ND80ZⅢ　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.alles.or.jp/~thisida/

2010. 9. 25　Rev.　1. 0

第1部　ゲームプログラム

ｉ）はじめに

本編はND80ZⅢ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、今回ND80ZⅢのために書き直したものです。

ND80ZⅢのオリジナルは30年前にNECのTK80とソフトコンパチブルとして発売したND80にまでさかのぼります。

ND80を発売した当初はTK80と同じくCPUは8080でしたが、その後CPUに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、当時に近い外観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月日が過ぎてしまいました。

しかし2003年ころ、参考プログラム集はありませんか？という問い合わせがあったのをきっかけにして、思い切って、ゴールデンウィークを棒に振って、作成したのが本書の前身となった「ND80K応用プログラム集」です。

本書中のプログラムのほとんどは、その昔に出版されていた、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、ND80ZⅢ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、ND80ZⅢではZ80（ザイログ）ニーモニックです。またZ80は8080に倍加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てみよう、という気持ちもあって、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が、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）。

この応用プログラム集は、「プロのプログラマとして」などという気負いが入ってしまったため、かなりアクの強いテクニックを随所で用いていますから、良い教科書となりうるものではないかもしれません。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。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。しかしこの応用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。

ND80ZⅢ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　初出
2010年9月　加筆訂正　(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DOS／Vパソコンを使い、当社オリジナルのZ80アセンブラで作成しました。

ND80ZⅢ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。

ND80ZⅢ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、ND80ZⅢ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E800～に割り当てています。
プログラムリストはファイル名末尾にバージョンを示すアルファベットがついた形でアセンブルしたものになっています。本書に記載したものが最終バージョンで、CDROMにも同じ最終バージョンが末尾のアルファベットを抜いたファイル名で収録してあります。

ⅲ）附属CDROMのプログラム、データの扱い方

ND80ZⅢ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（nd80z3フォルダ）にはプログラムがアセンブラソースファイル、バイナリファイル、TK80形式のバイナリファイルとして入っています。
nd80z3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ND80ZⅢ操作説明書、USB接続説明書を参照してください。
応用プログラムはZ80アセンブラで作成しました。

1）ND80Zモニタでの操作を前提にしていますから、ND80ZⅢはND80Zモニタモードで使ってください。
2）USBケーブルでND80ZⅢとWindowsパソコンを接続します。
3）ND80ZⅢのキーボードから[＊(I／O)][D LD]と入力してください。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HIDWRコマンドを使って、ND80ZⅢで実行したいプログラムを送信します。
＞HIDWR MOGURA. 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（OHAYO．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、ND80ZⅢに付属のZ80命令説明書を参照してください。またND80ZⅢ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ND80ZⅢ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（＄FFF8～＄FFFF）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＄FFF8～＄FFFFに00を書き込むことでLEDを全消灯しています。

ここで使用しているDJNZは8080には無い命令で、

DEC　B

JP　NZ, xxxx

と同じ機能です。1命令で済むのでZ80のプログラムではよく使われます。ただしジャンプ先は＋127～－128の範囲のアドレスに限られます。

;ASIATO　dispと;ohayo　dispで使っているLDIも8080には無い命令です。

LD　A,（HL）

LD　（DE）, A

INC　HL

INC　DE

DEC　BC

LD　A, B

OR　C

と同等の機能をAレジスタを使うことなく行います。ここでは1バイト転送ごとにタイマールーチンをCALLしてウエイトをかけているために、LDIを使っていますが、普通は指定バイト分を一度に転送できるLDIRやLDDRを使います。

```
2010/9/17  14:32  ohayob.txt
END=807C
                ; OHAYO for NDZ
                ; 03/04/28 5/9
                ;10/6/1 for ND80Z3
                ;
                     ORG $8000
                     LED1=$FFF8
                     LED4=$FFFB
                     START1=$080F
                     TM5M=$02DD
                ;
8000 CD4E80          CALL CLR
                ;ASIATO disp
```

```
8003 216D80        LD HL,ASIDT
8006 11F8FF        LD DE,LED1
8009 010800        LD BC,$0008
800C CD5980   ASIDP2:CALL TM1S
800F EDA0          LDI
8011 EA0C80        JP PE,ASIDP2
8014 CD5980        CALL TM1S
8017 CD4E80        CALL CLR
801A CD5980        CALL TM1S
              ;EYE close/open
801D 0603          LD B,03
801F 211C1C        LD HL,$1C1C
8022 113F3F        LD DE,$3F3F
8025 22FBFF   EYE:LD (LED4),HL
8028 CD5F80        CALL TM025
802B ED53FBFF      LD (LED4),DE
802F CD5C80        CALL TM05
8032 10F1          DJNZ *EYE
8034 CD4E80        CALL CLR
              ;ohayo disp
8037 217580        LD HL,OHAYODT
803A 11F8FF        LD DE,LED1
803D 010800        LD BC,$0008
8040 CD5C80   OHYDP2:CALL TM05
8043 EDA0          LDI
8045 EA4080        JP PE,OHYDP2
8048 CD5980      CALL TM1S
804B C30F08        JP START1
              ;LED clear
804E 21F8FF   CLR:LD HL,LED1
8051 010008        LD BC,$0800
8054 71       CLR2:LD (HL),C
8055 23            INC HL
8056 10FC          DJNZ *CLR2
8058 C9            RET
              ;1sec timer/0.5sec timer/0.25sec timer
8059 CD5C80   TM1S:CALL TM05
805C CD5F80   TM05:CALL TM025
805F D5       TM025:PUSH DE
8060 1E32          LD E,32;=50
8062 D5       TM025_2:PUSH DE
8063 CDDD02      CALL TM5M;4.496ms
8066 D1          POP DE
8067 1D            DEC E
8068 C26280        JP NZ,TM025_2
806B D1            POP DE
806C C9            RET
              ;asiato data
806D 43       ASIDT:DB 43
806E 4C            DB 4C
806F 43            DB 43
8070 4C            DB 4C
8071 43            DB 43
8072 4C            DB 4C
8073 43            DB 43
8074 4C            DB 4C
              ;"ohayo---"
8075 3F       OHAYODT:DB 3F;O
```

```
8076 76            DB 76;H
8077 77            DB 77;A
8078 6E            DB 6E;y
8079 3F            DB 3F;O
807A 40            DB 40;-
807B 40            DB,40;-
807C 40            DB 40;-
                 ;
ASIDP2       =800C  ASIDT        =806D  CLR          =804E
CLR2         =8054  EYE          =8025  LED1         =FFF8
LED4         =FFFB  OHAYODT      =8075  OHYDP2       =8040
START1       =080F  TM025        =805F  TM025_2      =8062
TM05         =805C  TM1S         =8059  TM5M         =02DD
```

プログラム№2　CRAZY　EIGHT（CRAZY8．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E800）、LEDの残り消灯数を格納するREST、乱数の作業用にRNDDT、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FFF7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。FFF7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CP　09はよいのですが、OR　A、JP　Zとしないのはなぜでしょう？

そうする代わりにDEC　A、JP　M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DECするなら、ここでDEC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、

```
LD　HL, LED1
LD　E, A
LD　D, 00
ADD　HL, DE
```

とするべきです。ここでは、ADD　A, 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＝FFF8という条件です。この場合にはADD　A, Lでも、OR　A, 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RNDをCALLすると、00～FFの範囲のランダムな数が得られます。その8ビットの数のうち下位3ビットだけを、AND　07で取り出すと0～7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DECを1回だけ実行します。その結果(REST)＝0になったらゲーム終了です。

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200F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RNDではなるべくデータが片寄らないように4バイト分のデータをもとにして計算し、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、Rレジスタの値を利用しています。Rレジスタは8080にはない特殊なレジスタで、Z80にダイナミックメモリを接続したとき、そのリフレッシュをZ80自身が行えるように工夫された機能のためのレジスタです(なんだかわからない人にはさっぱりわからないでしょうが、まあ理解する必要はありません)。ともかくプログラムには必要のないレジスタで、00～7Fの範囲の値をもち、命令実行毎に＋1されます。

RNDの実行結果はまあまあですが、ときとして同じ数がつながったり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。

```
2010/9/17  14:28  crazy8b.txt
END=8122
              ; CRAZY EIGHT for NDZ
              ; 03/05/07 5/9
              ;10/6/1 for ND80Z3
              ;
                   ORG $8000
                   YRADRS=$E800
                   REST=$E801
                   RNDDT=$E810;$E811
                   CNTR=$FFF7;=DP4
                   LED1=$FFF8
              ;
                   DTDPS=$05C0;DATA DISP
                   KEYIN1=$0616
                   D1_1=$02DF;about 125microsec
              ;
              ;data set
8000 CDB880   START:CALL CLR
8003 AF            XOR A
8004 32F7FF        LD (CNTR),A;DECIMAL COUNTER clear
8007 3E08          LD A,08
8009 3201E8        LD (REST),A
              ;GAME START
800C 3E10          LD A,10;puuuu
800E 060A          LD B,0A
8010 CDC380   START1:CALL SOUND
8013 10FB          DJNZ START1
8015 CD1606   KEY:CALL KEYIN1
8018 FE09          CP 09
801A D21580        JP NC,KEY
801D 3D            DEC A
```

```
801E FA1580       JP M,KEY
             ;keyin position check
8021 3200E8       LD (YRADRS),A;position save
8024 21F8FF       LD HL,LED1
8027 85           ADD A,L
8028 6F           LD L,A
8029 7E           LD A,(HL)
802A B7           OR A
802B C21580       JP NZ,KEY;already used
802E 365C         LD (HL),5C;"low" disp
             ;my turn
8030 CD9A80  MYTURN:CALL RND;1-8 select
8033 E607         AND 07
8035 57           LD D,A
8036 3A00E8       LD A,(YRADRS)
8039 BA           CP D
803A CA5580       JP Z,HIT
803D 7A           LD A,D
803E 21F8FF       LD HL,LED1
8041 85           ADD A,L
8042 6F           LD L,A
8043 7E           LD A,(HL)
8044 B7           OR A
8045 C23080       JP NZ,MYTURN;already used
8048 3663         LD (HL),63;"high" disp
804A 2101E8       LD HL,REST
804D 35           DEC (HL)
804E 35           DEC (HL)
804F C21580       JP NZ,KEY
8052 C38180       JP END
             ;
8055 21F8FF  HIT:LD HL,LED1
8058 85           ADD A,L
8059 6F           LD L,A
805A 0E05         LD C,05
805C 367F    HIT1:LD (HL),7F;"8" disp
805E 3E17         LD A,17;piii
8060 0603         LD B,03
8062 CDC380  HIT2:CALL SOUND
8065 10FB         DJNZ HIT2
8067 3600         LD (HL),00
8069 CD0C81       CALL TM02S
806C 0D           DEC C
806D C25C80       JP NZ,HIT1
8070 367F         LD (HL),7F
8072 2101E8       LD HL,REST
8075 3AF7FF       LD A,(CNTR);point up
8078 86           ADD A,(HL)
8079 27           DAA
807A 32F7FF       LD (CNTR),A
807D 35           DEC (HL)
807E C21580       JP NZ,KEY
8081 0614    END:LD B,14;=20 ,2sec wait
8083 CD0F81  WAIT:CALL TM01S
8086 10FB         DJNZ WAIT
8088 CDC005       CALL DTDPS
808B 21F8FF       LD HL,LED1
808E 0606         LD B,06
```

```
8090 AF            XOR A
8091 CDBE80    END2:CALL CLR1
8094 CD1606         CALL KEYIN1
8097 C30080         JP START
               ;
               ;ransu
809A E5        RND:PUSH HL
809B D5             PUSH DE
809C 2A10E8         LD HL,(RNDDT)
809F ED5F           LD A,R
80A1 5F             LD E,A
80A2 19             ADD HL,DE
80A3 7C             LD A,H
80A4 E603           AND 03
80A6 67             LD H,A
80A7 7E             LD A,(HL)
80A8 23             INC HL
80A9 86             ADD A,(HL)
80AA 5F             LD E,A
80AB 23             INC HL
80AC 86             ADD A,(HL)
80AD 23             INC HL
80AE 86             ADD A,(HL)
80AF 57             LD D,A
80B0 ED5310E8       LD (RNDDT),DE
80B4 7A             LD A,D
80B5 D1             POP DE
80B6 E1             POP HL
80B7 C9             RET
               ;
               ;LED all clear
80B8 21F8FF    CLR:LD HL,LED1
80BB AF             XOR A
80BC 0608           LD B,08
80BE 77        CLR1:LD (HL),A
80BF 23             INC HL
80C0 10FC           DJNZ *CLR1
80C2 C9             RET
               ;
               ;
80C3 F5        SOUND:PUSH AF
80C4 E5          PUSH HL
80C5 D5          PUSH DE
80C6 C5          PUSH BC
80C7 21F480      LD HL,SNDTBL
80CA 85          ADD A,L
80CB 6F          LD L,A
80CC 46          LD B,(HL)
80CD 1E1A        LD E,1A
80CF 50        SNDS1:LD D,B
80D0 3EFF        LD A,FF;SP OUT=H
80D2 D398        OUT (98),A
80D4 E5        SNDS2:PUSH HL;11clk---------
80D5 E5          PUSH HL;11clk            |11+11+10+10+4+4+10=60
80D6 E1               POP HL;10clk        |60/6=10
80D7 E1               POP HL;10clk        |
80D8 00               NOP;4clk            |10microsec
80D9 15          DEC D;4clk               |
```

```
80DA C2D480      JP NZ,SNDS2;10clk---
80DD 50          LD D,B
80DE 3EDF        LD A,DF;SP OUT=L
80E0 D398        OUT (98),A
80E2 E5        SNDS3:PUSH HL;11clk---------
80E3 E5          PUSH HL;11clk            |11+11+10+10+4+4+10=60
80E4 E1               POP HL;10clk        |60/6=10
80E5 E1               POP HL;10clk        |
80E6 00               NOP;4clk            |10microsec
80E7 15          DEC D;4clk               |
80E8 C2E280      JP NZ,SNDS3;10clk---
80EB 1D          DEC E
80EC C2CF80      JP NZ,SNDS1
80EF C1          POP BC
80F0 D1          POP DE
80F1 E1          POP HL
80F2 F1          POP AF
80F3 C9          RET
               ; SOUND TABLE
80F4 7F        SNDTBL:DB 7F;so4
80F5 77          DB 77;so#4
80F6 71          DB 71;ra4
80F7 6A          DB 6A;ra#4
80F8 5F          DB 5F;do5
80F9 59          DB 59;do#5
80FA 54          DB 54;re5
80FB 4F          DB 4F;re#
80FC 47          DB 47;fA5
80FD 43          DB 43;fa#5
80FE 3F          DB 3F;so5
80FF 3B          DB 3B;so5#
8100 35          DB 35;ra#5
8101 32          DB 32;si5
8102 2F          DB 2F;do6
8103 2C          DB 2C;do#6
8104 25          DB 25;mi6
8105 27          DB 27;re#6
8106 2A          DB 2A;re6
8107 4B          DB 4B;mi5
8108 38          DB 38;ra5
8109 64          DB 64;si4
810A 23          DB 23;fa6
810B 21          DB 21;fa#6
               ;
               ;0.2sec timer
810C CD0F81    TM02S:CALL TM01S
               ;0.1sec timer
810F D5        TM01S:PUSH DE
8110 1E64           LD E,64;=100
8112 CD1B81    TM01S2:CALL TM1M
8115 1D             DEC E
8116 C21281         JP NZ,TM01S2
8119 D1             POP DE
811A C9             RET
811B D5        TM1M:PUSH DE
811C 1608        LD D,08
811E CDDF02    TM1M_2:CALL D1_1
8121 D1          POP DE
```

```
8122 C9          RET
                 ;N
CLR         =80B8  CLR1        =80BE  CNTR        =FFF7
D1_1        =02DF  DTDPS       =05C0  END         =8081
END2        =8091  HIT         =8055  HIT1        =805C
HIT2        =8062  KEY         =8015  KEYIN1      =0616
LED1        =FFF8  MYTURN      =8030  REST        =E801
RND         =809A  RNDDT       =E810  SNDS1       =80CF
SNDS2       =80D4  SNDS3       =80E2  SNDTBL      =80F4
SOUND       =80C3  START       =8000  START1      =8010
TM01S       =810F  TM01S2      =8112  TM02S       =810C
TM1M        =811B  TM1M_2      =811E  WAIT        =8083
YRADRS      =E800
```

