う数を押したりすると失敗です。ブーという音がします。

モグラは一度に1匹だけ出て来ます。全部で20匹出るとゲームオーバーです。ヒットしたモグラの数がLEDに表示され約2秒間点滅します。そのあと2秒間表示が静止したあとはじめに戻って再びゲームが開始されます。

プログラムの説明

作業用のメモリとしてRNDDT、RNDDT2（乱数用）の他は1バイトのMCOUNTのみを使います。最初20（14H）から始って0になるまでモグラの表示回数をダウンカウントします。

はじめのMOGURA01：の処理は説明が必要です。KEYからの入力をチェックしていますが1～8が入力されると「ブー」音を出しています。じつはこの時点ではまだモグラは出現していないのです。ここは「お手つき」をチェックしているのです。

HAMMERサブルーチンはハンマーの表示とサウンド（ブーおよびピー）を行います。HAMMERサブルーチンで、ANI 14としているのはモグラをヒットしたとき、モグラがつぶれた形を表示するためです（下向きのコの字の上のバーが消える）。そのあとのORI 08でハンマーが落ちたところを表示しています。このANDとORによってモグラをヒットしたときでも、モグラがいない場所をたたいてしまったときでも同じサブルーチンが使えるようになります。そのあとのサウンド出力も同様でCレジスタの値によって、同じサブルーチンでヒットしたときのピーと失敗したときのブーが出るようになります。

さて失敗したときもヒットしたときもモグラの数は－1されます。0になったらゲーム終了処理へ行き、そうでないときは始めの部分に戻ってモグラの出現処理をくりかえします。失敗してもヒットしても同じ処理を行います。NOT1：の部分です。ここはサブルーチンではありませんがサブルーチンのときと同じ考え方です。NOT：の後半部分になっていますがHIT：の処理のあともここにジャンプしてきます。このように同じ処理にできる部分はCALLかJMPでひとつにまとめるのがプログラムのコツです。

```
2016/4/20  9:53  mogura8.txt
END=816E
                ; MOGURATATAKI for NDZ (HIT'N BLOW)
                ; 03/5/2 5/5 5/11 5/12
                ;10/6/2 for ND80Z3
                ;7/21 9/17
                ;16/4/16 for ND8080
                ;4/18
                ;4/19
                ;4/20
                ;
                     ORG $8000
                ;     RNDDT=$E810;$E811
                ;     MCOUNT=$E812
                     LED1=$83F8;$FFF8
                     DP1=$83F4;$FFF4
                     DP3=$83F6;$FFF6
                     CNTR=$83F7;$FFF7;=DP4
                   R=$FFD2
                ;
                     DTDPS=$01C0;$05C0
                     KEYIN2=$0223;$0623
                ;
8000 C30780        JMP START
8003 00         RNDDT:NOP
8004 00            NOP
8005 00         RNDDT2:NOP
```

```
8006 00       MCOUNT:NOP
              ;start data disp
8007 3AD2FF   START:LDA R
800A 210380      LXI H,RNDDT
800D 77          MOV M,A
800E 23          INX H
800F 77          MOV M,A
8010 210000      LXI H,$0000
8013 22F683        SHLD DP3;DECIMAL COUNTER clear
8016 22F483        SHLD DP1
8019 3E14          MVI A,14;=20
801B 320680        STA MCOUNT;mogura
              ;wait 2sec,if keyin then buuuu
801E CDF280   MOGURA:CALL STDTDP
8021 01D007        LXI B,$07D0;=2000 ,wait 2000msec
8024 CD1F81   MOGURA01:CALL TM1M
8027 CDFF80        CALL KEYIN2S
802A FE09          CPI 09
802C D23B80        JNC MOGURA02
802F 3D            DCR A
8030 FA3B80        JM MOGURA02
8033 0E00          MVI C,00;sound buuuu
8035 CDA080        CALL HAMMER
8038 C31E80        JMP MOGURA
803B 0B       MOGURA02:DCX B
803C 78            MOV A,B
803D B1            ORA C
803E C22480        JNZ MOGURA01
              ;"MOGURA" up
8041 CDBF80   MOGURA1:CALL RND3
              ;     ANI 07
8044 57            MOV D,A;position save
8045 21F883        LXI H,LED1
8048 85            ADD L
8049 6F            MOV L,A
804A 3654          MVI M,54;MOGURA disp
804C 01E803        LXI B,$03E8;=1000 ,wait 1000msec
804F CDFF80   MOGURA2:CALL KEYIN2S
8052 FE09          CPI 09
8054 D26780        JNC NOT
8057 3D            DCR A
8058 FA6780        JM NOT
805B BA            CMP D
805C CA7D80        JZ HIT
              ;BUUUUU
805F 0E00          MVI C,00;sound buuuuu
8061 CDA080        CALL HAMMER
8064 C37080        JMP NOT1
              ;not keyin
8067 CD1F81   NOT:CALL TM1M
806A 0B            DCX B
806B 78            MOV A,B
806C B1            ORA C
806D C24F80        JNZ MOGURA2
8070 3A0680   NOT1:LDA MCOUNT
8073 3D            DCR A
8074 CA8E80        JZ END
8077 320680        STA MCOUNT
```

```
807A C31E80        JMP MOGURA
                ;hit!
807D 0E17       HIT:MVI C,17;sound piiii
807F CDA080        CALL HAMMER
8082 3AF783        LDA CNTR;DECIMAL COUNTER up
8085 C601          ADI 01
8087 27            DAA
8088 32F783        STA CNTR
808B C37080        JMP NOT1
                ;game over! COUNTER disp and ALL LED blink
808E CDC001     END:CALL DTDPS
8091 CDDD80        CALL ALBLNK
8094 0614          MVI B,14;=20 ,2sec wait
8096 CD1381     WAIT:CALL TM01S
                ;?      DJNZ WAIT
8099 05            DCR B;
809A C29680        JNZ WAIT;
809D C30780        JMP START
                ;
                ;hammer disp
80A0 21F883     HAMMER:LXI H,LED1
80A3 85            ADD L
80A4 6F            MOV L,A
80A5 7E            MOV A,M
80A6 F601          ORI 01;hammer up
80A8 77            MOV M,A
80A9 CD0781        CALL TM05S
80AC E614          ANI 14;hammer down
80AE F608          ORI 08
80B0 77            MOV M,A
80B1 79            MOV A,C
80B2 0603          MVI B,03
80B4 CD2E81     HAMMER1:CALL SOUND
                ;?      DJNZ HAMMER1
80B7 05            DCR B;
80B8 C2B480        JNZ HAMMER1;
80BB CD0781        CALL TM05S
80BE C9            RET
                ;
                ;ransu
80BF E5         RND3:PUSH H
80C0 D5            PUSH D
80C1 2A0380     RND2:LHLD RNDDT
80C4 23            INX H
80C5 7C            MOV A,H
80C6 E603          ANI 03
80C8 67            MOV H,A
80C9 220380        SHLD RNDDT
80CC 3A0580        LDA RNDDT2
80CF 57            MOV D,A
80D0 7E            MOV A,M
80D1 E607          ANI 07
80D3 BA            CMP D
80D4 CAC180        JZ RND2
80D7 320580        STA RNDDT2
80DA D1            POP D
80DB E1            POP H
80DC C9            RET
```

```
                ;
                ;LED all blink
80DD 060A       ALBLNK:MVI B,0A;=10
80DF 3EEF       ALBLNK1:MVI A,EF
80E1 D398          OUT 98
80E3 CD1381           CALL TM01S
80E6 3EFF             MVI A,FF
80E8 D398          OUT 98
80EA CD1381           CALL TM01S
                ;?       DJNZ *ALBLNK1
80ED 05            DCR B;
80EE C2DF80        JNZ ALBLNK1;
80F1 C9               RET
                ;
                ;start data disp
80F2 21F883     STDTDP:LXI H,LED1
80F5 0608            MVI B,08
80F7 3608       STDTDP1:MVI M,08
80F9 23              INX H
                ;?       DJNZ STDTDP1
80FA 05            DCR B;
80FB C2F780        JNZ STDTDP1;
80FE C9               RET
                ;
                ;KEYIN with DE,BC save
80FF C5         KEYIN2S:PUSH B
8100 D5            PUSH D
8101 CD2302        CALL KEYIN2
8104 D1            POP D
8105 C1            POP B
8106 C9            RET
                ;
                ;0.5sec timer
8107 C5         TM05S:PUSH B
8108 0605            MVI B,05
810A CD1381     TM05S1:CALL TM01S
                ;?       DJNZ TM05S1
810D 05            DCR B;
810E C20A81        JNZ TM05S1;
8111 C1               POP B
8112 C9               RET
                ;0.1sec timer
8113 D5         TM01S:PUSH D
8114 1E64          MVI E,64;=100
8116 CD1F81     TM01S2:CALL TM1M
8119 1D              DCR E
811A C21681          JNZ TM01S2
811D D1              POP D
811E C9              RET
                ;
                ;1msec timer
811F D5         TM1M:PUSH D;11
8120 1607          MVI D,07; ck=5 (11+7+292*7+10+10)/2.048=2082/2.048=1016
8122 1E12       TM1M2:MVI E,12;=18 ck=7 7+15*18+15=292
8124 1D         TM1M3:DCR E;5
8125 C22481        JNZ TM1M3;10
8128 15            DCR D;5
8129 C22281        JNZ TM1M2;10
```

```
812C D1          POP D;10
812D C9          RET;10
                 ;
812E F5          SOUND:PUSH PSW
812F E5             PUSH H
8130 D5             PUSH D
8131 C5             PUSH B
8132 215781         LXI H,SNDTBL
8135 85             ADD L
8136 6F             MOV L,A
8137 46             MOV B,M
8138 1E1A           MVI E,1A
813A 50          SNDS1:MOV D,B
813B 3EFF           MVI A,FF;SP OUT=H
813D D398           OUT 98
813F 7F          SNDS2:MOV A,A;5---------
8140 15             DCR D;5          | 5+5+10=20  20/2=10microsec
8141 C23F81         JNZ SNDS2;10----
8144 50             MOV D,B
8145 3EDF           MVI A,DF;sp out=L
8147 D398           OUT 98
8149 7F          SNDS3:MOV A,A;5---------
814A 15             DCR D;5          | 5+5+10=20  20/2=10microsec
814B C24981         JNZ SNDS3;10----
814E 1D             DCR E
814F C23A81         JNZ SNDS1
8152 C1             POP B
8153 D1             POP D
8154 E1             POP H
8155 F1             POP PSW
8156 C9             RET
                 ; SOUND TABLE
8157 7F          SNDTBL:DB 7F;so4
8158 77             DB 77;so#4
8159 71             DB 71;ra4
815A 6A             DB 6A;ra#4
815B 5F             DB 5F;do5
815C 59             DB 59;do#5
815D 54             DB 54;re5
815E 4F             DB 4F;re#
815F 47             DB 47;fA5
8160 43             DB 43;fa#5
8161 3F             DB 3F;so5
8162 3B             DB 3B;so5#
8163 35             DB 35;ra#5
8164 32             DB 32;si5
8165 2F             DB 2F;do6
8166 2C             DB 2C;do#6
8167 25             DB 25;mi6
8168 27             DB 27;re#6
8169 2A             DB 2A;re6
816A 4B             DB 4B;mi5
816B 38             DB 38;ra5
816C 64             DB 64;si4
816D 23             DB 23;fa6
816E 21             DB 21;fa#6
                 ;
ALBLNK       =80DD  ALBLNK1      =80DF  CNTR         =83F7
```

DP1	=83F4	DP3	=83F6	DTDPS	=01C0
END	=808E	HAMMER	=80A0	HAMMER1	=80B4
HIT	=807D	KEYIN2	=0223	KEYIN2S	=80FF
LED1	=83F8	MCOUNT	=8006	MOGURA	=801E
MOGURA01	=8024	MOGURA02	=803B	MOGURA1	=8041
MOGURA2	=804F	NOT	=8067	NOT1	=8070
R	=FFD2	RND2	=80C1	RND3	=80BF
RNDDT	=8003	RNDDT2	=8005	SNDS1	=813A
SNDS2	=813F	SNDS3	=8149	SNDTBL	=8157
SOUND	=812E	START	=8007	STDTDP	=80F2
STDTDP1	=80F7	TM01S	=8113	TM01S2	=8116
TM05S	=8107	TM05S1	=810A	TM1M	=811F
TM1M2	=8122	TM1M3	=8124	WAIT	=8096

プログラム№8　神経衰弱（SINKSUI8．BTK）

ゲームの説明

神経衰弱ゲームです。

8000番地からRUNするとLEDが全部消えます。この時点でコンピータは乱数によって1～Fのうちから同じものを2個ずつ4組選んでLEDにばらばらにセットしています。

プレーヤーが1～8のキーを押すとその数で示される位置のLEDに隠されていた数（1～Fのどれか）が表示されます。続いてもう1ヶ所のLEDを選択してその場所を示す数のキーを押します。うまく正解して同じ数を当てたときは両方のLEDは点灯したままになります。もし失敗して違う数だった場合には、今まで表示されていた全部が消えて始めに戻ってしまいます（数の配置は変わりませんから点灯している間に覚えておくようにします）。

またうっかりしてすでに点灯しているLEDを示す数を入力した場合も「お手つき」で表示は全部消えてしまいます。

このようにしてうまく8個のLEDを全部点灯させることができればゲーム終了です。2秒間LEDが点滅し、そのあと試行回数が2秒間表示されます。そして始めに戻って再びゲームが開始されます。

プログラムの説明

LEDに割り当てた数をしまっておくメモリエリアとして8バイトエリア（DTBF）を用意します。はじめに乱数で4個の数を選択して、それぞれ2回乱数によって格納する場所を選んで、DTBFにLED表示パターンを格納します。こういうプログラムではDTBFにキー入力された数値そのものを格納するように考えるのが普通なのでしょうが、今回は数値を計算には利用しないので、どうせ表示するときには表示パターンに変換することを考えると、ここで先に表示パターンに変換してしまってもよいわけです。

ところでこのとき乱数の発生がうまくいかなくて発生させる数値が片寄ってしまい、全部の表示データを算出できないことがあります。このときはアドレス表示が8000のままで表示がブランクになりません。一度リセット（MONキーを押す）してから8000から再スタートしてください。

このプログラムでもちょいとしたテクニックを使っています。DTDP:の部分です。ここではキー入力された1～8の数（これを0～7に変換している）をもとに、対応するLEDに割り当てられた数値パターンを表示させるのですが、そのときすでに表示済み（点灯している）LEDアドレスが指定されていないかをチェックして、もしチェックされていたらそのことをメインルーチンに知らせるためにサインフラグ（S－FLAG）をセットしてリターンします。そうではないときには、LEDに表示するとともにそのデータエリア（DTBF）に表示済みのマークをつけます。

普通は表示済みマークをつけるためにデータエリア（ここではDTBFの8バイト）の他にマーク用に8バイトの場所を用意するなどと考えます。するとそれをチェックするために余計なアドレス計算が必要になってしまいます。DTDP:ではそこのところをちょいとした工夫でパスしています。

DTBFに格納される表示パターンは1～Fですが、共通しているところが1ヶ所あります。

そうです。最上位ビットは常に0です。最上位ビットはLEDの右下のピリオドに割り当てられているので普通は表示しません。これを点灯マークに使うのです。点灯しているときはここを1にします。16進数の約束事で符号付の数として考えるときはその最上位ビットが1のとき、この数はマイナスとして扱います。今回のプログラムでは点灯していないデータは＋で点灯しているデータは－になります。ORA　A命令を使うとAの内容によってS－FLAG（サインフラグ）がON／OFFするので、この情報をメインルーチンに持って帰ることができます。