プログラム№3　Hi－Lo（HILO．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RND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FFEFに割り当てています。先回は表示用データレジスタ（FFF4～FFF7）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（FFEC～FFEF）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

LD　（DTRG），HL

HLの数値をLED表示用データレジスタに入れる場合

EX　DE，HL

LD　HL，DP3；DP3＝＄FFF6

LD　（HL），D

INC　HL

LD　（HL），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ND80ZⅢ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。

```
2010/9/17  14:30  hilob.txt
END=80CA
              ; Hi-Lo for NDZ
              ; 03/04/29 4/30 5/1 5/3 5/5 5/9 5/10
              ;10/6/1 for ND80Z3
              ;
                   ORG $8000
                   HIDNDT=$E800;$E801
                   RNDDT=$E810;$E811
                   DTRG=$FFEC
                   ADRG=$FFEE
                   CNTR=$FFEF;=ADRGH
                   LED1=$FFF8
                   LED3=$FFFA
              ;
                   ADDPS=$05A1
                   KEYIN1=$0616
```

```
                D1_1=$02DF;about 125microsec
              ;
              ;data(hidden) set
8000 0604     START:LD B,04
8002 CD7180   DTST1:CALL RND
8005 E60F          AND 0F
8007 FE0A          CP 0A
8009 D20280        JP NC,DTST1
800C 29            ADD HL,HL
800D 29            ADD HL,HL
800E 29            ADD HL,HL
800F 29            ADD HL,HL
8010 B5            OR L
8011 6F            LD L,A
8012 10EE          DJNZ *DTST1
8014 2200E8        LD (HIDNDT),HL
8017 210000        LD HL,$0000
801A 22EEFF        LD (ADRG),HL;DECIMAL COUNTER clear
801D 22ECFF   REENT:LD (DTRG),HL
8020 E5            PUSH HL
8021 CDA105        CALL ADDPS
8024 21BF80        LD HL,STDT
8027 CDB680   REENT1:CALL H4DP
802A E1            POP HL
              ;key entry(0000-9999)
802B CD1606   KEY:CALL KEYIN1
802E FE15          CP 15;WRITE INC
8030 CA4180        JP Z,COMP
8033 FE0A          CP 0A
8035 D22B80        JP NC,KEY
8038 29            ADD HL,HL
8039 29            ADD HL,HL
803A 29            ADD HL,HL
803B 29            ADD HL,HL
803C B5            OR L
803D 6F            LD L,A
803E C31D80        JP REENT
              ;compare DATA
8041 3AEFFF   COMP:LD A,(CNTR);DECIMAL COUNTER up
8044 C601          ADD A,01
8046 27            DAA
8047 32EFFF        LD (CNTR),A
804A B7            OR A;clear CARRY
804B ED5B00E8      LD DE,(HIDNDT)
804F E5            PUSH HL
8050 ED52          SBC HL,DE
8052 CA6180        JP Z,SAME
8055 21C780        LD HL,LODT
8058 DA2780        JP C,REENT1
805B 21C380        LD HL,HIDT
805E C32780        JP REENT1
8061 E1       SAME:POP HL
8062 CDA105        CALL ADDPS
8065 210000        LD HL,$0000
8068 22FAFF        LD (LED3),HL
              ;OK! ALL LED blink
806B CD8F80        CALL ALBLNK
806E C30080        JP START
```

```
                ;
                ;ransu
8071 E5         RND:PUSH HL
8072 D5             PUSH DE
8073 2A10E8         LD HL,(RNDDT)
8076 ED5F           LD A,R
8078 5F             LD E,A
8079 19             ADD HL,DE
807A 7C             LD A,H
807B E603           AND 03
807D 67             LD H,A
807E 7E             LD A,(HL)
807F 23             INC HL
8080 86             ADD A,(HL)
8081 5F             LD E,A
8082 23             INC HL
8083 86             ADD A,(HL)
8084 23             INC HL
8085 86             ADD A,(HL)
8086 57             LD D,A
8087 ED5310E8       LD (RNDDT),DE
808B 7A             LD A,D
808C D1             POP DE
808D E1             POP HL
808E C9             RET
                ;
                ;LED all blink
808F 060A       ALBLNK:LD B,0A;=10
8091 3EEF       ALBLNK1:LD A,EF
8093 D398          OUT (98),A;LED OFF
8095 CDA280          CALL TM01S
8098 3EFF            LD A,FF
809A D398          OUT (98),A;LED ON
809C CDA280          CALL TM01S
809F 10F0            DJNZ *ALBLNK1
80A1 C9              RET
                ;
                ;0.1sec timer
80A2 D5         TM01S:PUSH DE
80A3 1E64            LD E,64;=100
80A5 CDAE80     TM01S2:CALL TM1M
80A8 1D              DEC E
80A9 C2A580          JP NZ,TM01S2
80AC D1              POP DE
80AD C9              RET
80AE D5         TM1M:PUSH DE
80AF 1608          LD D,08
80B1 CDDF02     TM1M_2:CALL D1_1
80B4 D1            POP DE
80B5 C9            RET
                ;
                ;High 4 disp
80B6 11F8FF     H4DP:LD DE,LED1
80B9 010400          LD BC,$0004
80BC EDB0            LDIR
80BE C9              RET
                ;
80BF 40         STDT:DB 40
```

```
80C0 40            DB 40
80C1 00            DB 00
80C2 00            DB 00
80C3 76       HIDT:DB 76;H
80C4 06            DB 06;i
80C5 00            DB 00
80C6 00            DB 00
80C7 38       LODT:DB 38;L
80C8 3F            DB 3F;O
80C9 00            DB 00
80CA 00            DB 00
                ;
ADDPS       =05A1  ADRG        =FFEE  ALBLNK      =808F
ALBLNK1     =8091  CNTR        =FFEF  COMP        =8041
D1_1        =02DF  DTRG        =FFEC  DTST1       =8002
H4DP        =80B6  HIDNDT      =E800  HIDT        =80C3
KEY         =802B  KEYIN1      =0616  LED1        =FFF8
LED3        =FFFA  LODT        =80C7  REENT       =801D
REENT1      =8027  RND         =8071  RNDDT       =E810
SAME        =8061  START       =8000  STDT        =80BF
TM01S       =80A2  TM01S2      =80A5  TM1M        =80AE
TM1M_2      =80B1
```

プログラムNo.4　FLIPFLOP（FLIPFLOP．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RND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（＄0616）をCAII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（＄0623）をCALLしています。

```
2010/9/17  14:29  flipflpb.txt
END=80EB
              ; FLIP-FLOP for NDZ
              ; 03/04/29 4/30 5/1 5/3 5/5 5/10
              ;10/6/1 for ND80Z3
              ;
                   ORG $8000
                   LINKBF=$E800
                   RNDDT=$E810;$E811
                   LED1=$FFF8
                   DP1=$FFF4
                   DP3=$FFF6
                   CNTR=$FFF7;=DP4
              ;
                   DTDPS=$05C0
                   KEYIN1=$0616
                   KEYIN2=$0623
                   D1_1=$02DF;about 125microsec
              ;
              ;all"UP" disp
8000 3E63     START:LD A,63;"UP"
8002 0608          LD B,08
8004 21F8FF        LD HL,LED1
8007 77       UPDP1:LD (HL),A
8008 23            INC HL
8009 10FC          DJNZ UPDP1
800B 210000        LD HL,$0000
800E 22F6FF        LD (DP3),HL;DECIMAL COUNTER clear
8011 22F4FF        LD (DP1),HL
              ;LINK set
8014 0608          LD B,08
8016 2100E8        LD HL,LINKBF
```

```
8019 CDA780   LNKST1:CALL RND
801C E607          AND 07
801E 77            LD (HL),A
801F 23            INC HL
8020 10F7          DJNZ LNKST1
              ;key entry
8022 CD1606   KEY:CALL KEYIN1
8025 FE09          CP 09
8027 D22280        JP NC,KEY
802A 3D            DEC A
802B FA2280        JP M,KEY
              ;LED blink and wait WINC in
802E 21F8FF   BLNK0:LD HL,LED1
8031 85            ADD A,L
8032 6F            LD L,A
8033 56            LD D,(HL);LED data save
8034 42            LD B,D
8035 0E00          LD C,00
8037 71       BLNK1:LD (HL),C
8038 CDD880        CALL TM01S
803B D5          PUSH DE
803C C5          PUSH BC
803D CD2306        CALL KEYIN2
8040 C1          POP BC
8041 D1          POP DE
8042 FE09          CP 09
8044 D24F80        JP NC,BLNK2
8047 3D            DEC A
8048 FA5480        JP M,BLNK3
804B 72            LD (HL),D
804C C32E80        JP BLNK0
804F FE15     BLNK2:CP 15;WRITE INC
8051 CA5A80        JP Z,REVRS
8054 79       BLNK3:LD A,C
8055 48            LD C,B
8056 47            LD B,A
8057 C33780        JP BLNK1
              ;LED reverse
805A 72       REVRS:LD (HL),D
805B 3E63          LD A,63;"UP"
805D BE            CP (HL)
805E C26380        JP NZ,REVRS1
8061 3E5C          LD A,5C;"DOWN"
8063 77       REVRS1:LD (HL),A
              ;link LED reverse
8064 7D            LD A,L
8065 E607          AND 07
8067 57            LD D,A; No. save
8068 2100E8        LD HL,LINKBF
806B 85            ADD A,L
806C 6F            LD L,A
806D 7E            LD A,(HL)
806E BA            CP D
806F CA8080        JP Z,REVRS3;same No.(no LINK)
8072 21F8FF        LD HL,LED1
8075 85            ADD A,L
8076 6F            LD L,A
8077 3E63          LD A,63;"UP"
```

```
8079 BE            CP (HL)
807A C27F80        JP NZ,REVRS2
807D 3E5C          LD A,5C;"DOWN"
807F 77       REVRS2:LD (HL),A
8080 3AF7FF   REVRS3:LD A,(CNTR);DECIMAL COUNTER up
8083 C601          ADD A,01
8085 27            DAA
8086 32F7FF        LD (CNTR),A
              ;check
8089 21F8FF        LD HL,LED1
808C 0608          LD B,08
808E 3E5C          LD A,5C;"DOWN"
8090 BE       CK1:CP (HL)
8091 C22280        JP NZ,KEY;not success
8094 23            INC HL
8095 10F9          DJNZ *CK1
              ;OK!  ALL LED blink and COUNTER disp
8097 CDC580        CALL ALBLNK
809A CDC005        CALL DTDPS
809D 0614          LD B,14;=20 ,2sec wait
809F CDD880   WAIT:CALL TM01S
80A2 10FB          DJNZ WAIT
80A4 C30080        JP START
              ;
              ;ransu
80A7 E5       RND:PUSH HL
80A8 D5            PUSH DE
80A9 2A10E8        LD HL,(RNDDT)
80AC ED5F          LD A,R
80AE 5F            LD E,A
80AF 19            ADD HL,DE
80B0 7C            LD A,H
80B1 E603          AND 03
80B3 67            LD H,A
80B4 7E            LD A,(HL)
80B5 23            INC HL
80B6 86            ADD A,(HL)
80B7 5F            LD E,A
80B8 23            INC HL
80B9 86            ADD A,(HL)
80BA 23            INC HL
80BB 86            ADD A,(HL)
80BC 57            LD D,A
80BD ED5310E8      LD (RNDDT),DE
80C1 7A            LD A,D
80C2 D1            POP DE
80C3 E1            POP HL
80C4 C9            RET
              ;
              ;LED all blink
80C5 060A     ALBLNK:LD B,0A;=10
80C7 3EEF     ALBLNK1:LD A,EF;LED OFF
80C9 D398        OUT (98),A
80CB CDD880        CALL TM01S
80CE 3EFF          LD A,FF;LED ON
80D0 D398        OUT (98),A
80D2 CDD880        CALL TM01S
80D5 10F0          DJNZ *ALBLNK1
```

```
80D7 C9            RET
                   ;
                   ;0.1sec timer
80D8 D5        TM01S:PUSH DE
80D9 1E64          LD E,64;=100
80DB CDE480    TM01S2:CALL TM1M
80DE 1D            DEC E
80DF C2DB80        JP NZ,TM01S2
80E2 D1            POP DE
80E3 C9            RET
80E4 D5        TM1M:PUSH DE
80E5 1608        LD D,08
80E7 CDDF02    TM1M_2:CALL D1_1
80EA D1          POP DE
80EB C9          RET
                   ;;
ALBLNK       =80C5  ALBLNK1      =80C7  BLNK0        =802E
BLNK1        =8037  BLNK2        =804F  BLNK3        =8054
CK1          =8090  CNTR         =FFF7  D1_1         =02DF
DP1          =FFF4  DP3          =FFF6  DTDPS        =05C0
KEY          =8022  KEYIN1       =0616  KEYIN2       =0623
LED1         =FFF8  LINKBF       =E800  LNKST1       =8019
REVRS        =805A  REVRS1       =8063  REVRS2       =807F
REVRS3       =8080  RND          =80A7  RNDDT        =E810
START        =8000  TM01S        =80D8  TM01S2       =80DB
TM1M         =80E4  TM1M_2       =80E7  UPDP1        =8007
WAIT         =809F
```

プログラム№5　SWAP（SWAP. 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。

```
2010/9/17  14:35  swapb.txt
END=80FF
              ; SWAP for NDZ
              ; 03/04/29 4/30 5/2 5/3 5/5 5/10
              ;10/6/2 for ND80Z3
              ;
                   ORG $8000
                   LED1=$FFF8
                   DP1=$FFF4
                   DP3=$FFF6
                   CNTR=$FFF7;=DP4
                   DTDPS=$05C0
                   KEYIN1=$0616
                   KEYIN2=$0623
                 D1_1=$02DF;about 125microsec
              ;
              ;start data disp
8000 21F180   START:LD HL,STDT
8003 11F8FF         LD DE,LED1
8006 010800         LD BC,$0008
```

```
8009 EDB0          LDIR
800B 210000        LD HL,$0000
800E 22F6FF        LD (DP3),HL;DECIMAL COUNTER clear
8011 22F4FF        LD (DP1),HL
                   ;key entry
8014 CDBA80   KEY:CALL KEYIN1S;1-7 input
8017 FE08          CP 08
8019 D21480        JP NC,KEY
801C 3D            DEC A
801D FA1480        JP M,KEY
                   ;LED blink and wait WINC in
8020 21F8FF   BLNK0:LD HL,LED1
8023 85            ADD A,L
8024 6F            LD L,A
8025 7E            LD A,(HL)
8026 B7            OR A
8027 CA1480        JP Z,KEY;space position
802A 57            LD D,A;LED data save
802B 47            LD B,A
802C 0E00          LD C,00
802E 71       BLNK1:LD (HL),C
802F CDDD80        CALL TM01S
8032 CDC280        CALL KEYIN2S;1-7,15 input
8035 FE08          CP 08
8037 D24280        JP NC,BLNK2
803A 3D            DEC A
803B FA4780        JP M,BLNK3
803E 72            LD (HL),D
803F C32080        JP BLNK0
8042 FE15     BLNK2:CP 15;WRITE INC
8044 CA4D80        JP Z,CHECK
8047 79       BLNK3:LD A,C
8048 48            LD C,B
8049 47            LD B,A
804A C32E80        JP BLNK1
                   ;check
804D 7D       CHECK:LD A,L
804E E5            PUSH HL;save HL(current position)
804F E607          AND 07
8051 CA6A80        JP Z,CHECKR;position=LED1
8054 2B            DEC HL
8055 7E            LD A,(HL)
8056 B7            OR A
8057 CA8880        JP Z,REP;left=space
805A BA            CP D
805B CA6A80        JP Z,CHECKR;left=same
805E 7D            LD A,L
805F E607          AND 07
8061 CA6A80        JP Z,CHECKR;left=LED1
8064 2B            DEC HL
8065 7E            LD A,(HL)
8066 B7            OR A
8067 CA8880        JP Z,REP;left of left=space
806A E1       CHECKR:POP HL
806B E5            PUSH HL
806C 7D            LD A,L
806D FEFE          CP FE
806F CAB680        JP Z,CANT;position=LED7,cannot replace
```

```
8072 23            INC HL
8073 7E            LD A,(HL)
8074 B7            OR A
8075 CA8880        JP Z,REP;right=space
8078 BA            CP D
8079 CAB680        JP Z,CANT;right=same,cannot replace
807C 7D            LD A,L
807D FEFE          CP FE
807F CAB680        JP Z,CANT;right=LED7,cannot replace
8082 23            INC HL
8083 7E            LD A,(HL)
8084 B7            OR A
8085 C2B680        JP NZ,CANT;right of right is not space,cannot replace
                ;replace
8088 72         REP:LD (HL),D;set current DATA to space position
8089 E1            POP HL
808A 3600          LD (HL),00;set space to current position
808C 3AF7FF        LD A,(CNTR);DECIMAL COUNTER up
808F C601          ADD A,01
8091 27            DAA
8092 32F7FF        LD (CNTR),A
                ;all check
8095 21F8FF        LD HL,LED1
8098 11F980        LD DE,ENDDT
809B 0607          LD B,07
809D 1A         ALCHK1:LD A,(DE)
809E BE            CP (HL)
809F C21480        JP NZ,KEY;not success
80A2 23            INC HL
80A3 13            INC DE
80A4 10F7          DJNZ *ALCHK1
                ;OK! ALL LED blink and COUNTER disp
80A6 CDCA80        CALL ALBLNK
80A9 CDC005        CALL DTDPS
80AC 0614          LD B,14;=20 ,2sec wait
80AE CDDD80     WAIT:CALL TM01S
80B1 10FB          DJNZ WAIT
80B3 C30080        JP START
                ;cannot replace
80B6 E1         CANT:POP HL
80B7 C34780        JP BLNK3
                ;
                ;KEYIN with DE,BC save
80BA C5         KEYIN1S:PUSH BC
80BB D5            PUSH DE
80BC CD1606        CALL KEYIN1
80BF D1            POP DE
80C0 C1            POP BC
80C1 C9            RET
80C2 C5         KEYIN2S:PUSH BC
80C3 D5            PUSH DE
80C4 CD2306        CALL KEYIN2
80C7 D1            POP DE
80C8 C1            POP BC
80C9 C9            RET
                ;
                ;LED all blink
80CA 060A       ALBLNK:LD B,0A;=10
```

```
80CC 3EEF     ALBLNK1:LD A,EF
80CE D398        OUT (98),A
80D0 CDDD80        CALL TM01S
80D3 3EFF        LD A,FF
80D5 D398        OUT (98),A
80D7 CDDD80        CALL TM01S
80DA 10F0          DJNZ *ALBLNK1
80DC C9            RET
             ;
             ;0.1sec timer
80DD D5      TM01S:PUSH DE
80DE 1E64          LD E,64;=100
80E0 CDE980  TM01S2:CALL TM1M
80E3 1D            DEC E
80E4 C2E080        JP NZ,TM01S2
80E7 D1            POP DE
80E8 C9            RET
80E9 D5      TM1M:PUSH DE
80EA 1608        LD D,08
80EC CDDF02  TM1M_2:CALL D1_1
80EF D1          POP DE
80F0 C9          RET
             ;
80F1 5C      STDT:DB 5C
80F2 5C            DB 5C
80F3 5C            DB 5C
80F4 00            DB 00
80F5 63            DB 63
80F6 63            DB 63
80F7 63            DB 63
80F8 00            DB 00
             ;
80F9 63      ENDDT:DB 63
80FA 63            DB 63
80FB 63            DB 63
80FC 00            DB 00
80FD 5C            DB 5C
80FE 5C            DB 5C
80FF 5C            DB 5C
             ;E
ALBLNK       =80CA  ALBLNK1      =80CC  ALCHK1       =809D
BLNK0        =8020  BLNK1        =802E  BLNK2        =8042
BLNK3        =8047  CANT         =80B6  CHECK        =804D
CHECKR       =806A  CNTR         =FFF7  D1_1         =02DF
DP1          =FFF4  DP3          =FFF6  DTDPS        =05C0
ENDDT        =80F9  KEY          =8014  KEYIN1       =0616
KEYIN1S      =80BA  KEYIN2       =0623  KEYIN2S      =80C2
LED1         =FFF8  REP          =8088  START        =8000
STDT         =80F1  TM01S        =80DD  TM01S2       =80E0
TM1M         =80E9  TM1M_2       =80EC  WAIT         =80AE
```

プログラム№6　CATCH（CATCH. 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（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（IXレジスタ）、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（FFF8～FFFF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DEC、＋１はINCを普通は使います。あるいはSUB　01かADD　A，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。SUB　01の代わりにADD　A，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　A，Cで切りぬけているのです。もうひとつ、加算後にOR　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。

さてこのサブルーチンでは普通あまり使われないIXレジスタが登場しています。同種のレジスタにIYがあります。ともに８０８０には無いレジスタでHL，BC，DEとは少し違った使い方をします。インデックスレジスタといいます。ここではレジスタ不足をおぎなうためにHLレジスタの代用として使っています。このあたりは８０８０では面倒なことになります（Z80の有難いところです）。

Z80はIXやIYのほかに、通常のレジスタのHL，BC，DE，AFと全く同じレジスタをもう 1 組もっています（ウラレジスタなどと言います）。これだけのレジスタがあるとかなり助かる場合があります。ただし裏レジスタは注意して扱わないとわけのわからないバグに悩まされることにもなりかねないため、慣れないうちは使わない方がよいでしょう。IXとIYは便利ですからレジスタが不足するときにHLレジスタの代わりにどんどん使いましょう。

```
2010/9/17  14:28  catchb.txt
END=814E
              ; CATCH for NDZ
              ; 03/05/06 5/10
              ;10/6/2 for ND80Z3
              ;
                   ORG $8000
                   MYADRS=$E800
                   YRADRS=$E801
                   RNDDT=$E810;$E811
                   LED1=$FFF8
              ;
                   KEYIN2=$0623
                   D1_1=$02DF;about 125microsec
              ;
              ;data set
8000 CDAA80   START:CALL CLR
              ;my position select
8003 1E23          LD E,23;cap
8005 0100E8        LD BC,MYADRS
8008 CDB580        CALL DTSET
              ;your posision select
```

```
800B 1E1C          LD E,1C;cup
800D 03            INC BC
800E CDB580        CALL DTSET
              ;GAME START
              ;your turn
8011 3A01E8   REENT:LD A,(YRADRS)
8014 21F8FF        LD HL,LED1
8017 85            ADD A,L
8018 6F            LD L,A
8019 56            LD D,(HL);LED data save
801A 42            LD B,D
801B 0E00          LD C,00
              ;blink and keyin
801D 71       KEY:LD (HL),C
801E CD3B81        CALL TM01S
8021 CDE780        CALL KEYIN2S
8024 B7            OR A
8025 CA3380        JP Z,LEFT
8028 FE15          CP 15;W INC
802A CA3880        JP Z,RIGHT
802D 59            LD E,C
802E 48            LD C,B
802F 43            LD B,E
8030 C31D80        JP KEY
8033 0EFF     LEFT:LD C,FF
8035 C33A80        JP RIGHT2
8038 0E01     RIGHT:LD C,01
803A 111CE3   RIGHT2:LD DE,$E31C;see STEP comment line
803D DD2101E8      LD IX,YRADRS
8041 CDC480        CALL STEP
              ;catch?
8044 DD2B          DEC IX;MYADRS
8046 DDBE00        CP (IX+00)
8049 C25580        JP NZ,MYTURN
              ;catch!
804C 7E            LD A,(HL)
804D E6DC          AND DC;cap clear
804F 77            LD (HL),A
8050 3E17          LD A,17;piiii
8052 C37B80        JP END
              ;my turn
8055 CD8C80   MYTURN:CALL RND;LEFT(FF) or RIGHT(01) select
8058 E601          AND 01
805A C25E80        JP NZ,MYTURN1;=01
805D 3D            DEC A;=FF
805E 4F       MYTURN1:LD C,A
805F DD7E00        LD A,(IX+00)
8062 21F8FF        LD HL,LED1
8065 85            ADD A,L
8066 6F            LD L,A
8067 1123DC        LD DE,$DC23;see STEP comment line
806A CDC480        CALL STEP
              ;catch?
806D DD23          INC IX;YRADRS
806F DDBE00        CP (IX+00)
8072 C21180        JP NZ,REENT
              ;catch!
8075 7E            LD A,(HL)
```

```
8076 E6E3          AND E3;cup clear
8078 77            LD (HL),A
8079 3E00          LD A,00;buuuu
807B 0605     END:LD B,05
807D CDEF80   END1:CALL SOUND
8080 10FB          DJNZ *END1
8082 0614          LD B,14;=20 ,2sec wait
8084 CD3B81   WAIT:CALL TM01S
8087 10FB          DJNZ WAIT
8089 C30080        JP START
               ;
               ;ransu
808C E5       RND:PUSH HL
808D D5            PUSH DE
808E 2A10E8        LD HL,(RNDDT)
8091 ED5F          LD A,R
8093 5F            LD E,A
8094 19            ADD HL,DE
8095 7C            LD A,H
8096 E603          AND 03
8098 67            LD H,A
8099 7E            LD A,(HL)
809A 23            INC HL
809B 86            ADD A,(HL)
809C 5F            LD E,A
809D 23            INC HL
809E 86            ADD A,(HL)
809F 23            INC HL
80A0 86            ADD A,(HL)
80A1 57            LD D,A
80A2 ED5310E8      LD (RNDDT),DE
80A6 7A            LD A,D
80A7 D1            POP DE
80A8 E1            POP HL
80A9 C9            RET
               ;
               ;LED all clear
80AA 21F8FF   CLR:LD HL,LED1
80AD AF            XOR A
80AE 0608          LD B,08
80B0 77       CLR1:LD (HL),A
80B1 23            INC HL
80B2 10FC          DJNZ *CLR1
80B4 C9            RET
               ;
               ;start data set;BC=MYADRS or YRADRS;E=disp data 23(cap) or 1C(cup)
80B5 CD8C80   DTSET:CALL RND
80B8 E607          AND 07
80BA 02            LD (BC),A
80BB 21F8FF        LD HL,LED1
80BE 85            ADD A,L
80BF 6F            LD L,A
80C0 7B            LD A,E
80C1 B6            OR (HL)
80C2 77            LD (HL),A;cap or cup or both disp
80C3 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
80C4 CD8C80     STEP:CALL RND
80C7 E607            AND 07
80C9 3C              INC A
80CA FE07            CP 07
80CC D2C480          JP NC,STEP
80CF 47              LD B,A
80D0 CD3881     STEP1:CALL TM02S
80D3 7E              LD A,(HL)
80D4 A2              AND D;clear
80D5 77              LD (HL),A
80D6 7D              LD A,L
80D7 81              ADD A,C
80D8 F6F8            OR F8
80DA 6F              LD L,A
80DB 7E              LD A,(HL)
80DC B3              OR E;disp
80DD 77              LD (HL),A
80DE 10F0            DJNZ STEP1
80E0 7D              LD A,L
80E1 E607            AND 07
80E3 DD7700          LD (IX+00),A
80E6 C9              RET
                ;
                ;KEYIN with DE,BC save
80E7 C5         KEYIN2S:PUSH BC
80E8 D5            PUSH DE
80E9 CD2306        CALL KEYIN2
80EC D1            POP DE
80ED C1            POP BC
80EE C9            RET
                ;
                ;
80EF F5         SOUND:PUSH AF
80F0 E5            PUSH HL
80F1 D5            PUSH DE
80F2 C5            PUSH BC
80F3 212081        LD HL,SNDTBL
80F6 85            ADD A,L
80F7 6F            LD L,A
80F8 46            LD B,(HL)
80F9 1E1A          LD E,1A
80FB 50         SNDS1:LD D,B
80FC 3EFF          LD A,FF;SP OUT=H
80FE D398          OUT (98),A
8100 E5         SNDS2:PUSH HL;11clk---------
8101 E5            PUSH HL;11clk            |11+11+10+10+4+4+10=60
8102 E1                 POP HL;10clk        |60/6=10
8103 E1                 POP HL;10clk        |
8104 00                 NOP;4clk            |10microsec
8105 15            DEC D;4clk               |
8106 C20081        JP NZ,SNDS2;10clk---
8109 50            LD D,B
810A 3EDF          LD A,DF;SP OUT=L
810C D398          OUT (98),A
810E E5         SNDS3:PUSH HL;11clk---------
```

```
810F E5          PUSH HL;11clk            |11+11+10+10+4+4+10=60
8110 E1               POP HL;10clk        |60/6=10
8111 E1               POP HL;10clk        |
8112 00               NOP;4clk            |10microsec
8113 15          DEC D;4clk               |
8114 C20E81      JP NZ,SNDS3;10clk---
8117 1D          DEC E
8118 C2FB80      JP NZ,SNDS1
811B C1          POP BC
811C D1          POP DE
811D E1          POP HL
811E F1          POP AF
811F C9          RET
                 ; SOUND TABLE
8120 7F          SNDTBL:DB 7F;so4
8121 77             DB 77;so#4
8122 71             DB 71;ra4
8123 6A             DB 6A;ra#4
8124 5F             DB 5F;do5
8125 59             DB 59;do#5
8126 54             DB 54;re5
8127 4F             DB 4F;re#
8128 47             DB 47;fA5
8129 43             DB 43;fa#5
812A 3F             DB 3F;so5
812B 3B             DB 3B;so5#
812C 35             DB 35;ra#5
812D 32             DB 32;si5
812E 2F             DB 2F;do6
812F 2C             DB 2C;do#6
8130 25             DB 25;mi6
8131 27             DB 27;re#6
8132 2A             DB 2A;re6
8133 4B             DB 4B;mi5
8134 38             DB 38;ra5
8135 64             DB 64;si4
8136 23             DB 23;fa6
8137 21             DB 21;fa#6
                 ;
                 ;0.2sec timer
8138 CD3B81      TM02S:CALL TM01S
                 ;0.1sec timer
813B D5          TM01S:PUSH DE
813C 1E64             LD E,64;=100
813E CD4781      TM01S2:CALL TM1M
8141 1D               DEC E
8142 C23E81           JP NZ,TM01S2
8145 D1               POP DE
8146 C9               RET
8147 D5          TM1M:PUSH DE
8148 1608           LD D,08
814A CDDF02      TM1M_2:CALL D1_1
814D D1             POP DE
814E C9             RET
                 ;CLR          =80AA  CLR1         =80B0  D1_1         =02DF
DTSET            =80B5  END          =807B  END1         =807D
IX               =0474  KEY          =801D  KEYIN2       =0623
KEYIN2S          =80E7  LED1         =FFF8  LEFT         =8033
```