もうひとつ、DTDP:のはじめの部分でキー入力された情報（00～07）をもとにしてDTBFの該当するアドレスを算出する部分で今までのようにADD　L、MOV　L, Aとしないで、いきなりMOV　L, Aを実行しています。こういうやり方は危険なので注意しなければいけません。ここではDTBFの先頭は8200になっていて、下位8ビットはAレジスタの値と一致するために、ADD　Lを省いたのです。危険なのは将来プログラムを変更するとか、この部分を他のプログラムに利用するとかといった場合に、もしDTBFに半端なアドレスを割り当ててしまうと、正しい結果が得られなくなってしまうことです。

```
2016/4/20  10:1  sinksui8.txt
END=8135
              ; SINKEISUIJAKU for NDZ (ODD COUPLES)
              ; 03/04/29 4/30 5/1 5/5 5/12 5/24
              ;10/6/2 for ND80Z3
              ;16/4/18 for nd8080
              ;4/19
              ;4/20
              ;
                   ORG $8000
                   DTBF=$8200;$E800
              ;     RNDDT=$E810;$E811
                   LED1=$83F8;$FFF8
```

```
                    DP1=$83F4;$FFF4
                    DP3=$83F6;$FFF6
                    CNTR=$83F7;$FFF7;=DP4
                  R=$FFD2
               ;
                    DTDPS=$01C0;$05C0
                    KEYIN1=$0216;$0616
               ;
8000 C30680       JMP START
8003 00        RNDDT:NOP
8004 00           NOP
8005 00        RNDDT2:NOP
               ;
               ;data set
8006 3AD2FF    START:LDA R
8009 210380       LXI H,RNDDT
800C 77           MOV M,A
800D 23           INX H
800E 77           MOV M,A
800F AF           XRA A
8010 0608           MVI B,08
8012 210082         LXI H,DTBF
8015 77        DTST1:MOV M,A
8016 23             INX H
               ;?       DJNZ *DTST1
8017 05           DCR B;
8018 C21580       JNZ DTST1;
801B 0E04           MVI C,04
               ;No. select
801D CD9A80    DTST2:CALL RND4
               ;     ANI 0F
8020 210381         LXI H,SEGDT
8023 5F             MOV E,A
8024 1600           MVI D,00
8026 19             DAD D
8027 7E             MOV A,M
8028 210082         LXI H,DTBF
802B 0608           MVI B,08
802D BE        DTST3:CMP M
802E CA1D80         JZ DTST2;already selected No.
8031 23             INX H
               ;?       DJNZ *DTST3
8032 05           DCR B;
8033 C22D80       JNZ DTST3;
8036 57             MOV D,A;DATA save
               ;ADDRESS select and DATA set
8037 0602           MVI B,02
8039 CD9A80    ADST1:CALL RND4
803C E607           ANI 07
803E 210082         LXI H,DTBF
8041 85             ADD L
8042 6F             MOV L,A
8043 7E             MOV A,M
8044 B7             ORA A
8045 C23980         JNZ ADST1;already used
8048 72             MOV M,D;DATA set
               ;?       DJNZ *ADST1
8049 05           DCR B;
```

```
804A C23980      JNZ ADST1;
804D 0D            DCR C
804E C21D80        JNZ DTST2
                ;GAME START
8051 210000        LXI H,$0000
8054 22F683        SHLD DP3;DECIMAL COUNTER clear
8057 22F483        SHLD DP1
                ;key entry(first one)
805A CDCD80     REENT:CALL CLR
805D CDE280     KEY:CALL KEYDP
8060 F5            PUSH PSW;DATA and S-FLAG save
8061 3AF783        LDA CNTR;DECIMAL COUNTER up
8064 C601          ADI 01
8066 27            DAA
8067 32F783        STA CNTR
806A F1            POP PSW;DATA and S-FLAG
806B FA5A80        JM REENT;otetuki
806E 47            MOV B,A;first DATA save
                ;key entry(second ONE)
806F CDE280     KEY2:CALL KEYDP
8072 FA5A80        JM REENT;otetuki
                ;DATA compere
8075 B8            CMP B; B=first data
8076 C25A80        JNZ REENT;not same
                ;check all open or not
8079 210082        LXI H,DTBF
807C 0608          MVI B,08
807E 7E         OPNCK1:MOV A,M
807F B7            ORA A
8080 F25D80        JP KEY;not open data,continue GAME
8083 23            INX H
                ;?      DJNZ OPNCK1
8084 05          DCR B;
8085 C27E80      JNZ OPNCK1;
                ;OK! ALL LED blink
8088 CDB880        CALL ALBLNK
808B CDC001        CALL DTDPS;COUNTER disp
808E 0614          MVI B,14;=20 ,2sec wait
8090 CD1B81     WAIT:CALL TM01S
                ;?      DJNZ WAIT
8093 05          DCR B;
8094 C29080      JNZ WAIT;
8097 C30680        JMP START
                ;
                ;ransu
809A E5         RND4:PUSH H
809B D5            PUSH D
809C 2A0380     RND2:LHLD RNDDT
809F 23          INX H
80A0 7C          MOV A,H
80A1 E603        ANI 03
80A3 67          MOV H,A
80A4 220380      SHLD RNDDT
80A7 3A0580      LDA RNDDT2
80AA 57          MOV D,A
80AB 7E            MOV A,M
80AC E60F        ANI 0F
80AE BA            CMP D
```

```
80AF CA9C80      JZ RND2
80B2 320580      STA RNDDT2
80B5 D1            POP D
80B6 E1            POP H
80B7 C9            RET
                 ;
                 ;LED all blink
80B8 060A      ALBLNK:MVI B,0A;=10
80BA 3EEF      ALBLNK1:MVI A,EF
80BC D398        OUT 98
80BE CD1B81        CALL TM01S
80C1 3EFF          MVI A,FF
80C3 D398        OUT 98
80C5 CD1B81        CALL TM01S
                 ;?      DJNZ *ALBLNK1
80C8 05          DCR B;
80C9 C2BA80      JNZ ALBLNK1;
80CC C9            RET
                 ;
                 ;LED clear and open mark clear
80CD 21F883    CLR:LXI H,LED1
80D0 110082        LXI D,DTBF
80D3 010008        LXI B,$0800
80D6 71        CLR2:MOV M,C
80D7 1A            LDAX D
80D8 E67F          ANI 7F
80DA 12            STAX D
80DB 23            INX H
80DC 13            INX D
                 ;?      DJNZ *CLR2
80DD 05          DCR B;
80DE C2D680      JNZ CLR2;
80E1 C9            RET
                 ;
                 ;KEYIN and DATA disp
80E2 CD1381    KEYDP:CALL KEYIN1S
80E5 FE09          CPI 09
80E7 D2E280        JNC KEYDP
80EA 3D            DCR A
80EB FAE280        JM KEYDP
80EE 210082    DTDP:LXI H,DTBF
80F1 6F            MOV L,A
80F2 7E            MOV A,M
80F3 B7            ORA A
80F4 F8            RM;otetuki!
80F5 F680          ORI 80
80F7 77            MOV M,A;open mark set
80F8 E67F          ANI 7F
80FA 11F883        LXI D,LED1
80FD 2600          MVI H,00
80FF 19            DAD D
8100 77            MOV M,A
8101 B7            ORA A;clear Sign-flag
8102 C9            RET
8103 5C        SEGDT:DB 5C;0
8104 06            DB 06;1
8105 5B            DB 5B;2
8106 4F            DB 4F;3
```

```
8107 66            DB 66;4
8108 6D            DB 6D;5
8109 7D            DB 7D;6
810A 27            DB 27;7
810B 7F            DB 7F;8
810C 6F            DB 6F;9
810D 77            DB 77;A
810E 7C            DB 7C;b
810F 39            DB 39;C
8110 5E            DB 5E;d
8111 79            DB 79;E
8112 71            DB 71;F
                ;
                ;KEYIN with DE,BC save
8113 C5         KEYIN1S:PUSH B
8114 D5            PUSH D
8115 CD1602        CALL KEYIN1
8118 D1            POP D
8119 C1            POP B
811A C9            RET
                ;
                ;0.1sec timer
811B D5         TM01S:PUSH D
811C 1E64            MVI E,64;=100
811E CD2781     TM01S2:CALL TM1M
8121 1D              DCR E
8122 C21E81          JNZ TM01S2
8125 D1              POP D
8126 C9              RET
                ;
                ;1msec timer
8127 D5         TM1M:PUSH D;11
8128 1607          MVI D,07; ck=5 (11+7+292*7+10+10)/2.048=2082/2.048=1016
812A 1E12       TM1M2:MVI E,12;=18 ck=7 7+15*18+15=292
812C 1D         TM1M3:DCR E;5
812D C22C81        JNZ TM1M3;10
8130 15            DCR D;5
8131 C22A81        JNZ TM1M2;10
8134 D1            POP D;10
8135 C9            RET;10
                ;
ADST1         =8039  ALBLNK       =80B8  ALBLNK1      =80BA
CLR           =80CD  CLR2         =80D6  CNTR         =83F7
DP1           =83F4  DP3          =83F6  DTBF         =8200
DTDP          =80EE  DTDPS        =01C0  DTST1        =8015
DTST2         =801D  DTST3        =802D  KEY          =805D
KEY2          =806F  KEYDP        =80E2  KEYIN1       =0216
KEYIN1S       =8113  LED1         =83F8  OPNCK1       =807E
R             =FFD2  REENT        =805A  RND2         =809C
RND4          =809A  RNDDT        =8003  RNDDT2       =8005
SEGDT         =8103  START        =8006  TM01S        =811B
TM01S2        =811E  TM1M         =8127  TM1M2        =812A
TM1M3         =812C  WAIT         =8090
```

プログラム№9　WANTED（WANTED8．BTK）

ゲームの説明

これはちょっと変わった数当てゲームです。

コンピュータはあらかじめ乱数によって4桁の数（0000～9999）を求めて隠しています。

プレーヤーにはこの数は知らされませんがヒントが示されるので、プレーヤーはヒントを参考にしてこの数を推測し、この数に4桁（1～3桁でもよい）の数を加算して最終的に加算した結果を9999にするとゲーム終了です。

8000番地からRUNするとピーと音がしてLEDの左1桁と3桁目が消灯し、残りは全部0になります。適当な数（たとえば1234）を入力してWRITE　INCキーを押すとコンピュータは入力した数ともとの数を加算して加算結果を記憶します。加算の結果9999になればゲーム終了です。加算結果が9999より大きくなったときはオーバーフローです。ブーという音がしてLEDの左から2桁目と4桁目に”F”が表示されます。この場合には入力した数は加算されず無視されます。オーバーフローしなかった場合にはコンピュータは加算結果の数をチェックしてどこかの桁に”9”があればその9の数をLEDの左から2桁目に表示します。しかし何桁目が9であるかは知らせてくれません。もうひとつヒントを出します。残りの9ではない数のうちどれか1つの数をLEDの左から4桁目に表示します。そして再びピーと音がして次の数の入力待ちになります。入力する数はWRITE　INCを押すまで何回でも入れなおすことができます。

9999が完成するとピーという音がしてLEDの左4桁に9999が表示されます。

何かキーを押すとLEDの左4桁に最初に隠されていた数値が表示され、右4桁に試行回数が表示されます。

もう1回何かキーを押すと始めに戻って再びゲームが開始されます。

「マイコンゲーム21」ではゲームスタートでLEDが全部0になるようにしていますが、そうするとリセットがかかったような気がしてとまどってしまいますので、ここにあるようにLEDの左1桁と3桁目をブランクにしました。

プログラムの説明

特に説明しなくても、多分理解できると思います。

```
2016/4/20  10:22  wanted8.txt
END=818F
                ; WANTED for NDZ
                ; 03/05/06 5/12
                ;10/6/3 for ND80Z3
                ;7/21 9/17
                ;16/4/19 for nd8080
                ;4/20
                ;
                     ORG $8000
                ;     MYDT=$E800
                ;     ACC=$E802
                ;     CNTR=$E805
                ;     RNDDT=$E810;$E811
                     DTRG=$83EC;$FFEC
                     ADRG=$83EE;$FFEE
                     LED1=$83F8;$FFF8
                     LED3=$83FA;$FFFA
                     DP1=$83F4;$FFF4
                     DP3=$83F6;$FFF6
                   R=$FFD2
                ;
                     ADDPS=$01A1;$05A1
                     KEYIN1=$0216;$0616
                ;
                ;data set
8000 C30B80        JMP START
8003 00         MYDT:NOP
8004 00            NOP
8005 00         ACC:NOP
8006 00            NOP
8007 00         CNTR:NOP
8008 00         RNDDT:NOP
```

```
8009 00          NOP
800A 00       RNDDT2:NOP
800B 3AD2FF   START:LDA R
800E 210880      LXI H,RNDDT
8011 77          MOV M,A
8012 23          INX H
8013 77          MOV M,A
8014 AF          XRA A
8015 320780        STA CNTR;COUNTER clear
8018 67            MOV H,A
8019 6F            MOV L,A
801A 22EC83        SHLD DTRG
801D 22EE83        SHLD ADRG
              ;data set
8020 0604          MVI B,04
8022 CD0E81   DTST1:CALL RND4
              ;     ANI 0F
8025 FE0A          CPI 0A
8027 D22280        JNC DTST1
802A 29            DAD H
802B 29            DAD H
802C 29            DAD H
802D 29            DAD H
802E B5            ORA L
802F 6F            MOV L,A
              ;?       DJNZ *DTST1
8030 05          DCR B
8031 C22280      JNZ DTST1
8034 220380        SHLD MYDT
8037 220580        SHLD ACC
              ;GAME START
803A 3E10     START1:MVI A,10;puuuu
803C 060A          MVI B,0A
803E CD4F81   START2:CALL SOUND
              ;?       DJNZ START2
8041 05          DCR B
8042 C23E80      JNZ START2
8045 CDA101   KEY:CALL ADDPS
8048 AF            XRA A
8049 32F883        STA LED1
804C 32FA83        STA LED3
804F 2AEC83   KEY1:LHLD DTRG
8052 CD2C81   KEY2:CALL KEYIN1S
8055 FE15          CPI 15;W INC
8057 CA6B80        JZ ADD
805A FE0A          CPI 0A
805C D25280        JNC KEY2
805F 29            DAD H
8060 29            DAD H
8061 29            DAD H
8062 29            DAD H
8063 B5            ORA L
8064 6F            MOV L,A
8065 22EC83        SHLD DTRG
8068 C34580        JMP KEY
              ;DATA add and check
806B 3A0780   ADD:LDA CNTR
806E C601          ADI 01;DECIMAL COUNTER up
```

```
8070 27          DAA
8071 320780      STA CNTR
             ;?      LD DE,(ACC)
8074 EB        XCHG;
8075 2A0580    LHLD ACC;
8078 EB        XCHG;
8079 7B          MOV A,E
807A 85          ADD L
807B 27          DAA
807C 6F          MOV L,A
807D 7A          MOV A,D
807E 8C          ADC H
807F 27          DAA
8080 DACB80      JC OVER
8083 67          MOV H,A
8084 220580      SHLD ACC
             ;check
8087 1600        MVI D,00
8089 0E04        MVI C,04
808B 0604    CK1:MVI B,04
808D 29      CK2:DAD H
808E 17          RAL
             ;?      DJNZ *CK2
808F 05        DCR B;
8090 C28D80    JNZ CK2;
8093 E60F        ANI 0F
8095 FE09        CPI 09
8097 C29B80      JNZ CK3
809A 14          INR D;count "9"
809B 0D      CK3:DCR C
809C C28B80      JNZ CK1
809F 7A          MOV A,D
80A0 FE04        CPI 04
80A2 CADF80      JZ OK
             ;one No.(except "9") select
80A5 2A0580  CK4:LHLD ACC
80A8 CD0E81      CALL RND4
80AB E603        ANI 03
80AD 3C          INR A
80AE 4F          MOV C,A
80AF 0604    CK5:MVI B,04
80B1 29      CK6:DAD H
80B2 17          RAL
             ;?      DJNZ *CK6
80B3 05        DCR B;
80B4 C2B180    JNZ CK6;
80B7 0D          DCR C
80B8 C2AF80      JNZ CK5
80BB E60F        ANI 0F
80BD FE09        CPI 09
80BF CAA580      JZ CK4
80C2 5F          MOV E,A
             ;result disp
             ;?      LD (ADRG),DE
80C3 EB        XCHG;
80C4 22EE83    SHLD ADRG;
80C7 EB        XCHG;
80C8 C33A80      JMP START1
```

```
80CB 3E00     OVER:MVI A,00;buuuu
80CD 0605          MVI B,05
80CF CD4F81   OVER1:CALL SOUND
              ;?      DJNZ OVER1
80D2 05          DCR B;
80D3 C2CF80      JNZ OVER1;
80D6 21FFFF        LXI H,$FFFF
80D9 22EE83        SHLD ADRG
80DC C34580        JMP KEY
              ;
              ;result="9999"
80DF 3E17     OK:MVI A,17;piiii
80E1 0605          MVI B,05
80E3 CD4F81   OK1:CALL SOUND
              ;?      DJNZ OK1
80E6 05          DCR B;
80E7 C2E380      JNZ OK1;
80EA 2A0580        LHLD ACC
80ED 22EE83        SHLD ADRG
80F0 CDA101        CALL ADDPS
80F3 CD2C81        CALL KEYIN1S
80F6 2A0380        LHLD MYDT
80F9 22EE83        SHLD ADRG
80FC 3A0780        LDA CNTR
80FF 6F            MOV L,A
8100 2600          MVI H,00
8102 22EC83        SHLD DTRG
8105 CDA101        CALL ADDPS
8108 CD2C81        CALL KEYIN1S
810B C30B80        JMP START
              ;
              ;ransu
810E E5       RND4:PUSH H
810F D5            PUSH D
8110 2A0880   RND2:LHLD RNDDT
8113 23          INX H
8114 7C          MOV A,H
8115 E603        ANI 03
8117 67          MOV H,A
8118 220880      SHLD RNDDT
811B 3A0A80      LDA RNDDT2
811E 57          MOV D,A
811F 7E            MOV A,M
8120 E60F        ANI 0F
8122 BA            CMP D
8123 CA1081      JZ RND2
8126 320A80      STA RNDDT2
8129 D1            POP D
812A E1            POP H
812B C9            RET
              ;
              ;KEYIN with DE,BC save
812C C5       KEYIN1S:PUSH B
812D D5          PUSH D
812E CD1602      CALL KEYIN1
8131 D1          POP D
8132 C1          POP B
8133 C9          RET
```

```
                ;
                ;0.1sec timer
8134 D5         TM01S:PUSH D
8135 1E64            MVI E,64;=100
8137 CD4081     TM01S2:CALL TM1M
813A 1D              DCR E
813B C23781          JNZ TM01S2
813E D1              POP D
813F C9              RET
                ;
                ;1msec timer
8140 D5         TM1M:PUSH D;11
8141 1607          MVI D,07; ck=5 (11+7+292*7+10+10)/2.048=2082/2.048=1016
8143 1E12       TM1M2:MVI E,12;=18 ck=7 7+15*18+15=292
8145 1D         TM1M3:DCR E;5
8146 C24581        JNZ TM1M3;10
8149 15            DCR D;5
814A C24381        JNZ TM1M2;10
814D D1            POP D;10
814E C9            RET;10
                ;
814F F5         SOUND:PUSH PSW
8150 E5            PUSH H
8151 D5            PUSH D
8152 C5            PUSH B
8153 217881        LXI H,SNDTBL
8156 85            ADD L
8157 6F            MOV L,A
8158 46            MOV B,M
8159 1E1A          MVI E,1A
815B 50         SNDS1:MOV D,B
815C 3EFF          MVI A,FF;SP OUT=H
815E D398          OUT 98
8160 7F         SNDS2:MOV A,A;5---------
8161 15            DCR D;5          | 5+5+10=20  20/2=10microsec
8162 C26081        JNZ SNDS2;10----
8165 50            MOV D,B
8166 3EDF          MVI A,DF;sp out=L
8168 D398          OUT 98
816A 7F         SNDS3:MOV A,A;5---------
816B 15            DCR D;5          | 5+5+10=20  20/2=10microsec
816C C26A81        JNZ SNDS3;10----
816F 1D            DCR E
8170 C25B81        JNZ SNDS1
8173 C1            POP B
8174 D1            POP D
8175 E1            POP H
8176 F1            POP PSW
8177 C9            RET
                ; SOUND TABLE
8178 7F         SNDTBL:DB 7F;so4
8179 77            DB 77;so#4
817A 71            DB 71;ra4
817B 6A            DB 6A;ra#4
817C 5F            DB 5F;do5
817D 59            DB 59;do#5
817E 54            DB 54;re5
817F 4F            DB 4F;re#
```

```
8180 47          DB 47;fA5
8181 43          DB 43;fa#5
8182 3F          DB 3F;so5
8183 3B          DB 3B;so5#
8184 35          DB 35;ra#5
8185 32          DB 32;si5
8186 2F          DB 2F;do6
8187 2C          DB 2C;do#6
8188 25          DB 25;mi6
8189 27          DB 27;re#6
818A 2A          DB 2A;re6
818B 4B          DB 4B;mi5
818C 38          DB 38;ra5
818D 64          DB 64;si4
818E 23          DB 23;fa6
818F 21          DB 21;fa#6
                 ;
                 ;
ACC         =8005  ADD         =806B  ADDPS       =01A1
ADRG        =83EE  CK1         =808B  CK2         =808D
CK3         =809B  CK4         =80A5  CK5         =80AF
CK6         =80B1  CNTR        =8007  DP1         =83F4
DP3         =83F6  DTRG        =83EC  DTST1       =8022
KEY         =8045  KEY1        =804F  KEY2        =8052
KEYIN1      =0216  KEYIN1S     =812C  LED1        =83F8
LED3        =83FA  MYDT        =8003  OK          =80DF
OK1         =80E3  OVER        =80CB  OVER1       =80CF
R           =FFD2  RND2        =8110  RND4        =810E
RNDDT       =8008  RNDDT2      =800A  SNDS1       =815B
SNDS2       =8160  SNDS3       =816A  SNDTBL      =8178
SOUND       =814F  START       =800B  START1      =803A
START2      =803E  TM01S       =8134  TM01S2      =8137
TM1M        =8140  TM1M2       =8143  TM1M3       =8145
```

プログラム№10　すごろく迷路（SUGOROK8．BTK）

ゲームの説明

LED表示を利用したすごろくゲームです。

8個のLEDのさらに右側にゴールがあります（見えませんがあることにします）。LEDの左側にも1つダミーポジションがあります（これも見えませんがあることにします）。