MYADRS	=E800	MYTURN	=8055	MYTURN1	=805E
REENT	=8011	RIGHT	=8038	RIGHT2	=803A
RND	=808C	RNDDT	=E810	SNDS1	=80FB
SNDS2	=8100	SNDS3	=810E	SNDTBL	=8120
SOUND	=80EF	START	=8000	STEP	=80C4
STEP1	=80D0	TM01S	=813B	TM01S2	=813E
TM02S	=8138	TM1M	=8147	TM1M_2	=814A
WAIT	=8084	YRADRS	=E801		

プログラムNo.7　もぐらたたき（MOGURA. 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（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、AND　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OR　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JPでひとつにまとめるのがプログラムのコツです。

```
2010/9/17  17:25  MOGURAC.TXT
END=8157
              ; MOGURATATAKI for NDZ (HIT'N BLOW)
              ; 03/5/2 5/5 5/11 5/12
              ;10/6/2 for ND80Z3
              ;7/21 9/17
              ;
                   ORG $8000
                   RNDDT=$E810;$E811
                   MCOUNT=$E812
                   LED1=$FFF8
                   DP1=$FFF4
                   DP3=$FFF6
                   CNTR=$FFF7;=DP4
              ;
                   DTDPS=$05C0
                   KEYIN2=$0623
                 D1_1=$02DF;about 125microsec
              ;
              ;start data disp
8000 210000   START:LD HL,$0000
8003 22F6FF        LD (DP3),HL;DECIMAL COUNTER clear
8006 22F4FF        LD (DP1),HL
8009 3E14          LD A,14;=20
800B 3212E8        LD (MCOUNT),A;mogura
              ;wait 2sec,if keyin then buuuu
800E CDDE80   MOGURA:CALL STDTDP
```

```
8011 01D007       LD BC,$07D0;=2000 ,wait 2000msec
8014 CD0781   MOGURA01:CALL TM1M
8017 CDE980       CALL KEYIN2S
801A FE09         CP 09
801C D22B80       JP NC,MOGURA02
801F 3D           DEC A
8020 FA2B80       JP M,MOGURA02
8023 0E00         LD C,00;sound buuuu
8025 CD9080       CALL HAMMER
8028 C30E80       JP MOGURA
802B 0B       MOGURA02:DEC BC
802C 78           LD A,B
802D B1           OR C
802E C21480       JP NZ,MOGURA01
                  ;"MOGURA" up
8031 CDAD80   MOGURA1:CALL RND
8034 E607         AND 07
8036 57           LD D,A;position save
8037 21F8FF       LD HL,LED1
803A 85           ADD A,L
803B 6F           LD L,A
803C 3654         LD (HL),54;MOGURA disp
803E 01E803       LD BC,$03E8;=1000 ,wait 1000msec
8041 CDE980   MOGURA2:CALL KEYIN2S
8044 FE09         CP 09
8046 D25980       JP NC,NOT
8049 3D           DEC A
804A FA5980       JP M,NOT
804D BA           CP D
804E CA6F80       JP Z,HIT
                  ;BUUUUU
8051 0E00         LD C,00;sound buuuuu
8053 CD9080       CALL HAMMER
8056 C36280       JP NOT1
                  ;not keyin
8059 CD0781   NOT:CALL TM1M
805C 0B           DEC BC
805D 78           LD A,B
805E B1           OR C
805F C24180       JP NZ,MOGURA2
8062 3A12E8   NOT1:LD A,(MCOUNT)
8065 3D           DEC A
8066 CA8080       JP Z,END
8069 3212E8       LD (MCOUNT),A
806C C30E80       JP MOGURA
                  ;hit!
806F 0E17     HIT:LD C,17;sound piiii
8071 CD9080       CALL HAMMER
8074 3AF7FF       LD A,(CNTR);DECIMAL COUNTER up
8077 C601         ADD A,01
8079 27           DAA
807A 32F7FF       LD (CNTR),A
807D C36280       JP NOT1
                  ;game over! COUNTER disp and ALL LED blink
8080 CDC005   END:CALL DTDPS
8083 CDCB80       CALL ALBLNK
8086 0614         LD B,14;=20 ,2sec wait
8088 CDFB80   WAIT:CALL TM01S
```

```
808B 10FB          DJNZ WAIT
808D C30080        JP START
                   ;
                   ;hammer disp
8090 21F8FF   HAMMER:LD HL,LED1
8093 85            ADD A,L
8094 6F            LD L,A
8095 7E            LD A,(HL)
8096 F601          OR 01;hammer up
8098 77            LD (HL),A
8099 CDF180        CALL TM05S
809C E614          AND 14;hammer down
809E F608          OR 08
80A0 77            LD (HL),A
80A1 79            LD A,C
80A2 0603          LD B,03
80A4 CD0F81   HAMMER1:CALL SOUND
80A7 10FB          DJNZ HAMMER1
80A9 CDF180        CALL TM05S
80AC C9            RET
                   ;
                   ;ransu
80AD E5       RND:PUSH HL
80AE D5            PUSH DE
80AF 2A10E8        LD HL,(RNDDT)
80B2 ED5F          LD A,R
80B4 5F            LD E,A
80B5 19            ADD HL,DE
80B6 7C            LD A,H
80B7 E603          AND 03
80B9 67            LD H,A
80BA 7E            LD A,(HL)
80BB 23            INC HL
80BC 86            ADD A,(HL)
80BD 5F            LD E,A
80BE 23            INC HL
80BF 86            ADD A,(HL)
80C0 23            INC HL
80C1 86            ADD A,(HL)
80C2 57            LD D,A
80C3 ED5310E8      LD (RNDDT),DE
80C7 7A            LD A,D
80C8 D1            POP DE
80C9 E1            POP HL
80CA C9            RET
                   ;
                   ;LED all blink
80CB 060A     ALBLNK:LD B,0A;=10
80CD 3EEF     ALBLNK1:LD A,EF
80CF D398        OUT (98),A
80D1 CDFB80        CALL TM01S
80D4 3EFF          LD A,FF
80D6 D398        OUT (98),A
80D8 CDFB80        CALL TM01S
80DB 10F0          DJNZ *ALBLNK1
80DD C9            RET
                   ;
                   ;start data disp
```

```
80DE 21F8FF   STDTDP:LD HL,LED1
80E1 0608          LD B,08
80E3 3608     STDTDP1:LD (HL),08
80E5 23            INC HL
80E6 10FB          DJNZ STDTDP1
80E8 C9            RET
              ;
              ;KEYIN with DE,BC save
80E9 C5       KEYIN2S:PUSH BC
80EA D5          PUSH DE
80EB CD2306      CALL KEYIN2
80EE D1          POP DE
80EF C1          POP BC
80F0 C9          RET
              ;
              ;0.5sec timer
80F1 C5       TM05S:PUSH BC
80F2 0605          LD B,05
80F4 CDFB80   TM05S1:CALL TM01S
80F7 10FB          DJNZ TM05S1
80F9 C1            POP BC
80FA C9            RET
              ;0.1sec timer
80FB D5       TM01S:PUSH DE
80FC 1E64          LD E,64;=100
80FE CD0781   TM01S2:CALL TM1M
8101 1D            DEC E
8102 C2FE80        JP NZ,TM01S2
8105 D1            POP DE
8106 C9            RET
8107 D5       TM1M:PUSH DE
8108 1608        LD D,08
810A CDDF02   TM1M_2:CALL D1_1
810D D1          POP DE
810E C9          RET
              ;
810F F5       SOUND:PUSH AF
8110 E5          PUSH HL
8111 D5          PUSH DE
8112 C5          PUSH BC
8113 214081      LD HL,SNDTBL
8116 85          ADD A,L
8117 6F          LD L,A
8118 46          LD B,(HL)
8119 1E1A        LD E,1A
811B 50       SNDS1:LD D,B
811C 3EFF        LD A,FF;SP OUT=H
811E D398        OUT (98),A
8120 E5       SNDS2:PUSH HL;11clk---------
8121 E5          PUSH HL;11clk            |11+11+10+10+4+4+10=60
8122 E1               POP HL;10clk        |60/6=10
8123 E1               POP HL;10clk        |
8124 00               NOP;4clk            |10microsec
8125 15          DEC D;4clk               |
8126 C22081      JP NZ,SNDS2;10clk---
8129 50          LD D,B
812A 3EDF        LD A,DF;SP OUT=L
812C D398        OUT (98),A
```

```
812E E5       SNDS3:PUSH HL;11clk---------
812F E5          PUSH HL;11clk            |11+11+10+10+4+4+10=60
8130 E1               POP HL;10clk        |60/6=10
8131 E1               POP HL;10clk        |
8132 00               NOP;4clk            |10microsec
8133 15          DEC D;4clk               |
8134 C22E81      JP NZ,SNDS3;10clk---
8137 1D          DEC E
8138 C21B81      JP NZ,SNDS1
813B C1          POP BC
813C D1          POP DE
813D E1          POP HL
813E F1          POP AF
813F C9          RET
                 ; SOUND TABLE
8140 7F       SNDTBL:DB 7F;so4
8141 77          DB 77;so#4
8142 71          DB 71;ra4
8143 6A          DB 6A;ra#4
8144 5F          DB 5F;do5
8145 59          DB 59;do#5
8146 54          DB 54;re5
8147 4F          DB 4F;re#
8148 47          DB 47;fA5
8149 43          DB 43;fa#5
814A 3F          DB 3F;so5
814B 3B          DB 3B;so5#
814C 35          DB 35;ra#5
814D 32          DB 32;si5
814E 2F          DB 2F;do6
814F 2C          DB 2C;do#6
8150 25          DB 25;mi6
8151 27          DB 27;re#6
8152 2A          DB 2A;re6
8153 4B          DB 4B;mi5
8154 38          DB 38;ra5
8155 64          DB 64;si4
8156 23          DB 23;fa6
8157 21          DB 21;fa#6
                 ;

ALBLNK       =80CB  ALBLNK1      =80CD  CNTR         =FFF7
D1_1         =02DF  DP1          =FFF4  DP3          =FFF6
DTDPS        =05C0  END          =8080  HAMMER       =8090
HAMMER1      =80A4  HIT          =806F  KEYIN2       =0623
KEYIN2S      =80E9  LED1         =FFF8  MCOUNT       =E812
MOGURA       =800E  MOGURA01     =8014  MOGURA02     =802B
MOGURA1      =8031  MOGURA2      =8041  NOT          =8059
NOT1         =8062  RND          =80AD  RNDDT        =E810
SNDS1        =811B  SNDS2        =8120  SNDS3        =812E
SNDTBL       =8140  SOUND        =810F  START        =8000
STDTDP       =80DE  STDTDP1      =80E3  TM01S        =80FB
TM01S2       =80FE  TM05S        =80F1  TM05S1       =80F4
TM1M         =8107  TM1M_2       =810A  WAIT         =8088
```

プログラム№8　神経衰弱（SINKESUI. 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。OR　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　A, L、LD　L, Aとしないで、いきなりLD　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E800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　A, 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。

```
2010/9/17  14:34  sinksuib.txt
END=8113
              ; SINKEISUIJAKU for NDZ (ODD COUPLES)
              ; 03/04/29 4/30 5/1 5/5 5/12 5/24
              ;10/6/2 for ND80Z3
              ;
                   ORG $8000
                   DTBF=$E800
                   RNDDT=$E810;$E811
                   LED1=$FFF8
                   DP1=$FFF4
                   DP3=$FFF6
                   CNTR=$FFF7;=DP4
```

```
                ;
                     DTDPS=$05C0
                     KEYIN1=$0616
                   D1_1=$02DF;about 125microsec
                ;
                ;data set
8000 AF         START:XOR A
8001 0608            LD B,08
8003 2100E8          LD HL,DTBF
8006 77         DTST1:LD (HL),A
8007 23              INC HL
8008 10FC            DJNZ *DTST1
800A 0E04            LD C,04
                ;No. select
800C CD8380     DTST2:CALL RND
800F E60F            AND 0F
8011 21E880          LD HL,SEGDT
8014 5F              LD E,A
8015 1600            LD D,00
8017 19              ADD HL,DE
8018 7E              LD A,(HL)
8019 2100E8          LD HL,DTBF
801C 0608            LD B,08
801E BE         DTST3:CP (HL)
801F CA0C80          JP Z,DTST2;already selected No.
8022 23              INC HL
8023 10F9            DJNZ *DTST3
8025 57              LD D,A;DATA save
                ;ADDRESS select and DATA set
8026 0602            LD B,02
8028 CD8380     ADST1:CALL RND
802B E607            AND 07
802D 2100E8          LD HL,DTBF
8030 85              ADD A,L
8031 6F              LD L,A
8032 7E              LD A,(HL)
8033 B7              OR A
8034 C22880          JP NZ,ADST1;already used
8037 72              LD (HL),D;DATA set
8038 10EE            DJNZ *ADST1
803A 0D              DEC C
803B C20C80          JP NZ,DTST2
                ;GAME START
803E 210000          LD HL,$0000
8041 22F6FF          LD (DP3),HL;DECIMAL COUNTER clear
8044 22F4FF          LD (DP1),HL
                ;key entry(first one)
8047 CDB480     REENT:CALL CLR
804A CDC780     KEY:CALL KEYDP
804D F5              PUSH AF;DATA and S-FLAG save
804E 3AF7FF          LD A,(CNTR);DECIMAL COUNTER up
8051 C601            ADD A,01
8053 27              DAA
8054 32F7FF          LD (CNTR),A
8057 F1              POP AF;DATA and S-FLAG
8058 FA4780          JP M,REENT;otetuki
805B 47              LD B,A;first DATA save
                ;key entry(second ONE)
```

```
805C CDC780   KEY2:CALL KEYDP
805F FA4780        JP M,REENT;otetuki
              ;DATA compere
8062 B8            CP B; B=first data
8063 C24780        JP NZ,REENT;not same
              ;check all open or not
8066 2100E8        LD HL,DTBF
8069 0608          LD B,08
806B 7E       OPNCK1:LD A,(HL)
806C B7            OR A
806D F24A80        JP P,KEY;not open data,continue GAME
8070 23            INC HL
8071 10F8          DJNZ OPNCK1
              ;OK! ALL LED blink
8073 CDA180        CALL ALBLNK
8076 CDC005        CALL DTDPS;COUNTER disp
8079 0614          LD B,14;=20 ,2sec wait
807B CD0081   WAIT:CALL TM01S
807E 10FB          DJNZ WAIT
8080 C30080        JP START
              ;
              ;ransu
8083 E5       RND:PUSH HL
8084 D5            PUSH DE
8085 2A10E8        LD HL,(RNDDT)
8088 ED5F          LD A,R
808A 5F            LD E,A
808B 19            ADD HL,DE
808C 7C            LD A,H
808D E603          AND 03
808F 67            LD H,A
8090 7E            LD A,(HL)
8091 23            INC HL
8092 86            ADD A,(HL)
8093 5F            LD E,A
8094 23            INC HL
8095 86            ADD A,(HL)
8096 23            INC HL
8097 86            ADD A,(HL)
8098 57            LD D,A
8099 ED5310E8      LD (RNDDT),DE
809D 7A            LD A,D
809E D1            POP DE
809F E1            POP HL
80A0 C9            RET
              ;
              ;LED all blink
80A1 060A     ALBLNK:LD B,0A;=10
80A3 3EEF     ALBLNK1:LD A,EF
80A5 D398        OUT (98),A
80A7 CD0081        CALL TM01S
80AA 3EFF          LD A,FF
80AC D398        OUT (98),A
80AE CD0081        CALL TM01S
80B1 10F0          DJNZ *ALBLNK1
80B3 C9            RET
              ;
              ;LED clear and open mark clear
```

```
80B4 21F8FF   CLR:LD HL,LED1
80B7 1100E8       LD DE,DTBF
80BA 010008       LD BC,$0800
80BD 71       CLR2:LD (HL),C
80BE 1A           LD A,(DE)
80BF E67F         AND 7F
80C1 12           LD (DE),A
80C2 23           INC HL
80C3 13           INC DE
80C4 10F7         DJNZ *CLR2
80C6 C9           RET
                  ;
                  ;KEYIN and DATA disp
80C7 CDF880   KEYDP:CALL KEYIN1S
80CA FE09         CP 09
80CC D2C780       JP NC,KEYDP
80CF 3D           DEC A
80D0 FAC780       JP M,KEYDP
80D3 2100E8   DTDP:LD HL,DTBF
80D6 6F           LD L,A
80D7 7E           LD A,(HL)
80D8 B7           OR A
80D9 F8           RET M;otetuki!
80DA F680         OR 80
80DC 77           LD (HL),A;open mark set
80DD E67F         AND 7F
80DF 11F8FF       LD DE,LED1
80E2 2600         LD H,00
80E4 19           ADD HL,DE
80E5 77           LD (HL),A
80E6 B7           OR A;clear Sign-flag
80E7 C9           RET
80E8 5C       SEGDT:DB 5C;0
80E9 06           DB 06;1
80EA 5B           DB 5B;2
80EB 4F           DB 4F;3
80EC 66           DB 66;4
80ED 6D           DB 6D;5
80EE 7D           DB 7D;6
80EF 27           DB 27;7
80F0 7F           DB 7F;8
80F1 6F           DB 6F;9
80F2 77           DB 77;A
80F3 7C           DB 7C;b
80F4 39           DB 39;C
80F5 5E           DB 5E;d
80F6 79           DB 79;E
80F7 71           DB 71;F
                  ;
                  ;KEYIN with DE,BC save
80F8 C5       KEYIN1S:PUSH BC
80F9 D5          PUSH DE
80FA CD1606      CALL KEYIN1
80FD D1          POP DE
80FE C1          POP BC
80FF C9          RET
                  ;
                  ;0.1sec timer
```

```
8100 D5       TM01S:PUSH DE
8101 1E64          LD E,64;=100
8103 CD0C81   TM01S2:CALL TM1M
8106 1D            DEC E
8107 C20381        JP NZ,TM01S2
810A D1            POP DE
810B C9            RET
810C D5       TM1M:PUSH DE
810D 1608        LD D,08
810F CDDF02   TM1M_2:CALL D1_1
8112 D1          POP DE
8113 C9          RET
              ;;
ADST1        =8028  ALBLNK       =80A1  ALBLNK1      =80A3
CLR          =80B4  CLR2         =80BD  CNTR         =FFF7
D1_1         =02DF  DP1          =FFF4  DP3          =FFF6
DTBF         =E800  DTDP         =80D3  DTDPS        =05C0
DTST1        =8006  DTST2        =800C  DTST3        =801E
KEY          =804A  KEY2         =805C  KEYDP        =80C7
KEYIN1       =0616  KEYIN1S      =80F8  LED1         =FFF8
OPNCK1       =806B  REENT        =8047  RND          =8083
RNDDT        =E810  SEGDT        =80E8  START        =8000
TM01S        =8100  TM01S2       =8103  TM1M         =810C
TM1M_2       =810F  WAIT         =807B
```

プログラム№9　WANTED（WANTED．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。

```
2010/9/17  17:47  WANTEDC.TXT
END=8170
                ; WANTED for NDZ
                ; 03/05/06 5/12
                ;10/6/3 for ND80Z3
                ;7/21 9/17
                ;
                      ORG $8000
                      MYDT=$E800
                      ACC=$E802
                      CNTR=$E805
                      RNDDT=$E810;$E811
                      DTRG=$FFEC
                      ADRG=$FFEE
                      LED1=$FFF8
                      LED3=$FFFA
                      DP1=$FFF4
                      DP3=$FFF6
                ;
                      ADDPS=$05A1
                      KEYIN1=$0616
                      D1_1=$02DF;about 125microsec
                ;
                ;data set
8000 AF         START:XOR A
8001 3205E8           LD (CNTR),A;COUNTER clear
8004 67               LD H,A
8005 6F               LD L,A
8006 22ECFF           LD (DTRG),HL
8009 22EEFF           LD (ADRG),HL
                ;data set
800C 0604             LD B,04
800E CDEE80     DTST1:CALL RND
```

```
8011 E60F          AND 0F
8013 FE0A          CP 0A
8015 D20E80        JP NC,DTST1
8018 29            ADD HL,HL
8019 29            ADD HL,HL
801A 29            ADD HL,HL
801B 29            ADD HL,HL
801C B5            OR L
801D 6F            LD L,A
801E 10EE          DJNZ *DTST1
8020 2200E8        LD (MYDT),HL
8023 2202E8        LD (ACC),HL
             ;GAME START
8026 3E10    START1:LD A,10;puuuu
8028 060A          LD B,0A
802A CD2881  START2:CALL SOUND
802D 10FB          DJNZ START2
802F CDA105  KEY:CALL ADDPS
8032 AF            XOR A
8033 32F8FF        LD (LED1),A
8036 32FAFF        LD (LED3),A
8039 2AECFF  KEY1:LD HL,(DTRG)
803C CD0C81  KEY2:CALL KEYIN1S
803F FE15          CP 15;W INC
8041 CA5580        JP Z,ADD
8044 FE0A          CP 0A
8046 D23C80        JP NC,KEY2
8049 29            ADD HL,HL
804A 29            ADD HL,HL
804B 29            ADD HL,HL
804C 29            ADD HL,HL
804D B5            OR L
804E 6F            LD L,A
804F 22ECFF        LD (DTRG),HL
8052 C32F80        JP KEY
             ;DATA add and check
8055 3A05E8  ADD:LD A,(CNTR)
8058 C601          ADD A,01;DECIMAL COUNTER up
805A 27            DAA
805B 3205E8        LD (CNTR),A
805E ED5B02E8      LD DE,(ACC)
8062 7B            LD A,E
8063 85            ADD A,L
8064 27            DAA
8065 6F            LD L,A
8066 7A            LD A,D
8067 8C            ADC A,H
8068 27            DAA
8069 DAAF80        JP C,OVER
806C 67            LD H,A
806D 2202E8        LD (ACC),HL
             ;check
8070 1600          LD D,00
8072 0E04          LD C,04
8074 0604    CK1:LD B,04
8076 29      CK2:ADD HL,HL
8077 17            RLA
8078 10FC          DJNZ *CK2
```

```
807A E60F          AND 0F
807C FE09          CP 09
807E C28280        JP NZ,CK3
8081 14            INC D;count "9"
8082 0D       CK3:DEC C
8083 C27480        JP NZ,CK1
8086 7A            LD A,D
8087 FE04          CP 04
8089 CAC180        JP Z,OK
              ;one No.(except "9") select
808C 2A02E8   CK4:LD HL,(ACC)
808F CDEE80        CALL RND
8092 E603          AND 03
8094 3C            INC A
8095 4F            LD C,A
8096 0604     CK5:LD B,04
8098 29       CK6:ADD HL,HL
8099 17            RLA
809A 10FC          DJNZ *CK6
809C 0D            DEC C
809D C29680        JP NZ,CK5
80A0 E60F          AND 0F
80A2 FE09          CP 09
80A4 CA8C80        JP Z,CK4
80A7 5F            LD E,A
              ;result disp
80A8 ED53EEFF      LD (ADRG),DE
80AC C32680        JP START1
80AF 3E00     OVER:LD A,00;buuuu
80B1 0605          LD B,05
80B3 CD2881   OVER1:CALL SOUND
80B6 10FB          DJNZ OVER1
80B8 21FFFF        LD HL,$FFFF
80BB 22EEFF        LD (ADRG),HL
80BE C32F80        JP KEY
              ;
              ;result="9999"
80C1 3E17     OK:LD A,17;piiii
80C3 0605          LD B,05
80C5 CD2881   OK1:CALL SOUND
80C8 10FB          DJNZ OK1
80CA 2A02E8        LD HL,(ACC)
80CD 22EEFF        LD (ADRG),HL
80D0 CDA105        CALL ADDPS
80D3 CD0C81        CALL KEYIN1S
80D6 2A00E8        LD HL,(MYDT)
80D9 22EEFF        LD (ADRG),HL
80DC 3A05E8        LD A,(CNTR)
80DF 6F            LD L,A
80E0 2600          LD H,00
80E2 22ECFF        LD (DTRG),HL
80E5 CDA105        CALL ADDPS
80E8 CD0C81        CALL KEYIN1S
80EB C30080        JP START
              ;
              ;ransu
80EE E5       RND:PUSH HL
80EF D5            PUSH DE
```

```
80F0 2A10E8        LD HL,(RNDDT)
80F3 ED5F          LD A,R
80F5 5F            LD E,A
80F6 19            ADD HL,DE
80F7 7C            LD A,H
80F8 E603          AND 03
80FA 67            LD H,A
80FB 7E            LD A,(HL)
80FC 23            INC HL
80FD 86            ADD A,(HL)
80FE 5F            LD E,A
80FF 23            INC HL
8100 86            ADD A,(HL)
8101 23            INC HL
8102 86            ADD A,(HL)
8103 57            LD D,A
8104 ED5310E8      LD (RNDDT),DE
8108 7A            LD A,D
8109 D1            POP DE
810A E1            POP HL
810B C9            RET
                ;
                ;KEYIN with DE,BC save
810C C5         KEYIN1S:PUSH BC
810D D5            PUSH DE
810E CD1606        CALL KEYIN1
8111 D1            POP DE
8112 C1            POP BC
8113 C9            RET
                ;
                ;0.1sec timer
8114 D5         TM01S:PUSH DE
8115 1E64            LD E,64;=100
8117 CD2081     TM01S2:CALL TM1M
811A 1D              DEC E
811B C21781          JP NZ,TM01S2
811E D1              POP DE
811F C9              RET
8120 D5         TM1M:PUSH DE
8121 1608          LD D,08
8123 CDDF02     TM1M_2:CALL D1_1
8126 D1            POP DE
8127 C9            RET
                ;
8128 F5         SOUND:PUSH AF
8129 E5            PUSH HL
812A D5            PUSH DE
812B C5            PUSH BC
812C 215981        LD HL,SNDTBL
812F 85            ADD A,L
8130 6F            LD L,A
8131 46            LD B,(HL)
8132 1E1A          LD E,1A
8134 50         SNDS1:LD D,B
8135 3EFF          LD A,FF;SP OUT=H
8137 D398          OUT (98),A
8139 E5         SNDS2:PUSH HL;11clk---------
813A E5            PUSH HL;11clk             |11+11+10+10+4+4+10=60
```

```
813B E1              POP HL;10clk        |60/6=10
813C E1              POP HL;10clk        |
813D 00              NOP;4clk            |10microsec
813E 15           DEC D;4clk             |
813F C23981       JP NZ,SNDS2;10clk---
8142 50           LD D,B
8143 3EDF         LD A,DF;SP OUT=L
8145 D398         OUT (98),A
8147 E5        SNDS3:PUSH HL;11clk---------
8148 E5           PUSH HL;11clk          |11+11+10+10+4+4+10=60
8149 E1              POP HL;10clk        |60/6=10
814A E1              POP HL;10clk        |
814B 00              NOP;4clk            |10microsec
814C 15           DEC D;4clk             |
814D C24781       JP NZ,SNDS3;10clk---
8150 1D           DEC E
8151 C23481       JP NZ,SNDS1
8154 C1           POP BC
8155 D1           POP DE
8156 E1           POP HL
8157 F1           POP AF
8158 C9           RET
               ; SOUND TABLE
8159 7F        SNDTBL:DB 7F;so4
815A 77           DB 77;so#4
815B 71           DB 71;ra4
815C 6A           DB 6A;ra#4
815D 5F           DB 5F;do5
815E 59           DB 59;do#5
815F 54           DB 54;re5
8160 4F           DB 4F;re#
8161 47           DB 47;fA5
8162 43           DB 43;fa#5
8163 3F           DB 3F;so5
8164 3B           DB 3B;so5#
8165 35           DB 35;ra#5
8166 32           DB 32;si5
8167 2F           DB 2F;do6
8168 2C           DB 2C;do#6
8169 25           DB 25;mi6
816A 27           DB 27;re#6
816B 2A           DB 2A;re6
816C 4B           DB 4B;mi5
816D 38           DB 38;ra5
816E 64           DB 64;si4
816F 23           DB 23;fa6
8170 21           DB 21;fa#6
               ;
               ;
ACC            =E802  ADD          =8055  ADDPS        =05A1
ADRG           =FFEE  CK1          =8074  CK2          =8076
CK3            =8082  CK4          =808C  CK5          =8096
CK6            =8098  CNTR         =E805  D1_1         =02DF
DP1            =FFF4  DP3          =FFF6  DTRG         =FFEC
DTST1          =800E  KEY          =802F  KEY1         =8039
KEY2           =803C  KEYIN1       =0616  KEYIN1S      =810C
LED1           =FFF8  LED3         =FFFA  MYDT         =E800
OK             =80C1  OK1          =80C5  OVER         =80AF
```

```
OVER1       =80B3  RND         =80EE  RNDDT       =E810
SNDS1       =8134  SNDS2       =8139  SNDS3       =8147
SNDTBL      =8159  SOUND       =8128  START       =8000
START1      =8026  START2      =802A  TM01S       =8114
TM01S2      =8117  TM1M        =8120  TM1M_2      =8123
```