8000番地からRUNするとピーと音がしてLEDが消灯します。ゲーム開始です。各LEDにはコンピュータが乱数を使って1～8の数を隠しています。そのうち1つは1回でゴールに到着できるようにセットされています。

プレーヤーはスタートポジションを選択するため1～8のどれかを押します。ピッと音がしてその数で示される桁位置のLEDが点灯します。表示される数は右または左に進むステップ数です。プレーヤーは右に進むか左に進むかをキー入力で示します。右ならWRITE　INCを、左なら0を押します。指定した方向にステップ数だけ移動すると、そこでまたピッと音がして次のLEDが点灯します。うまくLEDの右に進んでちょうどゴールで止まればゲーム終了です。ゴールに来てもまだ進むステップが残っているときは残った数だけ左に戻ってしまいます。左に進んだときも同じで左端のLEDを超えてダミーポジションも超えてしまうと残ったステップ数だけ右へもどってしまいます。ダミーポジションで止まってしまわないように計算されています。またゴールに直接行けるのは1ヶ所だけです。みごとゴールできるとプーと音がして、それまでの試行回数が表示されて、2秒間点滅します。そのあと2秒静止してからまた始めに戻って再びゲームが開始されます。

運が悪いと同じところをぐるぐる回るだけでゴールに行けないこともあります。リセットしてやり直してください。

プログラムの説明

まず最初に乱数によって1～8のうちのどれかを選びます。これが最後にゴールに到着するときのステップ数です。つぎにこの数をセットするLEDを計算で求めます。実際にはLED位置に対応するDTBFに数を格納します。最初にDTBFを全て00でクリアしています。その結果HLレジスタはDTBF8バイトの次のアドレス（8208）になっています。;last position　selectのところです。RNDルーチンによって、たとえば1が選ばれたとすると、ゴールに1ステップで行けるのは右端のLED（アドレスは83FF）です。ここに対応するDTBFのアドレスは8207です。つまり選択したステップ数をHL（8208）からマイナスすると、その数を格納するDTBFアドレスが求められます。そこでLからEを引く計算をしているのです。

次の;data　set　では残った7つの場所にやはりRNDによって1～8のいずれかを選んでセットしていきます。最初にセットした最終ポジションはパスします（DTST2：）。ルールにより、ゴールに行けるのは1個所だけ、また左のダミーポジジョンで止まってしまうこともないようにするため、RNDで選んだステップ数とそれを格納するDTBFアドレス（下位アドレスはLED位置と同じ）とから、次のポジションを計算してみて、その結果が08（ゴール）のときとFF（ダミー）のときを除外します（DTST3：）。

さてKEYから0（左）かWRITE　INC（右）かを入力すると、Dレジスタにある現在の位置情報（00～07）と、次の場所へのステップ数（Eレジスタ）から、次のポジションを算出します。右へ進む場合と左へ進む場合とで処理が異なります。;step to right と;step to left です。

ここでは何だかよくわからない計算をしています。まず右の場合です。D＋Eが次のポジションになること、それからその次のポジションが08ならゴール（右端のLEDのポジションが07なので）であることは納得できると思います。問題は加算の結果が08より大きい場合です。この場合は08を超えた分だけ左に戻ります。ここはどうしているかというと、NEGそしてANI　07で次のポジションとしています（？？？）。

NEGは8080には無い命令です。その代わりとしてCMAとINR　Aを使っています。NEGはネガティブで0－Aを結果とします。たとえば08より1だけ多い09について考えてみます。この場合ゴールから1ステップだけ左に戻りますから、次のポジションは07です。では09に対してNEG命令を実行すると結果はどうなるのでしょうか？0－9＝－9ですからAレジスタの値はF7になります。どーして－9がF7になるの？という人はこの説明書を最初から順に読んでこなかった人です。F7は10進に直すと247ですが、符号付の数として考えたときには－9なのです。

うーん。わからない？F7に09を加算してみてください。

```
    11110111 （F7）        247
    00001001 （09）          9
1   00000000 （0100）      256
```

結果は0100で16ビット（2バイト）の数として見れば10進に直して256となります。これは正しい。では2バイトではなくてオーバーフローを許して結果も1バイトだとするとどうなるでしょうか？あくまで結果も1バイトにするのですから0100ではなくて00になります。ほら－9＋9＝0で結果は合っています。

どーもおかしい。だまされているみたいだ、と思うかもしれませんがそういうことだと覚えてしまってください。慣れてくるとだんだん納得できるようになります。

さてもとに戻って説明を続けます。NEG命令によってもとのAレジスタの値は09からF7に変わります。そこでANI　07を実行するとあら不思議、行くべきポジションの07になります。

次は左の処理です。D－Eが次のポジションになることは分かると思います。ただし結果がプラスの場合に限ります。

減算結果がマイナス(Cフラグが立つ)の時はどうするか？またNEG命令を使っています。マイナスのマイナスでプラスになります(もー、余計にわからーん！)。右の処理で説明したことと考え方は同じです。左の処理ではNEGのあとでSUI 02を行っています。どーしてこれでよいのか各自確かめてみてそして考えてみてください。

```
2016/4/20  14:26  sugorok8.txt
END=8173
              ; SUGOROKU for NDZ (SKIP OUT)
              ; 03/05/01 5/2 5/5 5/12 5/13
              ;10/6/3 for ND80Z3
              ;7/21 9/17
              ;16/4/20 for nd8080
              ;
                   ORG $8000
                    DTBF=$8200;$E800
              ;     RNDDT=$E810;$E811
                   LED1=$83F8;$FFF8
                   DP1=$83F4;$FFF4
                   DP3=$83F6;$FFF6
                   CNTR=$83F7;$FFF7;=DP4
              ;
                   DTDPS=$01C0;$05C0
                   KEYIN1=$0216;$0616
                   D1=$02DD;4.684ms
              ;
8000 C30680      JMP START
8003 00       RNDDT:NOP
8004 00          NOP
8005 00          RNDDT2:NOP
              ;last position select
8006 CDB480   START:CALL RND3
              ;     ANI 07
8009 3C           INR A
800A 5F           MOV E,A;distance from goal
800B 3E08         MVI A,08;goal position
800D 93           SUB E
800E 6F           MOV L,A
800F 57           MOV D,A;last position save
8010 2682         MVI H,82;high address of DTBF
8012 73           MOV M,E
              ;data set
8013 210082       LXI H,DTBF
8016 0608         MVI B,08
8018 7D       DTST2:MOV A,L
8019 BA           CMP D
801A CA3080       JZ DTST4;skip last position
801D CDB480   DTST3:CALL RND3
              ;     ANI 07
8020 3C           INR A
8021 4F           MOV C,A;DATA save
8022 85           ADD L;next position(right)
8023 FE08         CPI 08;goal position
8025 CA1D80       JZ DTST3
8028 7D           MOV A,L
8029 91           SUB C;next position(left)
802A FEFF         CPI FF;dummy position
802C CA1D80       JZ DTST3
802F 71           MOV M,C
8030 23       DTST4:INX H
```

```
            ;?      DJNZ *DTST2
8031 05       DCR B;
8032 C21880   JNZ DTST2;
            ;GAME START
8035 210000     LXI H,$0000
8038 22F683     SHLD DP3;DECIMAL COUNTER clear
803B 22F483     SHLD DP1
            ;key entry(1-8)
803E CDD280     CALL CLR
8041 3E0A       MVI A,0A
8043 0610       MVI B,10
8045 CD0A81 START1:CALL SOUND
            ;?      DJNZ START1
8048 05       DCR B;
8049 C24580   JNZ START1;
804C CDF480 KEY:CALL KEYIN1S;1-8 input
804F FE09       CPI 09
8051 D24C80     JNC KEY
8054 3D         DCR A
8055 FA4C80     JM KEY
8058 57     NEXT:MOV D,A;position save
8059 3AF783     LDA CNTR;DECIMAL COUNTER up
805C C601       ADI 01
805E 27         DAA
805F 32F783     STA CNTR
8062 CD4B81     CALL DTDP;data disp and save to E
            ;right or left in
8065 CDF480 KEY1:CALL KEYIN1S
8068 B7         ORA A
8069 CA8580     JZ LEFT
806C FE15       CPI 15;WRITE INC
806E C26580     JNZ KEY1
            ;step to right
8071 7A         MOV A,D;current position(00-07)
8072 83         ADD E;E=step No.      D+E=next position
8073 FE08       CPI 08
8075 CA9180     JZ GOAL
8078 DA7F80     JC RIGHT1;    D+E<08
            ;?      NEG; 0-A
807B 2F       CMA;
807C 3C       INR A;
807D E607       ANI 07;next position
807F CDD280 RIGHT1:CALL CLR
8082 C35880       JMP NEXT
            ;step to left
8085 7A     LEFT:MOV A,D;current position(00-07)
8086 93         SUB E;E=step No.
8087 D27F80     JNC RIGHT1
            ;?      NEG
808A 2F       CMA;
808B 3C       INR A;
808C D602       SUI 02;next position
808E C37F80     JMP RIGHT1
            ;goal
8091 3E00   GOAL:MVI A,00
8093 0610       MVI B,10
8095 CD0A81 GOAL1:CALL SOUND
            ;?      DJNZ GOAL1
```

```
8098 05          DCR B;
8099 C29580      JNZ GOAL1;
809C CDD280        CALL CLR
809F CDFC80        CALL TM01S
80A2 CDC001        CALL DTDPS;COUNTER disp
80A5 CDDF80        CALL ALBLNK
80A8 0614          MVI B,14;=20 2sec wait
80AA CDFC80   WAIT:CALL TM01S
              ;?       DJNZ WAIT
80AD 05          DCR B;
80AE C2AA80      JNZ WAIT;
80B1 C30680        JMP START
              ;
              ;ransu
80B4 E5       RND3:PUSH H
80B5 D5            PUSH D
80B6 2A0380   RND2:LHLD RNDDT
80B9 23          INX H
80BA 7C          MOV A,H
80BB E603        ANI 03
80BD 67          MOV H,A
80BE 220380      SHLD RNDDT
80C1 3A0580      LDA RNDDT2
80C4 57          MOV D,A
80C5 7E            MOV A,M
80C6 E607        ANI 07
80C8 BA            CMP D
80C9 CAB680      JZ RND2
80CC 320580      STA RNDDT2
80CF D1            POP D
80D0 E1            POP H
80D1 C9            RET
              ;
              ;LED clear
80D2 21F883   CLR:LXI H,LED1
80D5 010008        LXI B,$0800
80D8 71       CLR2:MOV M,C
80D9 23            INX H
              ;?       DJNZ *CLR2
80DA 05          DCR B;
80DB C2D880      JNZ CLR2;
80DE C9            RET
              ;
              ;LED all blink
80DF 060A     ALBLNK:MVI B,0A;=10
80E1 3EEF     ALBLNK1:MVI A,EF
80E3 D398        OUT 98
80E5 CDFC80        CALL TM01S
80E8 3EFF          MVI A,FF
80EA D398        OUT 98
80EC CDFC80        CALL TM01S
              ;?       DJNZ *ALBLNK1
80EF 05          DCR B;
80F0 C2E180      JNZ ALBLNK1;
80F3 C9            RET
              ;
              ;KEYIN with DE,BC save
80F4 C5       KEYIN1S:PUSH B
```

```
80F5 D5          PUSH D
80F6 CD1602      CALL KEYIN1
80F9 D1          POP D
80FA C1          POP B
80FB C9          RET
                 ;
                 ;
                 ;0.1sec timer
80FC D5          TM01S:PUSH D
80FD 1E15          MVI E,15;=21
80FF D5          TM01S2:PUSH D
8100 CDDD02        CALL D1;4.684ms
8103 D1            POP D
8104 1D              DCR E
8105 C2FF80          JNZ TM01S2
8108 D1              POP D
8109 C9              RET
                 ;
810A F5          SOUND:PUSH PSW
810B E5            PUSH H
810C D5            PUSH D
810D C5            PUSH B
810E 213381        LXI H,SNDTBL
8111 85            ADD L
8112 6F            MOV L,A
8113 46            MOV B,M
8114 1E1A          MVI E,1A
8116 50          SNDS1:MOV D,B
8117 3EFF          MVI A,FF;SP OUT=H
8119 D398          OUT 98
811B 7F          SNDS2:MOV A,A;5---------
811C 15            DCR D;5         | 5+5+10=20  20/2=10microsec
811D C21B81        JNZ SNDS2;10----
8120 50            MOV D,B
8121 3EDF          MVI A,DF;sp out=L
8123 D398          OUT 98
8125 7F          SNDS3:MOV A,A;5---------
8126 15            DCR D;5         | 5+5+10=20  20/2=10microsec
8127 C22581        JNZ SNDS3;10----
812A 1D            DCR E
812B C21681        JNZ SNDS1
812E C1            POP B
812F D1            POP D
8130 E1            POP H
8131 F1            POP PSW
8132 C9            RET
                 ; SOUND TABLE
8133 7F          SNDTBL:DB 7F;so4
8134 77            DB 77;so#4
8135 71            DB 71;ra4
8136 6A            DB 6A;ra#4
8137 5F            DB 5F;do5
8138 59            DB 59;do#5
8139 54            DB 54;re5
813A 4F            DB 4F;re#
813B 47            DB 47;fA5
813C 43            DB 43;fa#5
813D 3F            DB 3F;so5
```

```
813E 3B          DB 3B;so5#
813F 35          DB 35;ra#5
8140 32          DB 32;si5
8141 2F          DB 2F;do6
8142 2C          DB 2C;do#6
8143 25          DB 25;mi6
8144 27          DB 27;re#6
8145 2A          DB 2A;re6
8146 4B          DB 4B;mi5
8147 38          DB 38;ra5
8148 64          DB 64;si4
8149 23          DB 23;fa6
814A 21          DB 21;fa#6
                 ;
                 ;
                 ;DATA disp and save to E
814B 7A       DTDP:MOV A,D
814C 210082         LXI H,DTBF
814F 85             ADD L
8150 6F             MOV L,A
8151 7E             MOV A,M
8152 5F             MOV E,A;DATA save
8153 216B81         LXI H,SEGDT
8156 4F             MOV C,A
8157 0600           MVI B,00
8159 09             DAD B
815A 7E             MOV A,M
815B 47             MOV B,A;segment data save
815C 21F883         LXI H,LED1
815F 7A             MOV A,D
8160 85             ADD L
8161 6F             MOV L,A
8162 70             MOV M,B
8163 3E10           MVI A,10
8165 0603           MVI B,03
8167 CD0A81   DTDP1:CALL SOUND
                 ;?      DJNZ *DTDP1
816A C9             RET
816B 5C       SEGDT:DB 5C;0
816C 06             DB 06;1
816D 5B             DB 5B;2
816E 4F             DB 4F;3
816F 66             DB 66;4
8170 6D             DB 6D;5
8171 7D             DB 7D;6
8172 27             DB 27;7
8173 7F             DB 7F;8
                 ;
ALBLNK       =80DF  ALBLNK1      =80E1  CLR          =80D2
CLR2         =80D8  CNTR         =83F7  D1           =02DD
DP1          =83F4  DP3          =83F6  DTBF         =8200
DTDP         =814B  DTDP1        =8167  DTDPS        =01C0
DTST2        =8018  DTST3        =801D  DTST4        =8030
GOAL         =8091  GOAL1        =8095  KEY          =804C
KEY1         =8065  KEYIN1       =0216  KEYIN1S      =80F4
LED1         =83F8  LEFT         =8085  NEXT         =8058
RIGHT1       =807F  RND2         =80B6  RND3         =80B4
RNDDT        =8003  RNDDT2       =8005  SEGDT        =816B
```

```
SNDS1       =8116  SNDS2       =811B  SNDS3       =8125
SNDTBL      =8133  SOUND       =810A  START       =8006
START1      =8045  TM01S       =80FC  TM01S2      =80FF
WAIT        =80AA
```

プログラム№11　REVERSE（REVERSE8．BTK）

ゲームの説明

数字の並べ替えゲームです。

8000番地からRUNすると1～8の数がばらばらの順序で表示されます。これをルールにしたがって12345678に並べ替えるとゲーム終了です。

1～8の数を入力すると左からその桁数分の表示が点滅します。WRITE　INCを押すと入力が確定します。WRITE　INCを押すまでは入力をやり直すことができます。WRITE　INCによって入力が確定すると点滅していた表示が左右反転します。これを繰り返して12345678が完成すると全体が2秒間点滅します。次に試行回数が表示され2秒間静止します。そのあとまた始めに戻って再びゲームが開始されます。

プログラムの説明

まず最初にDTBF（8バイト）をクリアします。ここに1～8の数をランダムに選んで格納します。8桁の数ですし、LEDに表示するわけですから、LED表示用のメモリエリア（83F4～83F7）を使うか、アドレスレジスタ及びデータレジスタ（83EC～83EF）を使った方が結果の表示は簡単になります。そうすると8桁のBCD数として扱うことになります。しかし今回のプログラムのように桁単位で数の入れ替えが必要な処理ではBCD数というのはなかなかに面倒なのです。1桁が4ビットなのですが8080には4ビット単位でデータを扱う命令はありません。4ビット単位で扱うにはシフト命令で1ビットずつ4回シフトして1桁ごとに分離するしかありません。そこでこのプログラムでは処理のたびにどうせ1桁ずつ分離しなければならない（今のところ、多分）のならはじめから分離して1桁ずつの数にして（たとえば、56というBCD数ではなくて、05、06という数にして）8バイトのデータエリアに格納し、ここで1バイト単位で数の入れ替えをするように考えました。1バイト単位ならいろんな命令が使えるので楽になりそうです。8バイトのデータバッファの内容を表示するには、4バイトのBCD数に直さなければいけませんが、これは結果を表示するときだけですから、変換するサブルーチンを作って処理することにしました。

まずDTBFを0クリアします。DTST0：のところです。ここまでにも出てきてると思いますが、XRA　AはMVI　A，00の代わりとしてよく使います。Aレジスタを00にするため以外の意図はありません。DTST1：以下のところで1～8の数を乱数によりランダムに作り、それをDTBFに格納しています。なんだかややこしいことをしているようですが、これは1～8を1度だけ選んで格納するための処理なのです。まずRNDにより1～8のうちの1つの数を選び、それをDTBFに格納するのですが、そのときDTBFの先頭から順にメモリの内容を確認します（DTST2：）。ここでテクニックを使っています。MOV　A，Mとして、ORA　Aなどとしたいところですが、AレジスタにはRNDで選んだ数がすでに入っています。PUSH PSWとするとか、MOV　D，Aとしたりしてまず Aの中身を退避させておいて…などと考えるのは、教条主義です（懐かしい言葉。知らないだろうなー。今どき）。ここは遊んでいるCレジスタを使ってDCR　Cとしてしまいます。メモリの内容が00ならDCR　Cの結果はFFになって、前のプログラムで出てきたようにマイナスになるのでSフラグが立ちます。でJMといくわけです。ついでですが、SUB命令などではマイナスになると同時にキャリー（Cフラグ）も立つので、よくJCを使いますが、DCRとINRについてはCフラグのみ変化しないので注意してください。