プログラム№10　すごろく迷路（SUGOROKU．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（E808）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FFFF）です。ここに対応するDTBFのアドレスはE807です。つまり選択したステップ数をHL（E808）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AND　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。こういうとき本当にZ80って便利だなーと感じます。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AND　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－flag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SUB 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。

```
2010/9/17  17:31  SUGOROKC.TXT
END=8175
              ; 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
              ;
                   ORG $8000
                   DTBF=$E800
                   RNDDT=$E810;$E811
                   LED1=$FFF8
                   DP1=$FFF4
                   DP3=$FFF6
                   CNTR=$FFF7;=DP4
              ;
                   DTDPS=$05C0
                   KEYIN1=$0616
                   D1_1=$02DF;about 125microsec
              ;

              ;last position select
8000 CDAA80   START:CALL RND
8003 E607          AND 07
8005 3C            INC A
8006 5F            LD E,A;distance from goal
8007 3E08          LD A,08;goal position
8009 93            SUB E
800A 6F            LD L,A
800B 57            LD D,A;last position save
800C 26E8          LD H,E8;high address of DTBF
800E 73            LD (HL),E
              ;data set
800F 210E08        LD HL,DTBF
8012 0608          LD B,08
8014 7D       DTST2:LD A,L
8015 BA            CP D
8016 CA2E80        JP Z,DTST4;skip last position
8019 CDAA80   DTST3:CALL RND
801C E607          AND 07
801E 3C            INC A
801F 4F            LD C,A;DATA save
8020 85            ADD A,L;next position(right)
8021 FE08          CP 08;goal position
8023 CA1980        JP Z,DTST3
8026 7D            LD A,L
8027 91            SUB C;next position(left)
8028 FEFF          CP FF;dummy position
802A CA1980        JP Z,DTST3
802D 71            LD (HL),C
802E 23       DTST4:INC HL
802F 10E3          DJNZ *DTST2
              ;GAME START
8031 210000        LD HL,$0000
8034 22F6FF        LD (DP3),HL;DECIMAL COUNTER clear
```

```
8037 22F4FF         LD (DP1),HL
             ;key entry(1-8)
803A CDC880         CALL CLR
803D 3E0A           LD A,0A
803F 0610           LD B,10
8041 CD0281    START1:CALL SOUND
8044 10FB           DJNZ START1
8046 CDE680    KEY:CALL KEYIN1S;1-8 input
8049 FE09           CP 09
804B D24680         JP NC,KEY
804E 3D             DEC A
804F FA4680         JP M,KEY
8052 57        NEXT:LD D,A;position save
8053 3AF7FF         LD A,(CNTR);DECIMAL COUNTER up
8056 C601           ADD A,01
8058 27             DAA
8059 32F7FF         LD (CNTR),A
805C CD4B81         CALL DTDP;data disp and save to E
             ;right or left in
805F CDE680    KEY1:CALL KEYIN1S
8062 B7             OR A
8063 CA7F80         JP Z,LEFT
8066 FE15           CP 15;WRITE INC
8068 C25F80         JP NZ,KEY1
             ;step to right
806B 7A             LD A,D;current position(00-07)
806C 83             ADD A,E;E=step No.      D+E=next position
806D FE08           CP 08
806F CA8B80         JP Z,GOAL
8072 DA7980         JP C,RIGHT1;    D+E<08
8075 ED44           NEG; 0-A
8077 E607           AND 07;next position
8079 CDC880    RIGHT1:CALL CLR
807C C35280           JP NEXT
             ;step to left
807F 7A        LEFT:LD A,D;current position(00-07)
8080 93             SUB E;E=step No.
8081 D27980         JP NC,RIGHT1
8084 ED44           NEG
8086 D602           SUB 02;next position
8088 C37980         JP RIGHT1
             ;goal
808B 3E00      GOAL:LD A,00
808D 0610           LD B,10
808F CD0281    GOAL1:CALL SOUND
8092 10FB           DJNZ GOAL1
8094 CDC880         CALL CLR
8097 CDEE80         CALL TM01S
809A CDC005         CALL DTDPS;COUNTER disp
809D CDD380         CALL ALBLNK
80A0 0614           LD B,14;=20 2sec wait
80A2 CDEE80    WAIT:CALL TM01S
80A5 10FB           DJNZ WAIT
80A7 C30080         JP START
             ;
             ;ransu
80AA E5        RND:PUSH HL
80AB D5             PUSH DE
```

```
80AC 2A10E8        LD HL,(RNDDT)
80AF ED5F          LD A,R
80B1 5F            LD E,A
80B2 19            ADD HL,DE
80B3 7C            LD A,H
80B4 E603          AND 03
80B6 67            LD H,A
80B7 7E            LD A,(HL)
80B8 23            INC HL
80B9 86            ADD A,(HL)
80BA 5F            LD E,A
80BB 23            INC HL
80BC 86            ADD A,(HL)
80BD 23            INC HL
80BE 86            ADD A,(HL)
80BF 57            LD D,A
80C0 ED5310E8      LD (RNDDT),DE
80C4 7A            LD A,D
80C5 D1            POP DE
80C6 E1            POP HL
80C7 C9            RET
                ;
                ;LED clear
80C8 21F8FF     CLR:LD HL,LED1
80CB 010008         LD BC,$0800
80CE 71         CLR2:LD (HL),C
80CF 23             INC HL
80D0 10FC           DJNZ *CLR2
80D2 C9             RET
                ;
                ;LED all blink
80D3 060A       ALBLNK:LD B,0A;=10
80D5 3EEF       ALBLNK1:LD A,EF
80D7 D398          OUT (98),A
80D9 CDEE80        CALL TM01S
80DC 3EFF          LD A,FF
80DE D398          OUT (98),A
80E0 CDEE80        CALL TM01S
80E3 10F0          DJNZ *ALBLNK1
80E5 C9            RET
                ;
                ;KEYIN with DE,BC save
80E6 C5         KEYIN1S:PUSH BC
80E7 D5            PUSH DE
80E8 CD1606        CALL KEYIN1
80EB D1            POP DE
80EC C1            POP BC
80ED C9            RET
                ;
                ;
                ;0.1sec timer
80EE D5         TM01S:PUSH DE
80EF 1E64          LD E,64;=100
80F1 CDFA80     TM01S2:CALL TM1M
80F4 1D            DEC E
80F5 C2F180        JP NZ,TM01S2
80F8 D1            POP DE
80F9 C9            RET
```

```
80FA D5       TM1M:PUSH DE
80FB 1608        LD D,08
80FD CDDF02   TM1M_2:CALL D1_1
8100 D1          POP DE
8101 C9          RET
              ;
8102 F5       SOUND:PUSH AF
8103 E5          PUSH HL
8104 D5          PUSH DE
8105 C5          PUSH BC
8106 213381      LD HL,SNDTBL
8109 85          ADD A,L
810A 6F          LD L,A
810B 46          LD B,(HL)
810C 1E1A        LD E,1A
810E 50       SNDS1:LD D,B
810F 3EFF        LD A,FF;SP OUT=H
8111 D398        OUT (98),A
8113 E5       SNDS2:PUSH HL;11clk---------
8114 E5          PUSH HL;11clk            |11+11+10+10+4+4+10=60
8115 E1               POP HL;10clk        |60/6=10
8116 E1               POP HL;10clk        |
8117 00               NOP;4clk            |10microsec
8118 15          DEC D;4clk               |
8119 C21381      JP NZ,SNDS2;10clk---
811C 50          LD D,B
811D 3EDF        LD A,DF;SP OUT=L
811F D398        OUT (98),A
8121 E5       SNDS3:PUSH HL;11clk---------
8122 E5          PUSH HL;11clk            |11+11+10+10+4+4+10=60
8123 E1               POP HL;10clk        |60/6=10
8124 E1               POP HL;10clk        |
8125 00               NOP;4clk            |10microsec
8126 15          DEC D;4clk               |
8127 C22181      JP NZ,SNDS3;10clk---
812A 1D          DEC E
812B C20E81      JP NZ,SNDS1
812E C1          POP BC
812F D1          POP DE
8130 E1          POP HL
8131 F1          POP AF
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:LD A,D
814C 2100E8          LD HL,DTBF
814F 85              ADD A,L
8150 6F              LD L,A
8151 7E              LD A,(HL)
8152 5F              LD E,A;DATA save
8153 216D81          LD HL,SEGDT
8156 4F              LD C,A
8157 0600            LD B,00
8159 09              ADD HL,BC
815A 7E              LD A,(HL)
815B 47              LD B,A;segment data save
815C 21F8FF          LD HL,LED1
815F 7A              LD A,D
8160 85              ADD A,L
8161 6F              LD L,A
8162 70              LD (HL),B
8163 3E10            LD A,10
8165 0603            LD B,03
8167 CD0281  DTDP1:CALL SOUND
816A 10FB            DJNZ *DTDP1
816C C9              RET
816D 5C      SEGDT:DB 5C;0
816E 06              DB 06;1
816F 5B              DB 5B;2
8170 4F              DB 4F;3
8171 66              DB 66;4
8172 6D              DB 6D;5
8173 7D              DB 7D;6
8174 27              DB 27;7
8175 7F              DB 7F;8
             ;
ALBLNK       =80D3  ALBLNK1      =80D5  CLR          =80C8
CLR2         =80CE  CNTR         =FFF7  D1_1         =02DF
DP1          =FFF4  DP3          =FFF6  DTBF         =E800
DTDP         =814B  DTDP1        =8167  DTDPS        =05C0
DTST2        =8014  DTST3        =8019  DTST4        =802E
GOAL         =808B  GOAL1        =808F  KEY          =8046
KEY1         =805F  KEYIN1       =0616  KEYIN1S      =80E6
LED1         =FFF8  LEFT         =807F  NEXT         =8052
RIGHT1       =8079  RND          =80AA  RNDDT        =E810
SEGDT        =816D  SNDS1        =810E  SNDS2        =8113
SNDS3        =8121  SNDTBL       =8133  SOUND        =8102
START        =8000  START1       =8041  TM01S        =80EE
TM01S2       =80F1  TM1M         =80FA  TM1M_2       =80FD
WAIT         =80A2
```

プログラム№11　REVERSE（REVERSE. 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（FFF4～FFF7）を使うか、アドレスレジスタ及びデータレジスタ（FFEC～FFEF）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。Z80にはあるにはあるのですが、この場合には面倒な処理であることに変わりはありません。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、XOR　AはLD　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。LD　A，（HL）として、OR　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 AFとするとか、LD　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DEC　Cとしてしまいます。メモリの内容が00ならDEC　Cの結果はFFになって、前のプログラムでててきたようにマイナスになるのでS－flagが立ちます。でJP　M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－flag）も立つので、よくJP　Cを使いますが、DECとINCについてはC－flag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DECしたものをもとに戻すためINCしたあとSRLを実行します。SRLも8080には無い命令です。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(CP (HL)はあるがCP (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。

```
2010/9/17  14:33  reverseb.txt
END=813A
              ; REVERSE for NDZ
              ; 03/04/29 4/30 5/1 5/2 5/5 5/13 5/15
              ;10/6/3 for ND80Z3
              ;
                   ORG $8000
                   DTBF=$E800
                   BLNKBF=$E808
                   RNDDT=$E810;$E811
                   CNTR=$E812
                   LED1=$FFF8
                   DP1=$FFF4
                   DP3=$FFF6
              ;
                   DTDPS=$05C0
                   KEYIN1=$0616
                   KEYIN2=$0623
                 D1_1=$02DF;about 125microsec
              ;
              ;data set
8000 0608     START:LD B,08
8002 2100E8          LD HL,DTBF
8005 AF              XOR A
8006 77       DTST0:LD (HL),A
8007 23              INC HL
8008 10FC            DJNZ *DTST0
800A CD9880   DTST1:CALL RND
800D E607            AND 07
800F 3C              INC A
8010 0608            LD B,08
8012 2100E8          LD HL,DTBF
8015 4E       DTST2:LD C,(HL)
8016 0D              DEC C
8017 F22080          JP P,DTST3
```

```
801A 77          LD (HL),A
801B 10ED        DJNZ *DTST1
801D C32780      JP ENTRY
8020 BE      DTST3:CP (HL)
8021 CA0A80      JP Z,DTST1
8024 23          INC HL
8025 10EE        DJNZ *DTST2
8027 CDFF80  ENTRY:CALL DTBFDP
802A AF          XOR A
802B 3212E8      LD (CNTR),A;DECIMAL COUNTER clear
             ;key entry(1-8)
802E 0E01    REENT:LD C,01
8030 CD1781  KEY:CALL KEYIN1S
8033 FE15    KEY1:CP 15;WRITE INC
8035 CA4980      JP Z,REV
8038 FE09        CP 09
803A D23080      JP NC,KEY
803D FE02        CP 02
803F DA3080      JP C,KEY
             ;blink select data
8042 4F          LD C,A
8043 CDC980      CALL BLNKC
8046 C33380      JP KEY1
             ;reverse
8049 0D      REV:DEC C
804A CA2E80      JP Z,REENT
804D 210DE8      LD HL,DTBF
8050 54          LD D,H
8051 79          LD A,C
8052 85          ADD A,L
8053 5F          LD E,A
8054 0C          INC C
8055 CB39        SRL C
8057 1A      REV1:LD A,(DE)
8058 46          LD B,(HL)
8059 77          LD (HL),A
805A 78          LD A,B
805B 12          LD (DE),A
805C 23          INC HL
805D 1B          DEC DE
805E 0D          DEC C
805F C25780      JP NZ,REV1
8062 CDFF80  REV2:CALL DTBFDP
8065 3A12E8      LD A,(CNTR);DECIMAL COUNTER up
8068 C601        ADD A,01
806A 27          DAA
806B 3212E8      LD (CNTR),A
             ;compare DATA
806E 210DE8      LD HL,DTBF
8071 0607        LD B,07
8073 7E      COMP1:LD A,(HL)
8074 23          INC HL
8075 BE          CP (HL)
8076 D22E80      JP NC,REENT;not yet
8079 10F8        DJNZ *COMP1
             ;OK! ALL LED blink
807B CDB680      CALL ALBLNK
             ;COUNTER disp
```

```
807E 210000        LD HL,$0000
8081 22F4FF        LD (DP1),HL
8084 3A12E8        LD A,(CNTR)
8087 67            LD H,A
8088 22F6FF        LD (DP3),HL
808B CDC005        CALL DTDPS
808E 0614          LD B,14;=20 ,2sec wait
8090 CD2781   WAIT:CALL TM01S
8093 10FB          DJNZ WAIT
8095 C30080        JP START
              ;
              ;ransu
8098 E5       RND:PUSH HL
8099 D5            PUSH DE
809A 2A10E8        LD HL,(RNDDT)
809D ED5F          LD A,R
809F 5F            LD E,A
80A0 19            ADD HL,DE
80A1 7C            LD A,H
80A2 E603          AND 03
80A4 67            LD H,A
80A5 7E            LD A,(HL)
80A6 23            INC HL
80A7 86            ADD A,(HL)
80A8 5F            LD E,A
80A9 23            INC HL
80AA 86            ADD A,(HL)
80AB 23            INC HL
80AC 86            ADD A,(HL)
80AD 57            LD D,A
80AE ED5310E8      LD (RNDDT),DE
80B2 7A            LD A,D
80B3 D1            POP DE
80B4 E1            POP HL
80B5 C9            RET
              ;
              ;LED all blink
80B6 060A     ALBLNK:LD B,0A;=10
80B8 3EEF     ALBLNK1:LD A,EF
80BA D398        OUT (98),A
80BC CD2781        CALL TM01S
80BF 3EFF               LD A,FF
80C1 D398        OUT (98),A
80C3 CD2781        CALL TM01S
80C6 10F0          DJNZ *ALBLNK1
80C8 C9            RET
              ;LED blink
80C9 79       BLNKC:LD A,C;save C
80CA 21F8FF        LD HL,LED1
80CD 1108E8        LD DE,BLNKBF
80D0 0600          LD B,00
80D2 EDB0          LDIR
80D4 4F            LD C,A
80D5 21F8FF   BLNKC1:LD HL,LED1
80D8 1108E8        LD DE,BLNKBF
80DB 41            LD B,C
80DC 1A       BLNKC2:LD A,(DE)
80DD BE            CP (HL)
```

```
80DE C2E280        JP NZ,BLNKC3
80E1 AF            XOR A
80E2 77       BLNKC3:LD (HL),A
80E3 23            INC HL
80E4 13            INC DE
80E5 10F5          DJNZ *BLNKC2
80E7 CD2781        CALL TM01S
80EA CD1F81        CALL KEYIN2S
80ED 3C            INC A
80EE CAD580        JP Z,BLNKC1;no key in
80F1 3D            DEC A;key data
80F2 2108E8        LD HL,BLNKBF
80F5 11F8FF        LD DE,LED1
80F8 0600          LD B,00
80FA C5            PUSH BC
80FB EDB0          LDIR
80FD C1            POP BC
80FE C9            RET
                   ;
                   ;DTBF data disp
80FF 2100E8   DTBFDP:LD HL,DTBF
8102 11F4FF        LD DE,DP1
8105 0604          LD B,04
8107 7E       DTBFDP1:LD A,(HL)
8108 07            RLCA
8109 07            RLCA
810A 07            RLCA
810B 07            RLCA
810C 23            INC HL
810D B6            OR (HL)
810E 12            LD (DE),A
810F 23            INC HL
8110 13            INC DE
8111 10F4          DJNZ *DTBFDP1
8113 CDC005        CALL DTDPS
8116 C9            RET
                   ;
                   ;KEYIN with DE,BC save
8117 C5       KEYIN1S:PUSH BC
8118 D5          PUSH DE
8119 CD1606      CALL KEYIN1
811C D1          POP DE
811D C1          POP BC
811E C9          RET
811F C5       KEYIN2S:PUSH BC
8120 D5          PUSH DE
8121 CD2306      CALL KEYIN2
8124 D1          POP DE
8125 C1          POP BC
8126 C9          RET
                   ;
                   ;0.1sec timer
8127 D5       TM01S:PUSH DE
8128 1E64          LD E,64;=100
812A CD3381   TM01S2:CALL TM1M
812D 1D            DEC E
812E C22A81        JP NZ,TM01S2
8131 D1            POP DE
```

```
8132 C9            RET
8133 D5       TM1M:PUSH DE
8134 1608          LD D,08
8136 CDDF02   TM1M_2:CALL D1_1
8139 D1            POP DE
813A C9            RET
                   ;
ALBLNK       =80B6  ALBLNK1      =80B8  BLNKBF       =E808
BLNKC        =80C9  BLNKC1       =80D5  BLNKC2       =80DC
BLNKC3       =80E2  CNTR         =E812  COMP1        =8073
D1_1         =02DF  DP1          =FFF4  DP3          =FFF6
DTBF         =E800  DTBFDP       =80FF  DTBFDP1      =8107
DTDPS        =05C0  DTST0        =8006  DTST1        =800A
DTST2        =8015  DTST3        =8020  ENTRY        =8027
KEY          =8030  KEY1         =8033  KEYIN1       =0616
KEYIN1S      =8117  KEYIN2       =0623  KEYIN2S      =811F
LED1         =FFF8  REENT        =802E  REV          =8049
REV1         =8057  REV2         =8062  RND          =8098
RNDDT        =E810  START        =8000  TM01S        =8127
TM01S2       =812A  TM1M         =8133  TM1M_2       =8136
WAIT         =8090
```

プログラム№12　ROCK　CLIMBING（ROCKCLIM. 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。ホント、こんなときZ80は助かります。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。IX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。IY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、IX、IY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。AND　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。DEC　CでS－flagがたてばREAD　INCなのでサイコロを振る処理に行きます（C＝00のときDEC　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（IX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。IXレジスタ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IYレジスタ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IXレジスタに入っている現在の高さを格納するメモリアドレスにIYレジスタ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=8135
             ; LOCK CLIMBING for NDZ
             ; 03/05/08 5/14
             ;10/6/3 for ND80Z3
             ;7/22 9/22
             ;
                  ORG $8000
                  ASI1=$E800
                  ASI2=$E801
                  BLNKBF=$E808
                  RNDDT=$E810;$E811
                  PLYR2=$FFEC;=DTRG
                  PLYR1=$FFEE;=ADRG
                  LED1=$FFF8
                  LED5=$FFFC
             ;
                  ADDPS=$05A1
                  KEYIN1=$0616
                  KEYIN2=$0623
                D1_1=$02DF;about 125microsec
             ;
8000 210000  START:LD HL,$0000
8003 22EEFF       LD (PLYR1),HL
8006 22ECFF       LD (PLYR2),HL
8009 2200E8       LD (ASI1),HL
             ;player1
800C 21F8FF  REENT1:LD HL,LED1
800F DD21EEFF     LD IX,PLYR1
8013 FD2100E8     LD IY,ASI1
8017 CDBA80  REENT2:CALL DTDP
801A CDDA80       CALL BLNK
             ;"UP" or "STAY" disp
801D E603    USDISP:AND 03
801F 4F           LD C,A;00 or 03 save
8020 3E01         LD A,01;"UP"
8022 CA2780       JP Z,USDISP1
8025 3E08         LD A,08;"STAY"
8027 77      USDISP1:LD (HL),A
8028 CD1281  KEY:CALL KEYIN1S
802B FE15         CP 15;write inc
802D CA3D80       JP Z,SAI
8030 FE13         CP 13;read dec("STAY")
8032 CA1D80       JP Z,USDISP
8035 FE14         CP 14;read inc("UP")
8037 C22880       JP NZ,KEY
803A C31D80       JP USDISP
             ;saikoro
803D 0D      SAI:DEC C
803E F26580       JP P,ASISET;C=03(keyin=RD_DEC)
8041 010002       LD BC,$0200
8044 CD9C80  SAI1:CALL RND
8047 E607         AND 07
8049 FE06         CP 06
804B D24480       JP NC,SAI1
804E B7           OR A
804F CA7F80       JP Z,DOWN
8052 3C           INC A
8053 81           ADD A,C
```

```
8054 27          DAA
8055 4F          LD C,A
8056 10EC        DJNZ SAI1
8058 DD8600      ADD A,(IX+00)
805B 27          DAA
805C DD7700      LD (IX+00),A
805F DA8880      JP C,WIN
8062 C31780      JP REENT2
             ;asiba set
8065 DD7E00  ASISET:LD A,(IX+00)
8068 FD7700      LD (IY+00),A
             ;change player
806B 7D      ASISET1:LD A,L
806C E607        AND 07
806E C20C80      JP NZ,REENT1;current HL=LED5
8071 21FCFF      LD HL,LED5
8074 DD21ECFF    LD IX,PLYR2
8078 FD2101E8    LD IY,ASI2
807C C31780      JP REENT2
807F FD7E00  DOWN:LD A,(IY+00)
8082 DD7700      LD (IX+00),A
8085 C36B80      JP ASISET1
8088 DD360101 WIN:LD (IX+01),01
808C CDBA80      CALL DTDP
808F CDC780      CALL ALBLNK
8092 0614        LD B,14;=20 2sec wait
8094 CD2281  WAIT:CALL TM01S
8097 10FB        DJNZ WAIT
8099 C30080      JP START
             ;
             ;ransu
809C E5      RND:PUSH HL
809D D5          PUSH DE
809E 2A10E8      LD HL,(RNDDT)
80A1 ED5F        LD A,R
80A3 5F          LD E,A
80A4 19          ADD HL,DE
80A5 7C          LD A,H
80A6 E603        AND 03
80A8 67          LD H,A
80A9 7E          LD A,(HL)
80AA 23          INC HL
80AB 86          ADD A,(HL)
80AC 5F          LD E,A
80AD 23          INC HL
80AE 86          ADD A,(HL)
80AF 23          INC HL
80B0 86          ADD A,(HL)
80B1 57          LD D,A
80B2 ED5310E8    LD (RNDDT),DE
80B6 7A          LD A,D
80B7 D1          POP DE
80B8 E1          POP HL
80B9 C9          RET
             ;
             ;data disp
80BA E5      DTDP:PUSH HL
80BB CDA105      CALL ADDPS
```

```
80BE E1           POP HL
80BF AF           XOR A
80C0 32F8FF       LD (LED1),A
80C3 32FCFF       LD (LED5),A
80C6 C9           RET
                  ;
                  ;LED all blink
80C7 060A         ALBLNK:LD B,0A;=10
80C9 3EEF         ALBLNK1:LD A,EF
80CB D398           OUT (98),A
80CD CD2281         CALL TM01S
80D0 3EFF           LD A,FF
80D2 D398           OUT (98),A
80D4 CD2281         CALL TM01S
80D7 10F0           DJNZ *ALBLNK1
80D9 C9             RET
                  ;LED blink and keyin HL=LED1 or LED5
80DA E5           BLNK:PUSH HL;HL save
80DB 23             INC HL
80DC E5             PUSH HL
80DD 1108E8         LD DE,BLNKBF
80E0 010300         LD BC,$0003
80E3 EDB0           LDIR
80E5 E1           BLNK1:POP HL
80E6 E5             PUSH HL
80E7 1108E8         LD DE,BLNKBF
80EA 0603           LD B,03
80EC 1A           BLNK2:LD A,(DE)
80ED BE             CP (HL)
80EE C2F280         JP NZ,BLNK3
80F1 AF             XOR A
80F2 77           BLNK3:LD (HL),A
80F3 23             INC HL
80F4 13             INC DE
80F5 10F5           DJNZ *BLNK2
80F7 CD2281         CALL TM01S
80FA CD1A81         CALL KEYIN2S
80FD FE13           CP 13;read dec
80FF CA0781         JP Z,BLNK4
8102 FE14           CP 14;read inc
8104 C2E580         JP NZ,BLNK1
8107 D1           BLNK4:POP DE
8108 2108E8         LD HL,BLNKBF
810B 010300         LD BC,$0003
810E EDB0           LDIR
8110 E1             POP HL
8111 C9             RET
                  ;
                  ;KEYIN with DE,BC save
8112 C5           KEYIN1S:PUSH BC
8113 D5             PUSH DE
8114 CD1606         CALL KEYIN1
8117 D1             POP DE
8118 C1             POP BC
8119 C9             RET
811A C5           KEYIN2S:PUSH BC
811B D5             PUSH DE
811C CD2306         CALL KEYIN2
```

```
811F D1          POP DE
8120 C1          POP BC
8121 C9          RET
                 ;
                 ;
                 ;0.1sec timer
8122 D5          TM01S:PUSH DE
8123 1E64             LD E,64;=100
8125 CD2E81      TM01S2:CALL TM1M
8128 1D               DEC E
8129 C22581           JP NZ,TM01S2
812C D1               POP DE
812D C9               RET
812E D5          TM1M:PUSH DE
812F 1608           LD D,08
8131 CDDF02      TM1M_2:CALL D1_1
8134 D1             POP DE
8135 C9             RET
                 ;
ADDPS        =05A1  ALBLNK       =80C7  ALBLNK1      =80C9
ASI1         =E800  ASI2         =E801  ASISET       =8065
ASISET1      =806B  BLNK         =80DA  BLNK1        =80E5
BLNK2        =80EC  BLNK3        =80F2  BLNK4        =8107
BLNKBF       =E808  D1_1         =02DF  DOWN         =807F
DTDP         =80BA  IX           =1E8B  IY           =1C72
KEY          =8028  KEYIN1       =0616  KEYIN1S      =8112
KEYIN2       =0623  KEYIN2S      =811A  LED1         =FFF8
LED5         =FFFC  PLYR1        =FFEE  PLYR2        =FFEC
REENT1       =800C  REENT2       =8017  RND          =809C
RNDDT        =E810  SAI          =803D  SAI1         =8044
START        =8000  TM01S        =8122  TM01S2       =8125
TM1M         =812E  TM1M_2       =8131  USDISP       =801D
USDISP1      =8027  WAIT         =8094  WIN          =8088
```

プログラムNo.13 WILD SEVEN(WILD7.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。

```
2010/9/18  9:12  WILD7C.TXT
END=8159
                ; 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
                ;
                      ORG $8000
                      RNDDT=$E810;$E811
                      CNTR=$E812
                      LED1=$FFF8
                      DP1=$FFF4
                      DP3=$FFF6
                ;
                      DTDPS=$05C0
                      KEYIN1=$0616
                      KEYIN2=$0623
                            D1_1=$02DF;about 125microsec
                ;
                ;data buffer clear
8000 3EFF       START:LD A,FF
8002 0608             LD B,08
8004 21F8FF           LD HL,LED1
8007 77         DTCLR:LD (HL),A
8008 23               INC HL
8009 10FC             DJNZ *DTCLR
                ;space position select
800B CDFD80           CALL RND
800E E607             AND 07
8010 21F8FF           LD HL,LED1
8013 85               ADD A,L
8014 6F               LD L,A
8015 3600             LD (HL),00
                ;data set
8017 0607          LD B,07
```