さてDTBFを前から順に確認していって最初に00の場所をみつけたら、Aに入っている数を格納してしまいます（あれ？同じ数を選んでしまったかどうかの確認はしなくてもよいのか？）。プログラムの流れでは逆になっているようで分かりにくいのですがちゃんとやっているのです（DTST3：）。RNDで新しい数を選んだら必ずDTBFの先頭から内容を確認しています。もし00でなかったらDTRD3：で同じ数か異なる数かをチェックします。ですからこのチェックをパスしてきて、もう比較するデータが無い（つまり次のDTBFの内容が00のとき）なら、新しく出てきた数になるのでDTBFに格納してもよいことになります。

RNDで乱数がうまく作り出せなくてアドレスが8000の表示のままでハングアップしてしまうことがあります。リセットしてもう一度8000から実行してみてください。

無事8桁の数が決まったらまずLEDに表示します。DTBFDPサブルーチンです。DTBFの8バイトの数をBCD4バイトに変換してDP1～に格納します。DTBFは8バイトですが内容は下位4ビットに1～8が入っているだけですから、最初のバイトを読んでそれを左に4ビットシフトして、そこに次のバイトの下位4ビット（上位4ビットは常に0）をORで合わせればBCD2桁が出来あがります。最後にLED表示ルーチンをCALLします。

キーの入力です。入力前にCレジスタに01を入れています。あとで出て来ますが、Cレジスタにはキー入力した1～8の桁指定数が入るのです。普通はまず桁指定の数を入力して、次にWRITE　INCの入力ですが、そのようにキー入力ルーチンを2段階に並べても、WRITE　INCの入力待ちのときに数値を入れなおす場合もでてきます。そうすると結構複雑なプログラムになってしまいます。ここはこのプログラムのように数値でもWRITE　INCでもその入力順序に関係無く受けつけてしまうとよいのです。数値が入力されたらそれをCレジスタに保存しておけば、次にWRITE　INCが押されたときにCレジスタの値をもって処理ルーチンへ行くことができます。しかし中にはまだ数値を入力していないのにWRITE　INCを押してしまうあわて者がいるかもしれません。最初にCレジスタに01を入れているのはこのあわて者対策です。データを反転させる桁数で1を指定しても結果は変わりません。このあとで出てくる反転処理ルーチン（REV：）ではC＝01のときは何もしないでリターンします。

ではいよいよそのREV：の説明です。何をしているのかちょっと見にはよくわかりません。でもここはごく当たり前の処理をしています。まずDTBFのトップ（LEDの左端に相当）アドレスをHLに、そしてCレジスタの値から、反転させる範囲

の右端のDTBFアドレスを算出してDEに入れます。CレジスタはC＝01をチェックするためとDEへのアドレス計算のためにDCRしたものをもとに戻すためINRしたあとSRLを実行します。SRLも8080には無い命令です（その代わりとして複数の命令を使っています）。Cレジスタの値が1／2になります。もとの値が奇数の場合は1／2した結果できる端数は切り捨てられます。

これで準備完了です。AレジスタとBレジスタを利用して両端のデータ、(HL)と(DE)を入れ替えます。このあとHLを＋1、DEを－1します。つまり外側から内側に1バイトずつアドレスを寄せるのです。この処理をC＝00になるまで繰り返すとデータの前後が全部入れ替わります。対象が奇数桁の場合には中央の桁は動きません。アドレスをHLとDEで1ずつせばめるのでCを1／2にしているのですが、奇数桁の場合には最後に残った1バイトは中央にあって動きませんからCを1／2したときに切り捨ててよいのです。むむ、我ながら、なんと巧妙な（自画自賛）。

この部分がもしBCD数であったとすると面倒な処理になります。

データを入れ替えたあと、結果をLEDに表示します（DTBFDP）。そのあと、数字が並んだかどうかチェックします（;compare DATA）。ここもBCD数だと面倒です（ただし今回は結果が12345678になったかどうかの比較をすればよいので各桁の大小比較をするよりずっと楽ではあります）。このプログラムではDTBFを先頭から順に読み出して、大小を比較するだけで済みますからいとも簡単です。

順序が逆になってしまいましたがBLNKCについて説明します。指定した桁の表示だけブリンクさせるものです。ブリンクする桁数をCレジスタに入れます。ブリンクする準備として対象になるLEDレジスタの内容をBLNKBFにコピーします。準備ができたら0．1秒ごとにLEDレジスタの内容を読み出して、BLNKBFと比較します。ここは比較しなくても00かどうかを確認すればよいのですが、その次にBLNKBFの値をLEDレジスタにコピーする作業があるため、先にAレジスタにBLNKBFの値を読み出すという、普通の考えとは逆のようなことをしています（CMP　MはあるがCMP　(DE)という命令は無い）。こういうところがちょいとした工夫です。内容が一致したら、LEDレジスタに00を書き込み、不一致なら（つまりLEDレジスタの内容が00なら）BLNKBFにコピーしてあるもとの表示データをLEDレジスタに書き込みます。これを繰り返すと表示がブリンクします。このルーチンではブリンク中にキーからの入力も確認しています。キー入力があるとブリンクを停止してBLNKBFの内容をもとのLEDレジスタに戻してリターンします。

```
2016/4/20  10:30  reverse8.txt
END=8182
              ; REVERSE for NDZ
              ; 03/04/29 4/30 5/1 5/2 5/5 5/13 5/15
              ;10/6/3 for ND80Z3
              ;16/4/19 for nd8080
              ;4/20
              ;
                   ORG $8000
              ;     DTBF=$E800
              ;     BLNKBF=$E808
              ;     RNDDT=$E810;$E811
              ;     CNTR=$E812
                   LED1=$83F8;$FFF8
                   DP1=$83F4;$FFF4
                   DP3=$83F6;$FFF6
                 R=$FFD2
              ;
                   DTDPS=$01C0;$05C0
                   KEYIN1=$0216;$0616
                   KEYIN2=$0223;$0623
              ;
              ;data set
8000 C31780      JMP START
8003 00        DTBF:NOP
8004 00          NOP
8005 00          NOP
8006 00          NOP
8007 00          NOP
8008 00          NOP
8009 00          NOP
800A 00          NOP
800B 00        BLNKBF:NOP
800C 00          NOP
```

```
800D 00          NOP
800E 00          NOP
800F 00          NOP
8010 00          NOP
8011 00          NOP
8012 00          NOP
8013 00       RNDDT:NOP
8014 00          NOP
8015 00       RNDDT2:NOP
8016 00       CNTR:NOP
              ;
8017 3AD2FF   START:LDA R
801A 211380      LXI H,RNDDT
801D 77          MOV M,A
801E 23          INX H
801F 77          MOV M,A
8020 0608        MVI B,08
8022 210380         LXI H,DTBF
8025 AF             XRA A
8026 77       DTST0:MOV M,A
8027 23             INX H
              ;?       DJNZ *DTST0
8028 05          DCR B;
8029 C22680      JNZ DTST0;
802C CDC480   DTST1:CALL RND3
              ;     ANI 07
802F 3C           INR A
8030 0608         MVI B,08
8032 210380       LXI H,DTBF
8035 4E       DTST2:MOV C,M
8036 0D           DCR C
8037 F24280       JP DTST3
803A 77           MOV M,A
              ;?      DJNZ *DTST1
803B 05          DCR B;
803C C22C80      JNZ DTST1;
803F C34B80        JMP ENTRY
8042 BE       DTST3:CMP M
8043 CA2C80       JZ DTST1
8046 23           INX H
              ;?      DJNZ *DTST2
8047 05          DCR B;
8048 C23580      JNZ DTST2;
804B CD3181   ENTRY:CALL DTBFDP
804E AF           XRA A
804F 321680       STA CNTR;DECIMAL COUNTER clear
              ;key entry(1-8)
8052 0E01     REENT:MVI C,01
8054 CD4B81   KEY:CALL KEYIN1S
8057 FE15     KEY1:CPI 15;WRITE INC
8059 CA6D80       JZ REV
805C FE09         CPI 09
805E D25480       JNC KEY
8061 FE02         CPI 02
8063 DA5480       JC KEY
              ;blink select data
8066 4F           MOV C,A
8067 CDF780       CALL BLNKC
```

```
806A C35780        JMP KEY1
                ;reverse
806D 0D         REV:DCR C
806E CA5280        JZ REENT
8071 210380        LXI H,DTBF
8074 54            MOV D,H
8075 79            MOV A,C
8076 85            ADD L
8077 5F            MOV E,A
8078 0C            INR C
                ;?      SRL C
8079 F5           PUSH PSW;
807A 79           MOV A,C;
807B B7           ORA A;
807C 1F           RAR;
807D 4F           MOV C,A;
807E F1           POP PSW;
807F 1A         REV1:LDAX D
8080 46            MOV B,M
8081 77            MOV M,A
8082 78            MOV A,B
8083 12            STAX D
8084 23            INX H
8085 1B            DCX D
8086 0D            DCR C
8087 C27F80        JNZ REV1
808A CD3181     REV2:CALL DTBFDP
808D 3A1680        LDA CNTR;DECIMAL COUNTER up
8090 C601          ADI 01
8092 27            DAA
8093 321680        STA CNTR
                ;compare DATA
8096 210380        LXI H,DTBF
8099 0607          MVI B,07
809B 7E         COMP1:MOV A,M
809C 23            INX H
809D BE            CMP M
809E D25280        JNC REENT;not yet
                ;?      DJNZ *COMP1
80A1 05           DCR B;
80A2 C29B80       JNZ COMP1;
                ;OK! ALL LED blink
80A5 CDE280        CALL ALBLNK
                ;COUNTER disp
80A8 210000        LXI H,$0000
80AB 22F483        SHLD DP1
80AE 3A1680        LDA CNTR
80B1 67            MOV H,A
80B2 22F683        SHLD DP3
80B5 CDC001        CALL DTDPS
80B8 0614          MVI B,14;=20 ,2sec wait
80BA CD5B81     WAIT:CALL TM01S
                ;?      DJNZ WAIT
80BD 05           DCR B;
80BE C2BA80       JNZ WAIT;
80C1 C31780        JMP START
                ;
                ;ransu
```

```
80C4 E5       RND3:PUSH H
80C5 D5            PUSH D
80C6 2A1380   RND2:LHLD RNDDT
80C9 23          INX H
80CA 7C          MOV A,H
80CB E603        ANI 03
80CD 67          MOV H,A
80CE 221380      SHLD RNDDT
80D1 3A1580      LDA RNDDT2
80D4 57          MOV D,A
80D5 7E            MOV A,M
80D6 E607        ANI 07
80D8 BA            CMP D
80D9 CAC680      JZ RND2
80DC 321580      STA RNDDT2
80DF D1            POP D
80E0 E1            POP H
80E1 C9            RET
             ;
             ;LED all blink
80E2 060A     ALBLNK:MVI B,0A;=10
80E4 3EEF     ALBLNK1:MVI A,EF
80E6 D398        OUT 98
80E8 CD5B81        CALL TM01S
80EB 3EFF                MVI A,FF
80ED D398        OUT 98
80EF CD5B81        CALL TM01S
             ;?      DJNZ *ALBLNK1
80F2 05          DCR B;
80F3 C2E480      JNZ ALBLNK1;
80F6 C9            RET
             ;LED blink
80F7 79       BLNKC:MOV A,C;save C
80F8 21F883        LXI H,LED1
80FB 110B80        LXI D,BLNKBF
80FE 0600          MVI B,00
             ;?      LDIR
8100 CD7681      CALL LDIR
8103 4F            MOV C,A
8104 21F883   BLNKC1:LXI H,LED1
8107 110B80        LXI D,BLNKBF
810A 41            MOV B,C
810B 1A       BLNKC2:LDAX D
810C BE            CMP M
810D C21181        JNZ BLNKC3
8110 AF            XRA A
8111 77       BLNKC3:MOV M,A
8112 23            INX H
8113 13            INX D
             ;?      DJNZ *BLNKC2
8114 05          DCR B;
8115 C20B81      JNZ BLNKC2;
8118 CD5B81        CALL TM01S
811B CD5381        CALL KEYIN2S
811E 3C            INR A
811F CA0481        JZ BLNKC1;no key in
8122 3D            DCR A;key data
8123 210B80        LXI H,BLNKBF
```

```
8126 11F883        LXI D,LED1
8129 0600          MVI B,00
812B C5            PUSH B
                ;?      LDIR
812C CD7681        CALL LDIR
812F C1            POP B
8130 C9            RET
                ;
                ;DTBF data disp
8131 210380     DTBFDP:LXI H,DTBF
8134 11F483        LXI D,DP1
8137 0604          MVI B,04
8139 7E         DTBFDP1:MOV A,M
813A 07            RLC
813B 07            RLC
813C 07            RLC
813D 07            RLC
813E 23            INX H
813F B6            ORA M
8140 12            STAX D
8141 23            INX H
8142 13            INX D
                ;?      DJNZ *DTBFDP1
8143 05            DCR B;
8144 C23981        JNZ DTBFDP1;
8147 CDC001        CALL DTDPS
814A C9            RET
                ;
                ;KEYIN with DE,BC save
814B C5         KEYIN1S:PUSH B
814C D5            PUSH D
814D CD1602        CALL KEYIN1
8150 D1            POP D
8151 C1            POP B
8152 C9            RET
8153 C5         KEYIN2S:PUSH B
8154 D5            PUSH D
8155 CD2302        CALL KEYIN2
8158 D1            POP D
8159 C1            POP B
815A C9            RET
                ;
                ;0.1sec timer
815B D5         TM01S:PUSH D
815C 1E64          MVI E,64;=100
815E CD6781     TM01S2:CALL TM1M
8161 1D            DCR E
8162 C25E81        JNZ TM01S2
8165 D1            POP D
8166 C9            RET
                ;
                ;1msec timer
8167 D5         TM1M:PUSH D;11
8168 1607          MVI D,07; ck=5 (11+7+292*7+10+10)/2.048=2082/2.048=1016
816A 1E12       TM1M2:MVI E,12;=18 ck=7 7+15*18+15=292
816C 1D         TM1M3:DCR E;5
816D C26C81        JNZ TM1M3;10
8170 15            DCR D;5
```

```
8171 C26A81      JNZ TM1M2;10
8174 D1          POP D;10
8175 C9          RET;10
                 ;
8176 F5          LDIR:PUSH PSW
8177 7E          LDIR2:MOV A,M
8178 12             STAX D
8179 23             INX H
817A 13             INX D
817B 0B             DCX B
817C 78             MOV A,B
817D B1             ORA C
817E C27781         JNZ LDIR2
8181 F1             POP PSW
8182 C9             RET
                 ;
ALBLNK       =80E2  ALBLNK1      =80E4  BLNKBF       =800B
BLNKC        =80F7  BLNKC1       =8104  BLNKC2       =810B
BLNKC3       =8111  CNTR         =8016  COMP1        =809B
DP1          =83F4  DP3          =83F6  DTBF         =8003
DTBFDP       =8131  DTBFDP1      =8139  DTDPS        =01C0
DTST0        =8026  DTST1        =802C  DTST2        =8035
DTST3        =8042  ENTRY        =804B  KEY          =8054
KEY1         =8057  KEYIN1       =0216  KEYIN1S      =814B
KEYIN2       =0223  KEYIN2S      =8153  LDIR         =8176
LDIR2        =8177  LED1         =83F8  R            =FFD2
REENT        =8052  REV          =806D  REV1         =807F
REV2         =808A  RND2         =80C6  RND3         =80C4
RNDDT        =8013  RNDDT2       =8015  START        =8017
TM01S        =815B  TM01S2       =815E  TM1M         =8167
TM1M2        =816A  TM1M3        =816C  WAIT         =80BA
```

プログラム№12　ROCK　CLIMBING（ROCKCLM8．BTK）

ゲームの説明

これは今までのゲームとは異なり、コンピュータ相手ではなく、二人で行う対戦ゲームです。お互いに0mからスタートし途中に足場を築きながら速く100mを登った方が勝ちというものです。運が悪かったりあまり先を急ぎすぎるとときどき転落してしまいます。

8000番地からRUNするとLEDの左側と右側にそれぞれスタートの000が表示されます。プレイは交互に行います。先攻は左側です。まず左のLEDが点滅してキー入力待ちになります。プレーヤーは上に登るかここで休憩して足場を築くか選択することができます。上に登るならREAD　INCを押します。休憩して足場を築くならREAD　DECを押します。登る場合にはLEDの1桁目（後攻は5桁目）に”8”の上側のセグメントのみ点灯表示されます。休憩なら”8”の下側のセグメントが点灯表示されます。確定するにはWRITE　INCを押します。WRITE　INCの入力前なら登るか休憩するかの選択を変更することができます。

上に登るを選択した場合、コンピュータは乱数を利用したサイコロを2回振って出た目の合計を上に登るm数として現在のmに加算して表示します。ただしどちらのサイコロでも1の目が出ると前の足場まで転落してしまいます。もし足場を築いていないときは0mまで転落してしまいます。転落したら攻守交代です。適当なところまで登ったら一旦休憩してそこを足場にする方がよいようです。休憩したときも攻守交代になります。転落する確率は1／3ですからかなり高率です（ちょっとサイコロに片寄りがあって続けて転落したり片方のプレーヤーだけやたら有利なときがあります）。

どちらかが先に100mに到達すると（100mを超えてもよい）全体が2秒間点滅します。そのあとまた始めに戻って再びゲームが開始されます。

プログラムの説明

足場を記憶するために各1バイトのメモリエリアを使います。BCDデータで00～99を入れればよいので1バイトで足ります。プレーヤー1の現在の位置を記憶するためにアドレスレジスタを、プレーヤー2のそれにはデータレジスタをそのまま使います。BCD数として扱うのでこうすれば表示のための処理の手間が省けます。

今回はもっと大胆に（？）手間を省くため、プレーヤー1とプレーヤー2の処理を同じプログラムでやってしまいます。そのためにHLレジスタだけでは足りないので、IXレジスタとIYレジスタを動員します（残念ながら8080にはIXレジスタもIYレジスタもありませんから代わりにワークエリアとしてIXWK、IYWKを使っています）。HLにはLEDの1桁目か5桁目のアドレスを入れます。登るか休むかのマークを表示するためです。IXWKには現在の位置データの格納アドレス（PLYR1かPLYR2）を入れます。IYWKには足場データを記憶するアドレス（ASI1かASI2）を入れます。両者の処理の違いはこれだけなので、その相違しているところを、HL、IXWK、IYWKに入れてしまうと、全く同じ処理で済んでしまいます。

それでは処理の内容を見ていきます。DTDPで両プレーヤーの現在位置をLEDに表示します。LED1とLED5はブランクにします。次にLEDブリンクとキー入力を行います（BLNK）。このサブルーチンは　№11　REVERSE　のBLNKCと同じ動作です。今回はブリンクする桁が3バイト固定なのでCレジスタの使い方は異なります。BLBKルーチンの中でキー入力をチェックしますが、READ　INC（14）かREAD　DEC（13）以外は受け付けません。いずれかをAレジスタに入れてリターンします。

USDISP:ではAレジスタの内容によりLEDに登りのマークか休憩のマークを表示します。ANI　03は入力されたコードの14と13を簡単に区別ためのものでそれ以外の意味はありません。この値は後の処理のためCレジスタにも保存されます。マーク表示後再びキー入力を待ちます（KEY:）。WRITE　INCの入力を待つためです。このときまたREAD　INCかREAD　DECが入力されたら、USDISP:へ戻ってマークを表示します。WRITE　INCが押されたら、Cレジスタを確認します。DCR　CでSフラグが立てばREAD　INCなのでサイコロを振る処理に行きます（C＝00のときDCR　Cを実行すると結果はFFでマイナスになる）。そうでなければ休憩（足場の処理）に行きます。

SAI:ではサイコロを2回振って出た目を加算します。Cレジスタを0クリアしておいて、ここに加算します。サイコロは1～6ですが処理を簡単にするため、乱数で0～5を求め、これに1を加えます。ただし＋1する前に0であれば、これは1の目ですから、墜落の処理（DOWN）へ行きます。

サイコロの目を2回加算した結果はCレジスタとAレジスタに入ります。Cは2回加算を行うためのワークとして使っただけなので、この後の処理では用済みです。Aレジスタを使います。Aレジスタと現在の高さデータ（IXWKにその格納メモリアドレスがある）を加算します。加算した結果、キャリーが発生すればBCD2桁の加算でオーバーフローが発生したわけですから、100mを超えたことになります。ゲーム終了です。そうでなければ同じプレーヤーで最初の表示ブリンクとキー入力待ちの処理に戻ります。

ASISET:では足場データの更新を行います。IXWKで示されるメモリアドレスに入っている現在の高さデータをIYWKで示される足場データの入っているメモリアドレスにCOPYします。そのあと選手交代の処理をします。現在がどちらのプレーヤーの番かはHLレジスタに入っているのが、LED1のアドレスかLED5のアドレスかを見ればわかります。そこで選手交代して最初の処理に戻ります。

DOWN:ではIXWKに入っている現在の高さを格納するメモリアドレスにIYWKの情報で示される足場データをCOPYします。そのあと選手交代の処理をしてゲームを続けます。

```
END=817F
                ; LOCK CLIMBING for NDZ
                ; 03/05/08 5/14
                ;10/6/3 for ND80Z3
                ;7/22 9/22
                ;16/4/19 for nd8080
                ;4/20
                ;
                     ORG $8000
                ;     ASI1=$E800
                ;     ASI2=$E801
                ;     BLNKBF=$E808
                ;     RNDDT=$E810;$E811
                     PLYR2=$83EC;$FFEC;=DTRG
                     PLYR1=$83EE;$FFEE;=ADRG
                     LED1=$83F8;$FFF8
                     LED5=$83FC;$FFFC
                   R=$FFD2
                ;
                     ADDPS=$01A1;$05A1
                     KEYIN1=$0216;$0616
                     KEYIN2=$0223;$0623
                ;
8000 C30F80        JMP START
8003 00         ASI1:NOP
8004 00         ASI2:NOP
8005 00         BLNKBF:NOP
8006 00            NOP
8007 00            NOP
8008 00         RNDDT:NOP
8009 00            NOP
800A 00         RNDDT2:NOP
800B 00         IXWK:NOP
800C 00            NOP
800D 00         IYWK:NOP
800E 00            NOP
800F 3AD2FF     START:LDA R
8012 210880        LXI H,RNDDT
8015 77            MOV M,A
8016 23            INX H
8017 77            MOV M,A
8018 210000        LXI H,$0000
801B 22EE83          SHLD PLYR1
801E 22EC83          SHLD PLYR2
8021 220380          SHLD ASI1
                ;player1
                REENT1:;LXI H,LED1
                ;?      LD IX,PLYR1
                ;?      LD IY,ASI1
8024 21EE83        LXI H,PLYR1;
8027 220B80        SHLD IXWK;
802A 210380        LXI H,ASI1;
802D 220D80        SHLD IYWK;
8030 21F883        LXI H,LED1;
8033 CDEA80     REENT2:CALL DTDP
8036 CD0C81          CALL BLNK
                ;"UP" or "STAY" disp
8039 E603       USDISP:ANI 03
```

```
803B 4F           MOV C,A;00 or 03 save
803C 3E01         MVI A,01;"UP"
803E CA4380       JZ USDISP1
8041 3E08         MVI A,08;"STAY"
8043 77       USDISP1:MOV M,A
8044 CD4881   KEY:CALL KEYIN1S
8047 FE15         CPI 15;write inc
8049 CA5980       JZ SAI
804C FE13         CPI 13;read dec("STAY")
804E CA3980       JZ USDISP
8051 FE14         CPI 14;read inc("UP")
8053 C24480       JNZ KEY
8056 C33980       JMP USDISP
              ;saikoro
8059 0D       SAI:DCR C
805A F28280       JP ASISET;C=03(keyin=RD_DEC)
805D 010002       LXI B,$0200
8060 CDCC80   SAI1:CALL RND3
              ;     ANI 07
8063 FE06         CPI 06
8065 D26080       JNC SAI1
8068 B7           ORA A
8069 CAA680       JZ DOWN
806C 3C           INR A
806D 81           ADD C
806E 27           DAA
806F 4F           MOV C,A
              ;?      DJNZ SAI1
8070 05         DCR B;
8071 C26080     JNZ SAI1;
              ;?      ADD A,(IX+00)
8074 E5         PUSH H;
8075 2A0B80     LHLD IXWK;
8078 86         ADD M;
8079 27           DAA
              ;     STA IX+00
807A 77         MOV M,A;
807B E1         POP H;
807C DAB380       JC WIN
807F C33380       JMP REENT2
              ;asiba set
              ASISET:;LDA IX+00
              ;     STA IY+00
8082 E5         PUSH H;
8083 2A0B80     LHLD IXWK;
8086 7E         MOV A,M;
8087 2A0D80     LHLD IYWK;
808A 77         MOV M,A;
808B E1         POP H;
              ;change player
808C 7D       ASISET1:MOV A,L
808D E607         ANI 07
808F C22480       JNZ REENT1;current HL=LED5
8092 21FC83       LXI H,LED5
              ;?      LD IX,PLYR2
              ;?      LD IY,ASI2
8095 E5         PUSH H;
8096 21EC83     LXI H,PLYR2;
```

```
8099 220B80      SHLD IXWK;
809C 210480      LXI H,ASI2;
809F 220D80      SHLD IYWK;
80A2 E1          POP H;
80A3 C33380        JMP REENT2
                 DOWN:;LDA IY+00
                 ;     STA IX+00
80A6 E5          PUSH H;
80A7 2A0D80      LHLD IYWK;
80AA 7E          MOV A,M
80AB 2A0B80      LHLD IXWK;
80AE 77          MOV M,A;
80AF E1          POP H;
80B0 C38C80        JMP ASISET1
                 ;? WIN:LD (IX+01),01
80B3 E5          WIN:PUSH H;
80B4 210B80      LXI H,IXWK;
80B7 3601        MVI M,01;
80B9 E1          POP H;
80BA CDEA80        CALL DTDP
80BD CDF780        CALL ALBLNK
80C0 0614          MVI B,14;=20 2sec wait
80C2 CD5881    WAIT:CALL TM01S
                 ;?      DJNZ WAIT
80C5 05          DCR B;
80C6 C2C280      JNZ WAIT;
80C9 C30F80        JMP START
                 ;
                 ;ransu
80CC E5          RND3:PUSH H
80CD D5            PUSH D
80CE 2A0880    RND2:LHLD RNDDT
80D1 23          INX H
80D2 7C          MOV A,H
80D3 E603        ANI 03
80D5 67          MOV H,A
80D6 220880      SHLD RNDDT
80D9 3A0A80      LDA RNDDT2
80DC 57          MOV D,A
80DD 7E            MOV A,M
80DE E607        ANI 07
80E0 BA            CMP D
80E1 CACE80      JZ RND2
80E4 320A80      STA RNDDT2
80E7 D1            POP D
80E8 E1            POP H
80E9 C9            RET
                 ;
                 ;data disp
80EA E5          DTDP:PUSH H
80EB CDA101        CALL ADDPS
80EE E1            POP H
80EF AF            XRA A
80F0 32F883        STA LED1
80F3 32FC83        STA LED5
80F6 C9            RET
                 ;
                 ;LED all blink
```

```
80F7 060A     ALBLNK:MVI B,0A;=10
80F9 3EEF     ALBLNK1:MVI A,EF
80FB D398        OUT 98
80FD CD5881        CALL TM01S
8100 3EFF          MVI A,FF
8102 D398        OUT 98
8104 CD5881        CALL TM01S
             ;?      DJNZ *ALBLNK1
8107 05          DCR B;
8108 C2F980      JNZ ALBLNK1;
810B C9            RET
             ;LED blink and keyin HL=LED1 or LED5
810C E5      BLNK:PUSH H;HL save
810D 23            INX H
810E E5            PUSH H
810F 110580        LXI D,BLNKBF
8112 010300        LXI B,$0003
             ;?      LDIR
8115 CD7381      CALL LDIR;
8118 E1      BLNK1:POP H
8119 E5            PUSH H
811A 110580        LXI D,BLNKBF
811D 0603          MVI B,03
811F 1A      BLNK2:LDAX D
8120 BE            CMP M
8121 C22581        JNZ BLNK3
8124 AF            XRA A
8125 77      BLNK3:MOV M,A
8126 23            INX H
8127 13            INX D
             ;?      DJNZ *BLNK2
8128 05          DCR B;
8129 C21F81      JNZ BLNK2;
812C CD5881        CALL TM01S
812F CD5081        CALL KEYIN2S
8132 FE13          CPI 13;read dec
8134 CA3C81        JZ BLNK4
8137 FE14          CPI 14;read inc
8139 C21881        JNZ BLNK1
813C D1      BLNK4:POP D
813D 210580        LXI H,BLNKBF
8140 010300        LXI B,$0003
             ;?      LDIR
8143 CD7381      CALL LDIR;
8146 E1            POP H
8147 C9            RET
             ;
             ;KEYIN with DE,BC save
8148 C5      KEYIN1S:PUSH B
8149 D5          PUSH D
814A CD1602      CALL KEYIN1
814D D1          POP D
814E C1          POP B
814F C9          RET
8150 C5      KEYIN2S:PUSH B
8151 D5          PUSH D
8152 CD2302      CALL KEYIN2
8155 D1          POP D
```

```
8156 C1          POP B
8157 C9          RET
                 ;
                 ;0.1sec timer
8158 D5       TM01S:PUSH D
8159 1E64        MVI E,64;=100
815B CD6481   TM01S2:CALL TM1M
815E 1D            DCR E
815F C25B81        JNZ TM01S2
8162 D1            POP D
8163 C9            RET
                 ;
                 ;1msec timer
8164 D5       TM1M:PUSH D;11
8165 1607        MVI D,07; ck=5 (11+7+292*7+10+10)/2.048=2082/2.048=1016
8167 1E12     TM1M2:MVI E,12;=18 ck=7 7+15*18+15=292
8169 1D       TM1M3:DCR E;5
816A C26981      JNZ TM1M3;10
816D 15          DCR D;5
816E C26781      JNZ TM1M2;10
8171 D1          POP D;10
8172 C9          RET;10
                 ;
8173 F5       LDIR:PUSH PSW
8174 7E       LDIR2:MOV A,M
8175 12          STAX D
8176 23          INX H
8177 13          INX D
8178 0B          DCX B
8179 78          MOV A,B
817A B1          ORA C
817B C27481      JNZ LDIR2
817E F1          POP PSW
817F C9          RET
                 ;
ADDPS        =01A1  ALBLNK       =80F7  ALBLNK1      =80F9
ASI1         =8003  ASI2         =8004  ASISET       =8082
ASISET1      =808C  BLNK         =810C  BLNK1        =8118
BLNK2        =811F  BLNK3        =8125  BLNK4        =813C
BLNKBF       =8005  DOWN         =80A6  DTDP         =80EA
IXWK         =800B  IYWK         =800D  KEY          =8044
KEYIN1       =0216  KEYIN1S      =8148  KEYIN2       =0223
KEYIN2S      =8150  LDIR         =8173  LDIR2        =8174
LED1         =83F8  LED5         =83FC  PLYR1        =83EE
PLYR2        =83EC  R            =FFD2  REENT1       =8024
REENT2       =8033  RND2         =80CE  RND3         =80CC
RNDDT        =8008  RNDDT2       =800A  SAI          =8059
SAI1         =8060  START        =800F  TM01S        =8158
TM01S2       =815B  TM1M         =8164  TM1M2        =8167
TM1M3        =8169  USDISP       =8039  USDISP1      =8043
WAIT         =80C2  WIN          =80B3
```

プログラムNo.13 WILD SEVEN(WILD78.BTK)

ゲームの説明

No.5 SWAPとNo.11 REVERSEを組み合わせたようなゲームです。

8000番地からRUNするとLEDに1～7の数と1個のブランク(スペース)がばらばらにならんでいます。これをルールにしたがって並べ替えて、1234567□にしてください。

ルールはNo.5 SWAPとほぼ同じです。スペースの隣にある数はスペースと入れ替えることができます。またスペースの1つ飛んで隣にある数も間の数をジャンプしてスペースと入れ替えることができます。

移動したい数の指定は、表示されているLEDの位置を示す1～8の数を押すことで行われます。1～8の数を押すとそのLED位置にある数が点滅します。

WRITE INCを押すことで入力が確定します。WRITE INCを押す前ならばLED位置の指定をやり直すことができます。WRITE INCを押すと指定したLEDにスペースが移動し、スペースがあった所に指定した場所の数が移ります。スペースと入れ替えができない位置を指定した場合にはWRITE INCを押しても移動は行われません。

この作業を繰り返して、1234567□(右端がスペース)が完成すると全体が2秒間点滅します。そのあと試行した回数が2秒間表示されてから、また始めに戻って再びゲームが開始されます。

プログラムの説明

プログラムはNo.5 SWAPとNo.11 REVERSEをミックスしたようなものなのでほとんど新たに説明するところはありません。REVERSEでは数値の表示を行うのにBCD数に直してLED表示データレジスタに入れて表示させていましたが今回は1桁ずつセグメントデータに変換して直接LED表示レジスタに入れています。REVERSEの方法ではスペースが0の表示になってしまうからです。

```
2016/4/20  11:17  wild78.txt
END=816F
                ; JUNNARABE for NDZ (WILD SEVEN)
                ; 03/05/02 5/5 5/14 5/23 5/24
                ;10/6/4 for ND80Z3
                ;9/17 9/18
                ;16/4/20 for nd8080
                ;
                     ORG $8000
                ;     RNDDT=$E810:$E811
                ;     CNTR=$E812
                     LED1=$83F8:$FFF8
                     DP1=$83F4:$FFF4
                     DP3=$83F6:$FFF6
                   R=$FFD2
                ;
                     DTDPS=$01C0:$05C0
                     KEYIN1=$0216:$0616
                     KEYIN2=$0223:$0623
                ;
8000 C30780        JMP START
8003 00         RNDDT:NOP
8004 00            NOP
8005 00         RNDDT2:NOP
8006 00         CNTR:NOP
                ;data buffer clear
8007 3AD2FF     START:LDA R
800A 210380        LXI H,RNDDT
800D 77            MOV M,A
800E 23            INX H
800F 77            MOV M,A
8010 3EFF          MVI A,FF
8012 0608            MVI B,08
8014 21F883          LXI H,LED1
8017 77         DTCLR:MOV M,A
```

```
8018 23            INX H
                ;?      DJNZ *DTCLR
8019 05           DCR B;
801A C21780       JNZ DTCLR;
                ;space position select
801D CD0A81         CALL RND3
                ;     ANI 07
8020 21F883         LXI H,LED1
8023 85             ADD L
8024 6F             MOV L,A
8025 3600           MVI M,00
                ;data set
8027 0607         MVI B,07
8029 CD0A81     DTST1:CALL RND3
                ;     ANI 07
802C B7           ORA A
802D CA2980       JZ DTST1
                ;?      LD IX,SEGDT
8030 216881       LXI H,SEGDT;
8033 1600           MVI D,00
8035 5F             MOV E,A
                ;?      ADD IX,DE
8036 19           DAD D;
                ;     LDA IX+00
8037 7E           MOV A,M;
8038 21F883         LXI H,LED1
803B 4E         DTST2:MOV C,M
803C 0C             INR C
803D C24880         JNZ DTST3
8040 77             MOV M,A
                ;?      DJNZ *DTST1
8041 05           DCR B;
8042 C22980       JNZ DTST1;
8045 C35380         JMP ENTRY
8048 BE         DTST3:CMP M
8049 CA2980         JZ DTST1
804C 2C             INR L
804D C23B80         JNZ DTST2
8050 C32980       JMP DTST1
                ;GAME START
8053 AF         ENTRY:XRA A
8054 320680         STA CNTR;DECIMAL COUNTER clear
                ;key entry(1-8)
8057 CD3D81     KEY:CALL KEYIN1S
805A FE09           CPI 09
805C D25780         JNC KEY
805F 3D             DCR A
8060 FA5780         JM KEY
                ;LED blink and wait WINC in
8063 21F883     BLNK0:LXI H,LED1
8066 85             ADD L
8067 6F             MOV L,A
8068 7E             MOV A,M
8069 B7             ORA A
806A CA5780         JZ KEY;space position
806D 57             MOV D,A;LED data save
806E 47             MOV B,A
806F 0E00           MVI C,00
```

```
8071 71       BLNK1:MOV M,C
8072 CD4D81         CALL TM01S
8075 CD4581         CALL KEYIN2S
8078 FE09           CPI 09
807A D28580         JNC BLNK2
807D 3D             DCR A
807E FA8A80         JM BLNK3
8081 72             MOV M,D
8082 C36380         JMP BLNK0
8085 FE15     BLNK2:CPI 15;WRITE INC
8087 CA9080         JZ CHECK
808A 79       BLNK3:MOV A,C
808B 48             MOV C,B
808C 47             MOV B,A
808D C37180         JMP BLNK1
              ;check
8090 E5       CHECK:PUSH H;save keyin position
8091 7D             MOV A,L
8092 E607           ANI 07
8094 CAA980         JZ CHECKR;position=LED1
8097 2B             DCX H
8098 7E             MOV A,M
8099 B7             ORA A
809A CAC780         JZ REP;left=space
809D 7D             MOV A,L
809E E607           ANI 07
80A0 CAA980         JZ CHECKR;left=LED1
80A3 2B             DCX H
80A4 7E             MOV A,M
80A5 B7             ORA A
80A6 CAC780         JZ REP;left of left=space
80A9 E1       CHECKR:POP H
80AA E5             PUSH H
80AB 7D             MOV A,L
80AC E607           ANI 07
80AE FE07           CPI 07
80B0 CA0681         JZ CANT;position=LED8,cannot replace
80B3 23             INX H
80B4 7E             MOV A,M
80B5 B7             ORA A
80B6 CAC780         JZ REP;right=space
80B9 7D             MOV A,L
80BA E607           ANI 07
80BC FE07           CPI 07
80BE CA0681         JZ CANT;right=LED8,cannot replace
80C1 23             INX H
80C2 7E             MOV A,M
80C3 B7             ORA A
80C4 C20681         JNZ CANT;right of right is not space,cannot replace
              ;replace
80C7 72       REP:MOV M,D
80C8 E1             POP H
80C9 3600           MVI M,00
80CB 3A0680         LDA CNTR;DECIMAL COUNTER up
80CE C601           ADI 01
80D0 27             DAA
80D1 320680         STA CNTR
              ;all check
```

```
80D4 21F883        LXI H,LED1
80D7 0607          MVI B,07
                ;?      LD IX,SEGDT1
80D9 116981       LXI D,SEGDT1;
                ALCHK1:;LDA IX+00
80DC 1A           LDAX D;
80DD BE             CMP M
80DE C25780         JNZ KEY;not done
80E1 23             INX H
                ;?      INC IX
80E2 13           INX D;
                ;?      DJNZ *ALCHK1
80E3 05           DCR B;
80E4 C2DC80       JNZ ALCHK1;
                ;OK! ALL LED blink
80E7 CD2881         CALL ALBLNK
                ;COUNTER disp
80EA 210000         LXI H,$0000
80ED 22F483         SHLD DP1
80F0 3A0680         LDA CNTR
80F3 67             MOV H,A
80F4 22F683         SHLD DP3
80F7 CDC001         CALL DTDPS
80FA 0614           MVI B,14;=20 ,2sec wait
80FC CD4D81    WAIT:CALL TM01S
                ;?      DJNZ WAIT
80FF 05           DCR B;
8100 C2FC80       JNZ WAIT;
8103 C30780         JMP START
                ;cannot replace
8106 E1        CANT:POP H
8107 C38A80         JMP BLNK3
                ;
                ;ransu
810A E5        RND3:PUSH H
810B D5             PUSH D
810C 2A0380    RND2:LHLD RNDDT
810F 23           INX H
8110 7C           MOV A,H
8111 E603         ANI 03
8113 67           MOV H,A
8114 220380       SHLD RNDDT
8117 3A0580       LDA RNDDT2
811A 57           MOV D,A
811B 7E             MOV A,M
811C E607         ANI 07
811E BA             CMP D
811F CA0C81       JZ RND2
8122 320580       STA RNDDT2
8125 D1             POP D
8126 E1             POP H
8127 C9             RET
                ;
                ;LED all blink
8128 060A      ALBLNK:MVI B,0A;=10
812A 3EEF      ALBLNK1:MVI A,EF
812C D398         OUT 98
812E CD4D81         CALL TM01S
```

```
8131 3EFF          MVI A,FF
8133 D398        OUT 98
8135 CD4D81        CALL TM01S
             ;?      DJNZ *ALBLNK1
8138 05          DCR B;
8139 C22A81      JNZ ALBLNK1;
813C C9            RET
             ;
             ;KEYIN with DE,BC save
813D C5      KEYIN1S:PUSH B
813E D5          PUSH D
813F CD1602      CALL KEYIN1
8142 D1          POP D
8143 C1          POP B
8144 C9          RET
8145 C5      KEYIN2S:PUSH B
8146 D5          PUSH D
8147 CD2302      CALL KEYIN2
814A D1          POP D
814B C1          POP B
814C C9          RET
             ;
             ;0.1sec timer
814D D5      TM01S:PUSH D
814E 1E64          MVI E,64;=100
8150 CD5981  TM01S2:CALL TM1M
8153 1D            DCR E
8154 C25081        JNZ TM01S2
8157 D1            POP D
8158 C9            RET
             ;
             ;1msec timer
8159 D5      TM1M:PUSH D;11
815A 1607        MVI D,07; ck=5 (11+7+292*7+10+10)/2.048=2082/2.048=1016
815C 1E12    TM1M2:MVI E,12;=18 ck=7 7+15*18+15=292
815E 1D      TM1M3:DCR E;5
815F C25E81      JNZ TM1M3;10
8162 15          DCR D;5
8163 C25C81      JNZ TM1M2;10
8166 D1          POP D;10
8167 C9          RET;10
             ;
8168 00      SEGDT:DB 00;space
8169 06      SEGDT1:DB 06;1
816A 5B            DB 5B;2
816B 4F            DB 4F;3
816C 66            DB 66;4
816D 6D            DB 6D;5
816E 7D            DB 7D;6
816F 27            DB 27;7
             ;
ALBLNK      =8128  ALBLNK1     =812A  ALCHK1      =80DC
BLNK0       =8063  BLNK1       =8071  BLNK2       =8085
BLNK3       =808A  CANT        =8106  CHECK       =8090
CHECKR      =80A9  CNTR        =8006  DP1         =83F4
DP3         =83F6  DTCLR       =8017  DTDPS       =01C0
DTST1       =8029  DTST2       =803B  DTST3       =8048
ENTRY       =8053  KEY         =8057  KEYIN1      =0216
```

```
KEYIN1S     =813D  KEYIN2      =0223  KEYIN2S     =8145
LED1        =83F8  R           =FFD2  REP         =80C7
RND2        =810C  RND3        =810A  RNDDT       =8003
RNDDT2      =8005  SEGDT       =8168  SEGDT1      =8169
START       =8007  TM01S       =814D  TM01S2      =8150
TM1M        =8159  TM1M2       =815C  TM1M3       =815E
WAIT        =80FC
```

プログラム№14　MOO（MOO8．BTK）

ゲームの説明

　MOOというのは牛の鳴き声（モー）です。別名BULLS　＆　COWSといって昔イギリスではやったゲームだそうです（「マイコンゲーム21」から。私は知りません）。

　コンピュータが乱数を使って4桁の数を隠しています。4桁の数には同じ数は含まれません（5074や3491はありうるが2335などはない）。

　8000番地からRUNするとブーと音がしてLEDが－－－－0000という表示になります。ゲームスタートです。コンピュータが隠している数を想像して4ケタの数を入力します。WRITE　INCを押すことで入力が確定します。WRITE　INCを押す前ならば数を入れ直すことができます。

コンピュータは入力された数と隠している数を比較して、同じ数が同じ桁位置にある場合、それを1頭の牡牛（BULL）と数えます。4桁とも一致したときは牡牛が4頭いることになります。位置は異なるけれど同じ数が含まれている場合、それを牝牛（COW）の数で示します。牡牛は”b”、牝牛は”C”の文字を使ってその頭数が、プーという音とともにLEDの左4桁に表示されます。全く該当しなかった場合には表示はブランクになります。この表示をヒントにしてできるだけ少ない回数で正解することを競うゲームです。

　テクニックとしてわざと同じ数の並びを入力するなどの方法も考えられますが、ルール違反とした方がよいでしょう（プログラムでチェックはしていません）。

　みごと正解するとピーという音がしてLEDの左4桁がbbbbになり、LED全部が2秒間点滅します。そのあと試行回数が2秒間表示され、また始めに戻って再びゲームが開始されます。

　途中でギブアップするときは”F”キーを押します。LEDの左4桁にコンピュータが隠していた数が表示されます。2秒たつとまた始めに戻って再びゲームが開始されます。

プログラムの説明

　4桁の数はBCD数ではなく、4バイトのセグメントデータに変換してデータバッファに格納します。各桁ごとに比較が必要なのでBCD数では処理に手間がかかります。№13　WILD　SEVENと同様、表示にブランク桁が含まれる場合があるため、表示する場合には直接セグメントに変換して表示する方が楽です。ちょっと意外な気がしますが、数の大小比較や計算はしないので、最初からセグメントデータに変換しておいても支障はないのです。 変換はCNVTサブルーチンで行います。

　最初に4桁のデータを作成します（;data set）。1桁目に0が来ると4桁の数になりませんから、そのチェックを行っています。Lに00を入れてスタートし、RNDで選んだデータが0のときに、Lをチェックしています。DTBFのトップアドレスの下位アドレスが00であることを利用したテクニックです。DTST4:でデータを格納したあとINC　Lをしていますが、本来不要な処理です。これを追加することによって、L＝00であるのはまだ最初の桁にデータがセットされていない（つまり1桁目の処理）であることが分かります。

　牡牛と牝牛のチェック（COMP:）はちょっと複雑です。牡牛からチェックするのがポイントです。先頭の桁から比較していって、データが一致したら、LEDに”b”を表示します。このときDTBFのデータのbit7を1にします。次の牝牛のチェックで除外するためのマークです。bit7はLED表示ではピリオドになっていて、このプログラムでのデータでは使わないビットなのでマークとして利用できます。さらにビット7＝1のときその数はマイナスの数として扱われるため、Sフラグの有無で判別が可能です。Cレジスタを牡牛用のダウンカウンタに使います。C＝04でスタートしデータが一致するごとに－1します。C＝00になれば4桁とも一致したことになりゲーム終了です。牡牛のチェックが終ったら、牝牛をチェックします。牡牛の場合と似ていますが、桁の違いを考慮しないため、DTBFの各桁ごとに、LED5～8のデータと照合します。このときbit7＝1のデータは除外すると同時に、そのマークを取り除きます（bit7＝0に戻す）。データが一致するごとにLEDに”C”を表示することは牡牛の処理と同じです。

```
2016/4/20  12:26  moo8.txt
END=81E0
              ; MOO for NDZ
              ; 03/05/06 5/15
              ;10/6/4 for ND80Z3
              ;9/17
              ;16/4/20 for nd8080
              ;
                   ORG $8000
              ;     DTBF=$E800
              ;     DTBF2=$E802
              ;     RNDDT=$E810:$E811
                   LED1=$83F8:$FFF8
                   LED3=$83FA:$FFFA
```

```
                    LED5=$83FC;$FFFC
                    LED8=$83FF;$FFFF
                    DP1=$83F4;$FFF4
                    DP3=$83F6;$FFF6
                    CNTR=$83F7;$FFF7;=DP4
                  R=$FFD2
               ;
                    DTDPS=$01C0;$05C0
                    KEYIN1=$0216;$0616
                          D1=$02DD;4.684msec
               ;
8000 C30A80       JMP START
8003 00        DTBF:NOP
8004 00           NOP
8005 00        DTBF2:NOP
8006 00           NOP
8007 00        RNDDT:NOP
8008 00           NOP
8009 00        RNDDT2:NOP
               ;data set
800A 3AD2FF    START:LDA R
800D 210780       LXI H,RNDDT
8010 77           MOV M,A
8011 23           INX H
8012 77           MOV M,A
8013 21FFFF       LXI H,$FFFF
8016 220380         SHLD DTBF
8019 220580         SHLD DTBF2
               ;data set
801C 2E00           MVI L,00
801E CD2281    DTST1:CALL RND4
               ;     ANI 0F
8021 FE0A           CPI 0A
8023 D21E80         JNC DTST1
8026 B7             ORA A
8027 C22E80         JNZ DTST2
802A BD             CMP L
802B CA1E80         JZ DTST1;first No. is not "0"
802E CD4081    DTST2:CALL CNVT
8031 210380         LXI H,DTBF
8034 0604           MVI B,04
8036 BE        DTST3:CMP M
8037 CA1E80         JZ DTST1;already selected No.
803A 4E             MOV C,M
803B 0C             INR C
803C CA4480         JZ DTST4;(HL)=FF
803F 23             INX H
               ;?      DJNZ *DTST3;B is never 0
8040 05           DCR B;
8041 C33680       JMP DTST3;
8044 77        DTST4:MOV M,A
8045 2C             INR L;for first No. check
               ;?      DJNZ *DTST1
8046 05           DCR B;
8047 C21E80       JNZ DTST1;
               ;GAME START
804A 210000         LXI H,$0000
804D 22F683         SHLD DP3;DECIMAL COUNTER clear
```

```
8050 22F483        SHLD DP1
                ;key entry
8053 CD5E81        CALL STDP
8056 3E00          MVI A,00;buuuu
8058 0605          MVI B,05
805A CD9381     START1:CALL SOUND
                ;?      DJNZ START1
805D 05            DCR B;
805E C25A80        JNZ START1;
8061 CD7D81     KEY:CALL KEYIN1S
8064 FE15          CPI 15;WRITE INC
8066 CA8680        JZ COMP
8069 FE0F          CPI 0F;"F"
806B CAF480        JZ GIVUP
806E D26180        JNC KEY
8071 11FC83        LXI D,LED5
8074 62            MOV H,D
8075 6B            MOV L,E
8076 23            INX H
8077 010300        LXI B,$0003
                ;?      LDIR
807A CDD481        CALL LDIR;
807D CD4081        CALL CNVT
8080 32FF83        STA LED8
8083 C36180        JMP KEY
                ;DATA compere
                ;BULL check
8086 3AF783     COMP:LDA CNTR
8089 C601          ADI 01;DECIMAL COUNTER up
808B 27            DAA
808C 32F783        STA CNTR
808F 210000        LXI H,$0000
8092 22F883        SHLD LED1
8095 22FA83        SHLD LED3
8098 21F883        LXI H,LED1;
809B E3            XTHL;
809C 210380        LXI H,DTBF
809F 11FC83        LXI D,LED5
                ;?      LD IX,LED1
80A2 010404        LXI B,$0404
80A5 1A         COMP1:LDAX D
80A6 BE            CMP M
80A7 C2B880        JNZ COMP2
                ;?      LD (IX+00),7C;b
80AA E3            XTHL;
80AB 367C          MVI M,7C;b
80AD E3            XTHL;
80AE 0D            DCR C
80AF CA0581        JZ OK
80B2 F680          ORI 80
80B4 77            MOV M,A;B "bull" mark set
                ;?      INC IX
80B5 E3            XTHL;
80B6 23            INX H;
80B7 E3            XTHL;
80B8 23         COMP2:INX H
80B9 13            INX D
                ;?      DJNZ *COMP1
```

```
80BA 05          DCR B;
80BB C2A580      JNZ COMP1;
                 ;COW check
80BE 110380          LXI D,DTBF
80C1 0604            MVI B,04
80C3 0E04        COMP3:MVI C,04
80C5 21FC83          LXI H,LED5
80C8 1A              LDAX D
80C9 B7              ORA A
80CA F2D380          JP COMP4
80CD E67F            ANI 7F
80CF 12              STAX D;"bull" mark clear
80D0 C3E180          JMP COMP6
80D3 BE          COMP4:CMP M
80D4 C2DC80          JNZ COMP5
                 ;?      LD (IX+00),39;C
80D7 E3          XTHL;
80D8 3639        MVI M,39;C
                 ;?      INC IX
80DA 23          INX H;
80DB E3          XTHL;
80DC 23          COMP5:INX H
80DD 0D              DCR C
80DE C2D380          JNZ COMP4
80E1 13          COMP6:INX D
                 ;?      DJNZ *COMP3
80E2 05          DCR B;
80E3 C2C380      JNZ COMP3;
80E6 3E0A            MVI A,0A
80E8 0605            MVI B,05
80EA CD9381      COMP8:CALL SOUND
                 ;?      DJNZ *COMP8
80ED 05          DCR B
80EE C2EA80      JNZ COMP8;
80F1 C36180          JMP KEY
                 ;give up
80F4 210380      GIVUP:LXI H,DTBF
80F7 11F883          LXI D,LED1
80FA 010400          LXI B,$0004
                 ;?      LDIR
80FD CDD481      CALL LDIR;
8100 0628            MVI B,28;=40 ,4sec wait
8102 C31881          JMP WAIT
8105 3E17        OK:MVI A,17
8107 060A            MVI B,0A
8109 CD9381      OK1:CALL SOUND
                 ;?      DJNZ OK1
810C 05          DCR B;
810D C20981      JNZ OK1;
8110 CD4981          CALL ALBLNK
8113 CDC001          CALL DTDPS;COUNTER disp
8116 0614            MVI B,14;=20 ,2sec wait
8118 CD8581      WAIT:CALL TM01S
                 ;?      DJNZ WAIT
811B 05          DCR B;
811C C21881      JNZ WAIT;
811F C30A80          JMP START
                 ;
```

```
                ;ransu
8122 E5         RND4:PUSH H
8123 D5              PUSH D
8124 2A0780     RND2:LHLD RNDDT
8127 23            INX H
8128 7C            MOV A,H
8129 E603          ANI 03
812B 67            MOV H,A
812C 220780        SHLD RNDDT
812F 3A0980        LDA RNDDT2
8132 57            MOV D,A
8133 7E              MOV A,M
8134 E60F          ANI 0F
8136 BA              CMP D
8137 CA2481        JZ RND2
813A 320980        STA RNDDT2
813D D1              POP D
813E E1              POP H
813F C9              RET
                ;
                ;from key data to segment data convert
8140 217381     CNVT:LXI H,SEGDT
8143 5F              MOV E,A
8144 1600            MVI D,00
8146 19              DAD D
8147 7E              MOV A,M
8148 C9              RET
                ;
                ;LED all blink
8149 060A       ALBLNK:MVI B,0A;=10
814B 3EEF       ALBLNK1:MVI A,EF
814D D398          OUT 98
814F CD8581          CALL TM01S
8152 3EFF            MVI A,FF
8154 D398          OUT 98
8156 CD8581          CALL TM01S
                ;?       DJNZ *ALBLNK1
8159 05            DCR B;
815A C24B81        JNZ ALBLNK1;
815D C9              RET
                ;
                ;start data disp
815E 216B81     STDP:LXI H,STDT
8161 11F883          LXI D,LED1
8164 010800          LXI B,$0008
                ;?       LDIR
8167 CDD481        CALL LDIR;
816A C9              RET
816B 40         STDT:DB 40
816C 40              DB 40
816D 40              DB 40
816E 40              DB 40
816F 5C              DB 5C
8170 5C              DB 5C
8171 5C              DB 5C
8172 5C              DB 5C
                ;
8173 5C         SEGDT:DB 5C;0
```

```
8174 06            DB 06;1
8175 5B            DB 5B;2
8176 4F            DB 4F;3
8177 66            DB 66;4
8178 6D            DB 6D;5
8179 7D            DB 7D;6
817A 27            DB 27;7
817B 7F            DB 7F;8
817C 6F            DB 6F;9
                ;
                ;KEYIN with DE,BC save
817D C5         KEYIN1S:PUSH B
817E D5            PUSH D
817F CD1602        CALL KEYIN1
8182 D1            POP D
8183 C1            POP B
8184 C9            RET
                ;
                ;0.1sec timer
8185 D5         TM01S:PUSH D
8186 1E15          MVI E,15;=21
8188 D5         TM01S2:PUSH D
8189 CDDD02        CALL D1;4.684ms
818C D1            POP D
818D 1D              DCR E
818E C28881          JNZ TM01S2
8191 D1              POP D
8192 C9              RET
                ;
                ;
8193 F5         SOUND:PUSH PSW
8194 E5            PUSH H
8195 D5            PUSH D
8196 C5            PUSH B
8197 21BC81        LXI H,SNDTBL
819A 85            ADD L
819B 6F            MOV L,A
819C 46            MOV B,M
819D 1E1A          MVI E,1A
819F 50         SNDS1:MOV D,B
81A0 3EFF          MVI A,FF;SP OUT=H
81A2 D398          OUT 98
81A4 7F         SNDS2:MOV A,A;5---------
81A5 15            DCR D;5          | 5+5+10=20  20/2=10microsec
81A6 C2A481        JNZ SNDS2;10----
81A9 50            MOV D,B
81AA 3EDF          MVI A,DF;sp out=L
81AC D398          OUT 98
81AE 7F         SNDS3:MOV A,A;5---------
81AF 15            DCR D;5          | 5+5+10=20  20/2=10microsec
81B0 C2AE81        JNZ SNDS3;10----
81B3 1D            DCR E
81B4 C29F81        JNZ SNDS1
81B7 C1            POP B
81B8 D1            POP D
81B9 E1            POP H
81BA F1            POP PSW
81BB C9            RET
```

```
                ; SOUND TABLE
81BC 7F         SNDTBL:DB 7F;so4
81BD 77            DB 77;so#4
81BE 71            DB 71;ra4
81BF 6A            DB 6A;ra#4
81C0 5F            DB 5F;do5
81C1 59            DB 59;do#5
81C2 54            DB 54;re5
81C3 4F            DB 4F;re#
81C4 47            DB 47;fA5
81C5 43            DB 43;fa#5
81C6 3F            DB 3F;so5
81C7 3B            DB 3B;so5#
81C8 35            DB 35;ra#5
81C9 32            DB 32;si5
81CA 2F            DB 2F;do6
81CB 2C            DB 2C;do#6
81CC 25            DB 25;mi6
81CD 27            DB 27;re#6
81CE 2A            DB 2A;re6
81CF 4B            DB 4B;mi5
81D0 38            DB 38;ra5
81D1 64            DB 64;si4
81D2 23            DB 23;fa6
81D3 21            DB 21;fa#6
                ;
81D4 F5         LDIR:PUSH PSW
81D5 7E         LDIR2:MOV A,M
81D6 12            STAX D
81D7 23            INX H
81D8 13            INX D
81D9 0B            DCX B
81DA 78            MOV A,B
81DB B1            ORA C
81DC C2D581        JNZ LDIR2
81DF F1            POP PSW
81E0 C9            RET
                ;
ALBLNK      =8149  ALBLNK1     =814B  CNTR        =83F7
CNVT        =8140  COMP        =8086  COMP1       =80A5
COMP2       =80B8  COMP3       =80C3  COMP4       =80D3
COMP5       =80DC  COMP6       =80E1  COMP8       =80EA
D1          =02DD  DP1         =83F4  DP3         =83F6
DTBF        =8003  DTBF2       =8005  DTDPS       =01C0
DTST1       =801E  DTST2       =802E  DTST3       =8036
DTST4       =8044  GIVUP       =80F4  KEY         =8061
KEYIN1      =0216  KEYIN1S     =817D  LDIR        =81D4
LDIR2       =81D5  LED1        =83F8  LED3        =83FA
LED5        =83FC  LED8        =83FF  OK          =8105
OK1         =8109  R           =FFD2  RND2        =8124
RND4        =8122  RNDDT       =8007  RNDDT2      =8009
SEGDT       =8173  SNDS1       =819F  SNDS2       =81A4
SNDS3       =81AE  SNDTBL      =81BC  SOUND       =8193
START       =800A  START1      =805A  STDP        =815E
STDT        =816B  TM01S       =8185  TM01S2      =8188
WAIT        =8118
```

プログラム№15　SWAP2（SWAP28．BTK）

ゲームの説明

№5　SWAPを発展させたものです。SWAPがちょっと単純なので新しく作りました。「マイコンゲーム21」にはありません。

SWAPと違ってLEDは8桁全部を使います。

8000番地からRUNするとLEDに”8”の下半分と”8”の上半分がランダムに並び（それぞれの数も毎回異なる）、それとともにどこかに1桁のブランクが表示されます。

ルールにしたがって表示を移動し、ブランクをはさんで左側に”8”の下半分を集め、右側に”8”の上半分を集めるとゲーム終了です。入れ替えに成功すると2秒間表示が点滅し、さらに2秒間試行回数が表示されたあと、最初に戻って再びゲームが開始されます。

ルールはSWAPと同じです。ブランクの隣の表示はブランクと位置を交代することができます。また隣に異なる表示があるとき、その表示をはさんでさらにその隣がブランクならば、間の表示を飛び越してブランクと表示位置を交代することができます。

入れ替えたいLED位置に対応する数をKEY入力するとその位置のLEDが点滅します。それでよければWRITE　INCキーを押します。WRITE　INCキーを押す前なら別の数を入れ直して位置の指定を変更することができます。WRITE　INCを押すとルールにしたがって表示とブランクが入れ替わります。ルールに合わない場合には点滅が続きますが入れ替えは行われません。

プログラムの説明

特に説明しなくても理解できると思います。

```
2016/4/20  13:55  swap28.txt
END=8173
                ; SWAP2 for NDZ
                ; 03/05/03 5/5 5/15
                ;10/6/4 for ND80Z3
                ;16/4/20 for nd8080
                ;
                     ORG $8000
                ;     RNDDT=$E810;$E811
                     LED1=$83F8;$FFF8
                     DP1=$83F4;$FFF4
                     DP3=$83F6;$FFF6
                     CNTR=$83F7;$FFF7;=DP4
                   R=$FFD2
                ;
                     DTDPS=$01C0;$05C0
                     KEYIN1=$0216;$0616
                     KEYIN2=$0223;$0623
                     D1=$02DD;4.684msec
                ;
8000 C30680        JMP START
8003 00         RNDDT:NOP
8004 00            NOP
8005 00         RNDDT2:NOP
                ;data buffer clear
8006 3AD2FF     START:LDA R
8009 210380        LXI H,RNDDT
800C 77            MOV M,A
800D 23            INX H
800E 77            MOV M,A
800F 3EEF          MVI A,EF
8011 D398          OUT 98
8013 3EFF            MVI A,FF
8015 0608            MVI B,08
8017 21F883          LXI H,LED1
```

```
801A 77       DTCLR:MOV M,A
801B 23            INX H
              ;?      DJNZ *DTCLR
801C 05          DCR B;
801D C21A80      JNZ DTCLR;
              ;space position select
8020 CD2381        CALL RND3
              ;     ANI 07
8023 21F883        LXI H,LED1
8026 85            ADD L
8027 6F            MOV L,A
8028 3600          MVI M,00
              ;"UP" No. select
802A CD2381   DTST1:CALL RND3
              ;  ANI 07
802D FE06          CPI 06
802F D22A80        JNC DTST1
8032 3C            INR A;from 1 to 6 select
8033 47            MOV B,A
              ;"UP" data position select
8034 CD2381   DTST2:CALL RND3
              ;     ANI 07
8037 21F883        LXI H,LED1
803A 85            ADD L
803B 6F            MOV L,A
803C 7E            MOV A,M
803D B7            ORA A
803E CA3480        JZ DTST2;space position
8041 3C            INR A
8042 C23480        JNZ DTST2;already used
8045 3663          MVI M,63;"UP"
              ;?      DJNZ *DTST2
8047 05          DCR B;
8048 C23480      JNZ DTST2;
              ;"down" data set
804B 21F883        LXI H,LED1
804E 0608          MVI B,08
8050 7E       DTST3:MOV A,M
8051 B7            ORA A
8052 CA5B80        JZ DTST4;space position
8055 3C            INR A
8056 C25B80        JNZ DTST4;alredy used
8059 365C          MVI M,5C;"DOWN"
805B 23       DTST4:INX H
              ;?      DJNZ *DTST3
805C 05          DCR B;
805D C25080      JNZ DTST3;
              ;
8060 210000        LXI H,$0000
8063 22F683        SHLD DP3;DECIMAL COUNTER clear
8066 22F483        SHLD DP1
8069 3EFF          MVI A,FF
806B D398        OUT 98
              ;key entry
806D CD5681   KEY:CALL KEYIN1S;1-8 input
8070 FE09          CPI 09
8072 D26D80        JNC KEY
8075 3D            DCR A
```

```
8076 FA6D80         JM KEY
                ;LED blink and wait WINC in
8079 21F883     BLNK0:LXI H,LED1
807C 85             ADD L
807D 6F             MOV L,A
807E 7E             MOV A,M
807F B7             ORA A
8080 CA6D80         JZ KEY;space position
8083 57             MOV D,A;LED data save
8084 47             MOV B,A
8085 0E00           MVI C,00
8087 71         BLNK1:MOV M,C
8088 CD6681         CALL TM01S
808B CD5E81         CALL KEYIN2S;1-8,15 input
808E FE09           CPI 09
8090 D29B80         JNC BLNK2
8093 3D             DCR A
8094 FAA080         JM BLNK3
8097 72             MOV M,D
8098 C37980         JMP BLNK0
809B FE15       BLNK2:CPI 15;WRITE INC
809D CAA680         JZ CHECK
80A0 79         BLNK3:MOV A,C
80A1 48             MOV C,B
80A2 47             MOV B,A
80A3 C38780         JMP BLNK1
                ;check
80A6 7D         CHECK:MOV A,L
80A7 E5             PUSH H;save HL(current position)
80A8 E607           ANI 07
80AA CAC380         JZ CHECKR;current position=LED1
80AD 2B             DCX H
80AE 7E             MOV A,M
80AF B7             ORA A
80B0 CADF80         JZ REP;left=space
80B3 BA             CMP D
80B4 CAC380         JZ CHECKR;left=same
80B7 7D             MOV A,L
80B8 E607           ANI 07
80BA CAC380         JZ CHECKR;left=LED1
80BD 2B             DCX H
80BE 7E             MOV A,M
80BF B7             ORA A
80C0 CADF80         JZ REP;left of left=space
80C3 E1         CHECKR:POP H
80C4 E5             PUSH H
80C5 7D             MOV A,L
80C6 3C             INR A
80C7 CA1F81         JZ CANT;position=L8,cannot replace
80CA 23             INX H
80CB 7E             MOV A,M
80CC B7             ORA A
80CD CADF80         JZ REP;right=space
80D0 BA             CMP D
80D1 CA1F81         JZ CANT;right=same,cannot replace
80D4 7D             MOV A,L
80D5 3C             INR A
80D6 CA1F81         JZ CANT;right=L8,cannot replace
```

```
80D9 23           INX H
80DA 7E           MOV A,M
80DB B7           ORA A
80DC C21F81       JNZ CANT;right of right is not space,cannot replace
                  ;replace
80DF 72       REP:MOV M,D
80E0 E1           POP H
80E1 3600         MVI M,00
80E3 3AF783       LDA CNTR;DECIMAL COUNTER up
80E6 C601         ADI 01
80E8 27           DAA
80E9 32F783       STA CNTR
                  ;all check
80EC 21F883       LXI H,LED1
80EF 0608         MVI B,08
80F1 7E       ALCHK1:MOV A,M
80F2 FE5C         CPI 5C;"DOWN"
80F4 C2FC80       JNZ ALCHK2
80F7 23           INX H
                  ;?      DJNZ *ALCHK1
80F8 05           DCR B;
80F9 C2F180       JNZ ALCHK1;
80FC B7       ALCHK2:ORA A
80FD C26D80       JNZ KEY;not success
8100 23           INX H
8101 05           DCR B
8102 7E       ALCHK3:MOV A,M
8103 FE63         CPI 63;"UP"
8105 C26D80       JNZ KEY;not success
8108 23           INX H
                  ;?      DJNZ *ALCHK3
8109 05           DCR B;
810A C20281       JNZ ALCHK3;
                  ;OK! ALL LED blink and COUNTER disp
810D CD4181       CALL ALBLNK
8110 CDC001       CALL DTDPS
8113 0614         MVI B,14;=20 ,2sec wait
8115 CD6681   WAIT:CALL TM01S
                  ;?      DJNZ WAIT
8118 05           DCR B;
8119 C21581       JNZ WAIT;
811C C30680       JMP START
                  ;cannot replace
811F E1       CANT:POP H
8120 C3A080       JMP BLNK3
                  ;
                  ;ransu
8123 E5       RND3:PUSH H
8124 D5           PUSH D
8125 2A0380   RND2:LHLD RNDDT
8128 23           INX H
8129 7C           MOV A,H
812A E603         ANI 03
812C 67           MOV H,A
812D 220380       SHLD RNDDT
8130 3A0580       LDA RNDDT2
8133 57           MOV D,A
8134 7E           MOV A,M
```

```
8135 E607        ANI 07
8137 BA            CMP D
8138 CA2581      JZ RND2
813B 320580      STA RNDDT2
813E D1            POP D
813F E1            POP H
8140 C9            RET
                 ;
                 ;LED all blink
8141 060A      ALBLNK:MVI B,0A;=10
8143 3EEF      ALBLNK1:MVI A,EF
8145 D398        OUT 98
8147 CD6681        CALL TM01S
814A 3EFF          MVI A,FF
814C D398        OUT 98
814E CD6681        CALL TM01S
                 ;?      DJNZ *ALBLNK1
8151 05          DCR B;
8152 C24381      JNZ ALBLNK1;
8155 C9            RET
                 ;
                 ;KEYIN with DE,BC save
8156 C5        KEYIN1S:PUSH B
8157 D5          PUSH D
8158 CD1602      CALL KEYIN1
815B D1          POP D
815C C1          POP B
815D C9          RET
                 ;
815E C5        KEYIN2S:PUSH B
815F D5          PUSH D
8160 CD2302      CALL KEYIN2
8163 D1          POP D
8164 C1          POP B
8165 C9          RET
                 ;
                 ;0.1sec timer
8166 D5        TM01S:PUSH D
8167 1E15          MVI E,15;=21
8169 D5        TM01S2:PUSH D
816A CDDD02      CALL D1;4.684ms
816D D1          POP D
816E 1D            DCR E
816F C26981        JNZ TM01S2
8172 D1            POP D
8173 C9            RET
                 ;
ALBLNK       =8141  ALBLNK1      =8143  ALCHK1       =80F1
ALCHK2       =80FC  ALCHK3       =8102  BLNK0        =8079
BLNK1        =8087  BLNK2        =809B  BLNK3        =80A0
CANT         =811F  CHECK        =80A6  CHECKR       =80C3
CNTR         =83F7  D1           =02DD  DP1          =83F4
DP3          =83F6  DTCLR        =801A  DTDPS        =01C0
DTST1        =802A  DTST2        =8034  DTST3        =8050
DTST4        =805B  KEY          =806D  KEYIN1       =0216
KEYIN1S      =8156  KEYIN2       =0223  KEYIN2S      =815E
LED1         =83F8  R            =FFD2  REP          =80DF
RND2         =8125  RND3         =8123  RNDDT        =8003
```

```
RNDDT2      =8005  START       =8006  TM01S       =8166
TM01S2      =8169  WAIT        =8115
```

第2部　電子オルゴール

1．　電子オルゴールプログラム

　MUSIC8．BTKは自動演奏プログラムです。
　電子オルガンプログラム　SOUND8．BTKはキーを押すことでオルガンのように音を出すプログラムですが、MUSIC8．BTKはオルゴールのように自動演奏をします。

ⅰ)プログラムの使い方

　nd8080フォルダにあるMUSIC8．BTKを、ND8080（ND80Zモニタ）に、HIDWR8コマンドで送信してください。
　プログラムは8000番地からスタートしますが、その前に演奏データが必要です。
　サンプルとして、演奏データファイルが、nd8080フォルダに入っています。
　一度に1曲しか送ることはできません。
　下記のうちどれかをHIDWR8コマンドでND8080（TK80モニタモード）に送信してください。

　KINJIRA．BTK（速さのパラメータ03）
　KOKYO．BTK（速さのパラメータ03）
　NEKO．BTK（速さのパラメータ03）
　SEIJA．BTK（速さのパラメータ02）
　SEIYA．BTK（速さのパラメータ04）
　YOROKOBI．BTK（速さのパラメータ03）
　SUZANNA．BTK（速さのパラメータ01）
　ELISE．BTK（速さのパラメータ03）

　曲データファイルは8100番地からLOADされます。
　8100番地に上書き送信することで、別の曲を演奏させることができます。
　曲によっては速さのパラメータ（アドレス8003の値）を書き換える必要があります。プログラムをロードした直後は03になっています。必要に応じて上の（）の値に変更してから、プログラムを実行してください。
　プログラムと曲ファイルをLOADしたら、[8][0][0][0][ADRSSET][RUN]と入力すると自動演奏がはじまります。
　演奏が終了してもLED表示はアドレス8000を表示したままモニタプログラムに戻りますから、そこでまた[RUN]を押せば、ふたたび自動演奏がはじまります。
　演奏終了後かまたはリセット後に、別の曲ファイルをLOADすると、8100からのメモリエリアが新しい曲データで上書きされ、別の曲を演奏させることができます。
　8000番地からのMUSICプログラムを書き換えてしまわない限りは、自動演奏を繰り返し実行させることができます。

ⅱ)曲データコード

　曲データサンプルに限らず、楽譜さえあれば任意の曲データを書いて演奏することができます。データは音の高さのコードと長さのコードで構成されます。最初のデータ（偶数アドレス）が音の高さのデータで、後のデータ（奇数アドレス）は音の長さのデータです。
　休符も高さコードの1種として扱います。コード00を使います。
　8100番地から、楽譜を見ながら音符データを書き込んでいくことで、任意の曲を演奏させることができます。
　曲の最後（偶数アドレス）にはFFを書きます。

[音の長さデータ]

音符	コード
16分音符	01
8分音符	02
付点8分音符	03
4分音符	04
付点4分音符	06
2分音符	08
付点2分音符	0C
全音符	10

休符は高さコードが00で、長さコードは上の音の長さデータ表と同じコードにします。
たとえば4分休符は、0004になります。

［音の高さデータ］

オクターブ	音階	コード	周波数（参考）
6	シ	1D	
6	ラ#	1C	
6	ラ	1B	1760Hz
6	ソ#	1A	
6	ソ	19	
6	ファ#	18	
6	ファ	17	
6	ミ	16	
6	レ#	15	
6	レ	14	
6	ド#	13	
6	ド	12	
5	シ	11	
5	ラ#	10	
5	ラ	0F	880Hz
5	ソ#	0E	
5	ソ	0D	
5	ファ#	0C	
5	ファ	0B	
5	ミ	0A	
5	レ#	09	
5	レ	08	
5	ド#	07	
5	ド	06	
4	シ	05	
4	ラ#	04	
4	ラ	03	440Hz
4	ソ#	02	
4	ソ	01	
	休符	00	

ⅲ）曲の速さ

プログラムを簡略にしたため、演奏速度は下の4段階しか選択できません。

4分音符／分	コード
375	01
188	02
94	03
47	04

MUSICプログラムのアドレス8003を書き換えることで、演奏速度を変えることができます。初期値はコード03になっています。

ⅳ）プログラムの説明

データは2バイトで1組です。前の1バイトが音の高さ、後ろの1バイトが音の長さです。高さのデータはSNDTBLを参照することで、対応する音の高さの周波数データに変換されます。

長さのデータは40ms×8001番地の値を16分音符の長さとしたとき、その長さに対する各音符の長さ比になります。たとえば8003番地の値が標準の03のとき、40ms×3＝120msですから、このときの4分音符の長さは120ms×4＝480msです。これは1分間では60／0．48＝125ですから、1分間に4分音符が125回演奏される速さということになります。

SND1：では8100番地からの曲データを順に読みこみ、音の高さデータがFFなら終りの処理を、00なら休符の処理を行います。その他の場合には音の出力処理を行いますが、その前にCレジスタと比較しています。Cレジスタには1つ前の音の高さデータが入っています（初期値は？です。手抜きですが問題はありません）。同じ高さの音が続く場合に切れ目（約5ms）を入れるためです。

音の高さデータはテープルを参照することで、周波数を決定するデータになり、Dレジスタに保持されます。
SNDS1:が音を出力する中心部分です。周波数データDはカウンタCに入れられ、スピーカーにHを出力したあとSNDS2:のループがCレジスタの値だけ繰り返されます。次にスピーカーにLを出力したあと、SNDS3:のループが同じ回数分繰り返されます。
SNDS2:、SNDS3:のループは1回につき10μsecかかります。
たとえばオクターブ5のラの周波数データはプログラムリストのテーブルデータを見ると、38です。10進数に直すと56ですから、スピーカーからはH＝560μsec、L＝560μsecのパルスが出力されることになります。
つまり周期は1120μsecになります。
オクターブ5のラの音の周波数は880Hzですから、その周期は1136μsecです。計算上の周期は本来の周期より少し短いのですが、データは整数値なので20μsecきざみになることと、ループ以外の部分でも若干の実行時間がかかることから、本来の周期よりも短い値になるようにしているためです。
音階データテーブルも2バイトで構成され、周波数データと繰り返し回数データがペアになっています。
周波数データは1回の周期を決定し、それで音の高さが決まります。しかし1回の周期は音の高さによって異なり、音が高ければ短く、低ければ長くなってしまいます。
このままでは音の長さが高さによって異なってきてしまいます。そこで周波数データとペアにした2番目のデータによって、音の高さが異なっていても、一定の期間、音が出力されるようにしてあります。
オクターブ5のラでは、そのデータは23になっています。10進数に直すと35です。このデータは、プログラムではアドレス8004(REPEAT)に保持され、Eレジスタに入れられてダウンカウントされます。
周期×Eレジスタを計算すると、(1120＋10)×35＝39550μsecです。
これをオクターブ6のラと比較してみます。
周波数データは1Bなので、10進数の27になります。ですから周期は540μsecです(オクターブ6のラの周波数は1760Hzですから、本来の周期は568μsecです)。
2番目のデータは47ですから、10進数では71になります。
周期×Eレジスタは、(540＋10)×71＝39050μsecになります。
高さによって多少の差異はありますが概ね40msecで一定になるようにしてあります。
音符の長さデータは、この40msecの乗数です。Hレジスタに保持され、その値はBレジスタに入れられてダウンカウントされます。
たとえば4分音符の長さデータは04ですから、40×4＝160msecの長さになります。
これは1分間では60／0．16＝375回演奏される長さです。
実際の音の長さは、さらに速さのパラメータ(アドレス8001の値)が乗じられることで決まります。速さのパラメータはLレジスタに入れられてダウンカウントされます。
上で計算した音符の長さ時間は、速さのパラメータが01のときのものです。
速さのパラメータが、02、03、04のときの音符の長さ時間は、上で計算した長さの2倍、3倍、4倍になります。

```
2016/4/29  11:25  music8.txt
END=80D5
              ;;; music 09/10/9 10/10 10/12 10/15
              ;;;
              ;16/4/20 for nd8080
              ;4/29
              ;
                 ORG $8000
              ;
              ;  SPEED=$EFFF
              ;  REPEAT=$EFFE
              ;  MAE=$EFFD
                 DATA=$8100
                 END=$0051;MONITOR START
              ;
8000 C30680       JMP START
8003 03        SPEED:DB 03
8004 00        REPEAT:NOP
8005 00        MAE:NOP
8006 210081    START:LXI H,DATA
8009 3A0580    SND1:LDA MAE
800C 4F           MOV C,A
800D 7E           MOV A,M
800E FEFF         CPI FF
```

```
8010 CA5100      JZ END
8013 B7          ORA A
8014 CA6380      JZ YASUMI
8017 FE1E        CPI 1E
8019 D27C80      JNC ERROR
801C F5          PUSH PSW
801D B9          CMP C
801E CC8180      CZ T5MS
8021 23          INX H
8022 46          MOV B,M;NAGASA
8023 23          INX H
8024 F1          POP PSW
8025 E5          PUSH H
8026 219A80      LXI H,SNDTBL
8029 87          ADD A
802A 5F          MOV E,A
802B 1600        MVI D,00
802D 19          DAD D
802E 56          MOV D,M;TAKASA
802F 23          INX H
8030 5E          MOV E,M;KURIKAESI KAISU
8031 3A0380      LDA SPEED
8034 6F          MOV L,A
8035 60          MOV H,B
8036 7B          MOV A,E
8037 320480      STA REPEAT
                 ;
803A 4A        SNDS1:MOV C,D
803B 3EFF        MVI A,FF;sp out H
803D D398        OUT 98
803F 7F        SNDS2:MOV A,A;5
8040 0D          DCR C;5
8041 C23F80      JNZ SNDS2;10
8044 4A          MOV C,D
8045 3EDF        MVI A,DF;sp out L
8047 D398        OUT 98
8049 7F        SNDS3:MOV A,A;5
804A 0D          DCR C;5
804B C24980      JNZ SNDS3;10
804E 1D          DCR E
804F C23A80      JNZ SNDS1
8052 3A0480      LDA REPEAT
8055 5F          MOV E,A
8056 05          DCR B
8057 C23A80      JNZ SNDS1
805A 44          MOV B,H
805B 2D          DCR L
805C C23A80      JNZ SNDS1
805F E1          POP H
8060 C30980      JMP SND1
                 ;;;
8063 23        YASUMI:INX H
8064 46          MOV B,M;NAGASA
8065 23          INX H
8066 E5          PUSH H
8067 3A0380      LDA SPEED
806A 6F          MOV L,A
806B 60          MOV H,B
```

```
806C CD8680   YASUMI2:CALL T4OMS
806F 05          DCR B
8070 C26C80      JNZ YASUMI2
8073 44          MOV B,H
8074 2D          DCR L
8075 C26C80      JNZ YASUMI2
8078 E1          POP H
8079 C30980      JMP SND1
              ;;;
807C 23       ERROR:INX H
807D 23          INX H
807E C30980      JMP SND1
              ;
8081 0E05     T5MS:MVI C,05
8083 C38880      JMP T4OMS2
              ;
8086 0E28     T4OMS:MVI C,28;=40
8088 CD9080   T4OMS2:CALL T1MS
808B 0D          DCR C
808C C28880      JNZ T4OMS2
808F C9          RET
              ;
8090 F5       T1MS:PUSH PSW
8091 3E60        MVI A,60;=96
8093 7F       T1MS2:MOV A,A;5
8094 3D          DCR A;5
8095 C29380      JNZ T1MS2;10
8098 F1          POP PSW
8099 C9          RET
              ;
              ; SOUND TABLE
809A 00       SNDTBL:DB 00;DUMMY
809B 00          DB 00;DUMMY
809C 7F          DB 7F;so4
809D 10          DB 10
809E 77          DB 77;so#4
809F 11          DB 11
80A0 71          DB 71;ra4
80A1 12          DB 12
80A2 6A          DB 6A;ra#4
80A3 13          DB 13
80A4 64          DB 64;si4
80A5 14          DB 14
80A6 5F          DB 5F;do5
80A7 15          DB 15
80A8 59          DB 59;do#5
80A9 16          DB 16
80AA 54          DB 54;re5
80AB 18          DB 18
80AC 4F          DB 4F;re#5
80AD 19          DB 19
80AE 4B          DB 4B;mi5
80AF 1A          DB 1A
80B0 47          DB 47;fa5
80B1 1C          DB 1C
80B2 43          DB 43;fa#5
80B3 1E          DB 1E
80B4 3F          DB 3F;so5
```

```
80B5 1F          DB 1F
80B6 3B          DB 3B;so#5
80B7 21          DB 21
80B8 38          DB 38;ra5
80B9 23          DB 23
80BA 35          DB 35;ra#5
80BB 25          DB 25
80BC 32          DB 32;si5
80BD 27          DB 27
80BE 2F          DB 2F;do6
80BF 2A          DB 2A
80C0 2C          DB 2C;do#6
80C1 2C          DB 2C
80C2 2A          DB 2A;re6
80C3 2F          DB 2F
80C4 27          DB 27;re#6
80C5 32          DB 32
80C6 25          DB 25;mi6
80C7 35          DB 35
80C8 23          DB 23;fa6
80C9 38          DB 38
80CA 21          DB 21;fa#6
80CB 3B          DB 3B
80CC 1F          DB 1F;so6
80CD 3F          DB 3F;
80CE 1D          DB 1D;so#6
80CF 43          DB 43;
80D0 1B          DB 1B;ra6
80D1 47          DB 47;
80D2 1A          DB 1A;ra#6
80D3 4A          DB 4A;
80D4 18          DB 18;si6
80D5 50          DB 50;
                ;END
DATA        =8100  END         =0051  ERROR       =807C
MAE         =8005  REPEAT      =8004  SND1        =8009
SNDS1       =803A  SNDS2       =803F  SNDS3       =8049
SNDTBL      =809A  SPEED       =8003  START       =8006
T1MS        =8090  T1MS2       =8093  T40MS       =8086
T40MS2      =8088  T5MS        =8081  YASUMI      =8063
YASUMI2     =806C
```

２．　電子オルゴールデータ

①プログラムの8003番地を速度のパラメータの値に変更してください。
②プログラムは8000番地から実行します。

禁じられた遊び（ＫＩＮＪＩＲＡ．ＢＴＫ）
速度のパラメータ(8003番地の値)＝03
8100　0F 04 0F 04 0F 04 0F 04 0D 04 0B 04 0B 04 0A 04
8110　08 04 08 04 0B 04 0F 04 14 04 14 04 14 04 14 04
8120　12 04 10 04 10 04 0F 04 0D 04 0D 04 0F 04 10 04
8130　0F 04 10 04 0F 04 13 04 10 04 0F 04 0F 04 0D 04
8140　0B 04 0B 04 0A 04 08 04 0A 04 0A 04 0A 04 0A 04
8150　0B 04 0A 04 08 04 08 04 08 04 08 08 00 04 0C 04
8160　0C 04 0C 04 0C 04 0A 04 08 04 08 04 07 04 07 04
8170　07 04 06 04 07 04 11 04 11 04 11 04 11 04 13 04
8180　11 04 11 04 0F 04 0F 04 0F 04 11 04 13 04 14 04
8190　14 04 14 04 14 04 13 04 12 04 11 04 11 04 11 04
81A0　11 04 0F 04 0D 04 0C 04 0C 04 0C 04 0C 04 0D 04
81B0　0A 04 08 10 FF

故郷の廃家（ＫＯＫＹＯ．ＢＴＫ）
速度のパラメータ(8003番地の値)＝03
8100　0D 06 0F 02 0D 04 0A 04 12 06 14 02 12 04 0F 04
8110　0D 04 0A 04 12 04 0A 04 08 0C 00 04 0D 06 0F 02
8120　0D 04 0A 04 12 06 14 02 12 04 0F 04 12 04 0A 04
8130　0A 06 08 02 06 0C 00 04 08 06 07 02 08 04 0B 04
8140　0A 06 09 02 0A 04 0D 04 0F 04 0D 04 0B 04 0A 04
8150　08 0C 00 04 0D 06 0F 02 0D 04 0A 04 12 06 14 02
8160　12 04 0F 04 0D 04 0A 04 0A 02 08 06 06 0C 00 04
8170　0D 06 09 02 0A 04 0D 0C 00 08 0D 04 08 04 11 06
8180　0F 02 0D 08 00 08 0D 06 09 02 0A 04 0D 0C 00 08
8190　12 04 0A 04 0A 02 08 06 06 0C FF 02 0F 02 0D 02

ねこふんじゃった（ＮＥＫＯ．ＢＴＫ）
速度のパラメータ(8003番地の値)＝03
8100　0F 02 0D 02 06 02 00 02 12 02 00 02 12 02 00 02
8110　0F 02 0D 02 06 02 00 02 12 02 00 02 12 02 00 02
8120　0F 02 0D 02 06 02 00 02 12 02 00 02 03 02 00 02
8130　12 02 00 02 01 04 11 02 00 02 11 02 00 02 0F 02
8140　0D 02 01 02 00 02 11 02 00 02 11 02 00 02 0F 02
8150　0D 02 01 02 00 02 11 02 00 02 11 02 00 02 0F 02
8160　0D 02 01 02 00 02 11 02 00 02 03 02 00 02 11 02
8170　00 02 06 02 00 02 12 02 00 02 12 04 0F 02 0D 02
8180　16 02 00 02 12 02 00 02 12 02 00 02 0F 02 0D 02
8190　16 02 00 02 12 02 00 02 12 02 00 02 0F 02 0D 02
81A0　16 02 00 02 12 02 00 02 19 02 00 02 12 02 00 02
81B0　1B 04 11 02 00 02 11 02 00 02 0F 02 0D 02 1B 02
81C0　00 02 11 02 00 02 11 02 00 02 0F 02 0D 02 1B 02
81D0　00 02 11 02 00 02 11 02 00 02 0F 02 0D 02 1B 02
81E0　00 02 11 02 00 02 19 02 00 02 11 02 00 02 16 04
81F0　12 02 00 02 12 02 00 02 0F 02 0D 02 06 02 00 02
8200　12 02 00 02 01 02 00 02 12 02 00 02 06 02 00 02
8210　12 02 00 02 01 02 00 02 12 02 00 02 06 04 05 04
8220　06 04 07 04 08 04 11 02 00 02 11 02 00 02 0F 02
8230　0D 02 08 02 00 02 11 02 00 02 01 02 00 02 11 02
8240　00 02 08 02 00 02 11 02 00 02 01 02 00 02 11 02
8250　00 02 08 04 07 04 08 04 09 04 0A 04 16 02 00 02

```
8260  16 02 00 02 0F 02 0D 02 06 02 00 02 12 02 00 02
8270  12 02 00 02 0F 02 0D 02 06 02 00 02 12 02 00 02
8280  12 02 00 02 0F 02 0D 02 06 02 00 02 12 02 00 02
8290  03 02 00 02 12 02 00 02 01 04 11 02 00 02 11 02
82A0  00 02 0F 02 0D 02 01 02 00 02 11 02 00 02 11 02
82B0  00 02 0F 02 0D 02 01 02 00 02 11 02 00 02 11 02
82C0  00 02 0F 02 0D 02 01 02 00 02 11 02 00 02 03 02
82D0  00 02 11 02 00 02 12 02 00 02 12 02 00 02 12 04
82E0  0D 02 0C 02 0D 02 0E 02 0F 02 10 02 11 02 12 04
82F0  FF
```

聖者の行進（ＳＥＩＪＡ．ＢＴＫ）

速度のパラメータ(8003番地の値)＝02

```
8100  0B 04 0F 04 10 04 12 10 00 06 0B 02 0F 04 10 04
8110  12 10 00 06 0B 02 0F 04 10 04 12 08 0F 08 0B 08
8120  0F 08 0D 10 00 08 0F 04 0D 04 0B 0C 0B 04 0F 08
8130  12 04 12 04 12 04 10 0C 00 06 0B 02 0F 04 10 04
8140  12 08 0F 08 0B 08 0D 08 0B 10 FF
```

聖夜（ＳＥＩＹＡ．ＢＴＫ）

速度のパラメータ(8003番地の値)＝04

```
8100  0D 06 0F 02 0D 04 0A 0C 0D 06 0F 02 0D 04 0A 0C
8110  14 08 14 04 11 0C 12 08 12 04 0D 0C 0F 08 0F 04
8120  12 06 11 02 0F 04 0D 06 0F 02 0D 04 0A 0C 0F 08
8130  0F 04 12 06 11 02 0F 04 0D 06 0F 02 0D 04 0A 0C
8140  14 08 14 04 17 06 14 02 11 04 12 0C 16 0C 12 06
8150  0D 02 0A 04 0D 06 0B 02 08 04 06 0C FF
```

よろこびのうた（ＹＯＲＯＫＯＢＩ．ＢＴＫ）

速度のパラメータ(8003番地の値)＝03

```
8100  14 04 14 04 15 04 17 04 17 04 15 04 14 04 12 04
8110  10 04 10 04 12 04 14 04 14 06 12 02 12 08 14 04
8120  14 04 15 04 17 04 17 04 15 04 14 04 12 04 10 04
8130  10 04 12 04 14 04 12 06 10 02 10 08 12 04 12 04
8140  14 04 10 04 12 04 14 02 15 02 14 04 10 04 12 04
8150  14 02 15 02 14 04 12 04 10 04 12 04 0B 04 14 08
8160  14 04 15 04 17 04 17 04 15 04 14 04 12 04 10 04
8170  10 04 12 04 14 04 12 06 10 02 10 08 FF
```

おおスザンナ（ＳＵＺＡＮＮＡ．ＢＴＫ）

速度のパラメータ(8003番地の値)＝01

```
8100  06 04 08 04 0A 08 0D 08 0D 08 0F 08 0D 08 0A 08
8110  06 0C 08 04 0A 08 0A 08 08 08 06 08 08 18 06 04
8120  08 04 0A 08 0D 08 0D 0C 0F 04 0D 08 0A 08 06 0C
8130  08 04 0A 08 0A 08 08 08 08 08 06 18 06 04 08 04
8140  0A 08 0D 08 0D 08 0F 08 0D 08 0A 08 06 0C 08 04
8150  0A 08 0A 08 08 08 06 08 08 18 06 04 08 04 0A 08
8160  0D 08 0D 08 0F 08 0D 08 0A 08 06 0C 08 04 0A 08
8170  0A 08 08 08 08 08 06 18 00 08 0B 10 0B 10 0F 08
8180  0F 10 0F 08 0D 08 0D 08 0A 08 06 08 08 18 06 04
8190  08 04 0A 08 0D 08 0D 08 0F 08 0D 08 0A 08 06 0C
81A0  08 04 0A 08 0A 08 08 08 08 08 06 10 FF
```

エリーゼのために（ＥＬＩＳＥ．ＢＴＫ）

速度のパラメータ(8003番地の値)＝02

```
8100  16 04 15 04 16 04 15 04 16 04 11 04 14 04 12 04
8110  0F 08 00 04 06 04 0A 04 0F 04 11 08 00 04 0A 04
8120  0E 04 11 04 12 08 00 04 0A 04 16 04 15 04 16 04
```

```
8130  15 04 16 04 11 04 14 04 12 04 0F 08 00 04 06 04
8140  0A 04 0F 04 11 08 00 04 0A 04 12 04 11 04 0F 08
8150  00 04 11 04 12 04 14 04 16 0C 0D 04 17 04 16 04
8160  14 0C 0B 04 16 04 14 04 12 0C 0A 04 14 04 12 04
8170  11 04 00 04 16 04 15 04 16 04 15 04 16 04 11 04
8180  14 04 12 04 0F 08 00 04 06 04 0A 04 0F 04 11 08
8190  00 04 0A 04 0E 04 11 04 12 08 00 04 0A 04 16 04
81A0  15 04 16 04 15 04 16 04 11 04 14 04 12 04 0F 08
81B0  00 04 06 04 0A 04 0F 04 11 08 00 04 0A 04 12 04
81C0  11 04 7A 00 0F 08 FF
```

[memo]