```
8019 CDFD80   DTST1:CALL RND
801C E607          AND 07
801E CA1980      JP Z,DTST1
8021 DD215281      LD IX,SEGDT
8025 1600          LD D,00
8027 5F            LD E,A
8028 DD19          ADD IX,DE
802A DD7E00        LD A,(IX+00)
802D 21F8FF        LD HL,LED1
8030 4E       DTST2:LD C,(HL)
8031 0C            INC C
8032 C23B80        JP NZ,DTST3
8035 77            LD (HL),A
8036 10E1          DJNZ *DTST1
8038 C34680        JP ENTRY
803B BE       DTST3:CP (HL)
803C CA1980        JP Z,DTST1
803F 2C            INC L
8040 C23080        JP NZ,DTST2
8043 C31980      JP DTST1
             ;GAME START
8046 AF       ENTRY:XOR A
8047 3212E8        LD (CNTR),A;DECIMAL COUNTER clear
             ;key entry(1-8)
804A CD2E81   KEY:CALL KEYIN1S
804D FE09          CP 09
804F D24A80        JP NC,KEY
8052 3D            DEC A
8053 FA4A80        JP M,KEY
             ;LED blink and wait WINC in
8056 21F8FF   BLNK0:LD HL,LED1
8059 85            ADD A,L
805A 6F            LD L,A
805B 7E            LD A,(HL)
805C B7            OR A
805D CA4A80        JP Z,KEY;space position
8060 57            LD D,A;LED data save
8061 47            LD B,A
8062 0E00          LD C,00
8064 71       BLNK1:LD (HL),C
8065 CD3E81        CALL TM01S
8068 CD3681        CALL KEYIN2S
806B FE09          CP 09
806D D27880        JP NC,BLNK2
8070 3D            DEC A
8071 FA7D80        JP M,BLNK3
8074 72            LD (HL),D
8075 C35680        JP BLNK0
8078 FE15     BLNK2:CP 15;WRITE INC
807A CA8380        JP Z,CHECK
807D 79       BLNK3:LD A,C
807E 48            LD C,B
807F 47            LD B,A
8080 C36480        JP BLNK1
             ;check
8083 E5       CHECK:PUSH HL;save keyin position
8084 7D            LD A,L
8085 E607          AND 07
```

```
8087 CA9C80        JP Z,CHECKR;position=LED1
808A 2B            DEC HL
808B 7E            LD A,(HL)
808C B7            OR A
808D CABA80        JP Z,REP;left=space
8090 7D            LD A,L
8091 E607          AND 07
8093 CA9C80        JP Z,CHECKR;left=LED1
8096 2B            DEC HL
8097 7E            LD A,(HL)
8098 B7            OR A
8099 CABA80        JP Z,REP;left of left=space
809C E1       CHECKR:POP HL
809D E5            PUSH HL
809E 7D            LD A,L
809F E607          AND 07
80A1 FE07          CP 07
80A3 CAF980        JP Z,CANT;position=LED8,cannot replace
80A6 23            INC HL
80A7 7E            LD A,(HL)
80A8 B7            OR A
80A9 CABA80        JP Z,REP;right=space
80AC 7D            LD A,L
80AD E607          AND 07
80AF FE07          CP 07
80B1 CAF980        JP Z,CANT;right=LED8,cannot replace
80B4 23            INC HL
80B5 7E            LD A,(HL)
80B6 B7            OR A
80B7 C2F980        JP NZ,CANT;right of right is not space,cannot replace
              ;replace
80BA 72       REP:LD (HL),D
80BB E1            POP HL
80BC 3600          LD (HL),00
80BE 3A12E8        LD A,(CNTR);DECIMAL COUNTER up
80C1 C601          ADD A,01
80C3 27            DAA
80C4 3212E8        LD (CNTR),A
              ;all check
80C7 21F8FF        LD HL,LED1
80CA 0607          LD B,07
80CC DD215381      LD IX,SEGDT1
80D0 DD7E00   ALCHK1:LD A,(IX+00)
80D3 BE            CP (HL)
80D4 C24A80        JP NZ,KEY;not done
80D7 23            INC HL
80D8 DD23          INC IX
80DA 10F4          DJNZ *ALCHK1
              ;OK! ALL LED blink
80DC CD1B81        CALL ALBLNK
              ;COUNTER disp
80DF 210000        LD HL,$0000
80E2 22F4FF        LD (DP1),HL
80E5 3A12E8        LD A,(CNTR)
80E8 67            LD H,A
80E9 22F6FF        LD (DP3),HL
80EC CDC005        CALL DTDPS
80EF 0614          LD B,14;=20 ,2sec wait
```

```
80F1 CD3E81   WAIT:CALL TM01S
80F4 10FB          DJNZ WAIT
80F6 C30080        JP START
              ;cannot replace
80F9 E1       CANT:POP HL
80FA C37D80        JP BLNK3
              ;
              ;ransu
80FD E5       RND:PUSH HL
80FE D5            PUSH DE
80FF 2A10E8        LD HL,(RNDDT)
8102 ED5F          LD A,R
8104 5F            LD E,A
8105 19            ADD HL,DE
8106 7C            LD A,H
8107 E603          AND 03
8109 67            LD H,A
810A 7E            LD A,(HL)
810B 23            INC HL
810C 86            ADD A,(HL)
810D 5F            LD E,A
810E 23            INC HL
810F 86            ADD A,(HL)
8110 23            INC HL
8111 86            ADD A,(HL)
8112 57            LD D,A
8113 ED5310E8      LD (RNDDT),DE
8117 7A            LD A,D
8118 D1            POP DE
8119 E1            POP HL
811A C9            RET
              ;
              ;LED all blink
811B 060A     ALBLNK:LD B,0A;=10
811D 3EEF     ALBLNK1:LD A,EF
811F D398        OUT (98),A
8121 CD3E81        CALL TM01S
8124 3EFF          LD A,FF
8126 D398        OUT (98),A
8128 CD3E81        CALL TM01S
812B 10F0          DJNZ *ALBLNK1
812D C9            RET
              ;
              ;KEYIN with DE,BC save
812E C5       KEYIN1S:PUSH BC
812F D5          PUSH DE
8130 CD1606      CALL KEYIN1
8133 D1          POP DE
8134 C1          POP BC
8135 C9          RET
8136 C5       KEYIN2S:PUSH BC
8137 D5          PUSH DE
8138 CD2306      CALL KEYIN2
813B D1          POP DE
813C C1          POP BC
813D C9          RET
              ;
              ;0.1sec timer
```

```
813E D5       TM01S:PUSH DE
813F 1E64          LD E,64;=100
8141 CD4A81   TM01S2:CALL TM1M
8144 1D            DEC E
8145 C24181        JP NZ,TM01S2
8148 D1            POP DE
8149 C9            RET
814A D5       TM1M:PUSH DE
814B 1608        LD D,08
814D CDDF02   TM1M_2:CALL D1_1
8150 D1          POP DE
8151 C9          RET
              ;
8152 00       SEGDT:DB 00;space
8153 06       SEGDT1:DB 06;1
8154 5B            DB 5B;2
8155 4F            DB 4F;3
8156 66            DB 66;4
8157 6D            DB 6D;5
8158 7D            DB 7D;6
8159 27            DB 27;7
              ;
ALBLNK       =811B  ALBLNK1      =811D  ALCHK1       =80D0
BLNK0        =8056  BLNK1        =8064  BLNK2        =8078
BLNK3        =807D  CANT         =80F9  CHECK        =8083
CHECKR       =809C  CNTR         =E812  D1_1         =02DF
DP1          =FFF4  DP3          =FFF6  DTCLR        =8007
DTDPS        =05C0  DTST1        =8019  DTST2        =8030
DTST3        =803B  ENTRY        =8046  IX           =1689
KEY          =804A  KEYIN1       =0616  KEYIN1S      =812E
KEYIN2       =0623  KEYIN2S      =8136  LED1         =FFF8
REP          =80BA  RND          =80FD  RNDDT        =E810
SEGDT        =8152  SEGDT1       =8153  START        =8000
TM01S        =813E  TM01S2       =8141  TM1M         =814A
TM1M_2       =814D  WAIT         =80F1
```

プログラム№14　MOO（MOO. 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がーーーー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。ちょっと以外な気がしますが、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。なおDTST4:の前のDJNZにはコメントがついているように、ここでB＝00になることはありません。ここのDJNZはDCR　B、JP　DTST3の代わりとして使用しています。

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－flug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。

```
2010/9/17  17:58  MOOB.TXT
END=81BD
                ; MOO for NDZ
                ; 03/05/06 5/15
                ;10/6/4 for ND80Z3
                ;9/17
                ;
                     ORG $8000
                     DTBF=$E800
                     DTBF2=$E802
                     RNDDT=$E810;$E811
                     LED1=$FFF8
                     LED3=$FFFA
```

```
                    LED5=$FFFC
                    LED8=$FFFF
                    DP1=$FFF4
                    DP3=$FFF6
                    CNTR=$FFF7;=DP4
               ;
                    DTDPS=$05C0
                    KEYIN1=$0616
                          D1_1=$02DF;about 125microsec
               ;
               ;data set
8000 21FFFF    START:LD HL,$FFFF
8003 2200E8         LD (DTBF),HL
8006 2202E8         LD (DTBF2),HL
               ;data set
8009 2E00           LD L,00
800B CDFF80    DTST1:CALL RND
800E E60F           AND 0F
8010 FE0A           CP 0A
8012 D20B80         JP NC,DTST1
8015 B7             OR A
8016 C21D80         JP NZ,DTST2
8019 BD             CP L
801A CA0B80         JP Z,DTST1;first No. is not "0"
801D CD1D81    DTST2:CALL CNVT
8020 2100E8         LD HL,DTBF
8023 0604           LD B,04
8025 BE        DTST3:CP (HL)
8026 CA0B80         JP Z,DTST1;already selected No.
8029 4E             LD C,(HL)
802A 0C             INC C
802B CA3180         JP Z,DTST4;(HL)=FF
802E 23             INC HL
802F 10F4           DJNZ *DTST3;B is never 0
8031 77        DTST4:LD (HL),A
8032 2C             INC L;for first No. check
8033 10D6           DJNZ *DTST1
               ;GAME START
8035 210000         LD HL,$0000
8038 22F6FF         LD (DP3),HL;DECIMAL COUNTER clear
803B 22F4FF         LD (DP1),HL
               ;key entry
803E CD3981         CALL STDP
8041 3E00           LD A,00;buuuu
8043 0605           LD B,05
8045 CD7381    START1:CALL SOUND
8048 10FB           DJNZ START1
804A CD5781    KEY:CALL KEYIN1S
804D FE15           CP 15;WRITE INC
804F CA6E80         JP Z,COMP
8052 FE0F           CP 0F;"F"
8054 CAD680         JP Z,GIVUP
8057 D24A80         JP NC,KEY
805A 11FCFF         LD DE,LED5
805D 62             LD H,D
805E 6B             LD L,E
805F 23             INC HL
8060 010300         LD BC,$0003
```

```
8063 EDB0          LDIR
8065 CD1D81        CALL CNVT
8068 32FFFF        LD (LED8),A
806B C34A80        JP KEY
              ;DATA compere
              ;BULL check
806E 3AF7FF   COMP:LD A,(CNTR)
8071 C601          ADD A,01;DECIMAL COUNTER up
8073 27            DAA
8074 32F7FF        LD (CNTR),A
8077 210000        LD HL,$0000
807A 22F8FF        LD (LED1),HL
807D 22FAFF        LD (LED3),HL
8080 2100E8        LD HL,DTBF
8083 11FCFF        LD DE,LED5
8086 DD21F8FF      LD IX,LED1
808A 010404        LD BC,$0404
808D 1A       COMP1:LD A,(DE)
808E BE            CP (HL)
808F C29F80        JP NZ,COMP2
8092 DD36007C      LD (IX+00),7C;b
8096 0D            DEC C
8097 CAE680        JP Z,OK
809A F680          OR 80
809C 77            LD (HL),A;B "bull" mark set
809D DD23          INC IX
809F 23       COMP2:INC HL
80A0 13            INC DE
80A1 10EA          DJNZ *COMP1
              ;COW check
80A3 1100E8        LD DE,DTBF
80A6 0604          LD B,04
80A8 0E04     COMP3:LD C,04
80AA 21FCFF        LD HL,LED5
80AD 1A            LD A,(DE)
80AE B7            OR A
80AF F2B880        JP P,COMP4
80B2 E67F          AND 7F
80B4 12            LD (DE),A;"bull" mark clear
80B5 C3C780        JP COMP6
80B8 BE       COMP4:CP (HL)
80B9 C2C280        JP NZ,COMP5
80BC DD360039      LD (IX+00),39;C
80C0 DD23          INC IX
80C2 23       COMP5:INC HL
80C3 0D            DEC C
80C4 C2B880        JP NZ,COMP4
80C7 13       COMP6:INC DE
80C8 10DE          DJNZ *COMP3
80CA 3E0A          LD A,0A
80CC 0605          LD B,05
80CE CD7381   COMP8:CALL SOUND
80D1 10FB          DJNZ *COMP8
80D3 C34A80        JP KEY
              ;give up
80D6 2100E8   GIVUP:LD HL,DTBF
80D9 11F8FF        LD DE,LED1
80DC 010400        LD BC,$0004
```

```
80DF EDB0          LDIR
80E1 0628          LD B,28;=40 ,4sec wait
80E3 C3F780        JP WAIT
80E6 3E17     OK:LD A,17
80E8 060A          LD B,0A
80EA CD7381   OK1:CALL SOUND
80ED 10FB          DJNZ OK1
80EF CD2681        CALL ALBLNK
80F2 CDC005        CALL DTDPS;COUNTER disp
80F5 0614          LD B,14;=20 ,2sec wait
80F7 CD5F81   WAIT:CALL TM01S
80FA 10FB          DJNZ WAIT
80FC C30080        JP START
              ;
              ;ransu
80FF E5       RND:PUSH HL
8100 D5            PUSH DE
8101 2A10E8        LD HL,(RNDDT)
8104 ED5F          LD A,R
8106 5F            LD E,A
8107 19            ADD HL,DE
8108 7C            LD A,H
8109 E603          AND 03
810B 67            LD H,A
810C 7E            LD A,(HL)
810D 23            INC HL
810E 86            ADD A,(HL)
810F 5F            LD E,A
8110 23            INC HL
8111 86            ADD A,(HL)
8112 23            INC HL
8113 86            ADD A,(HL)
8114 57            LD D,A
8115 ED5310E8      LD (RNDDT),DE
8119 7A            LD A,D
811A D1            POP DE
811B E1            POP HL
811C C9            RET
              ;
              ;from key data to segment data convert
811D 214D81   CNVT:LD HL,SEGDT
8120 5F            LD E,A
8121 1600          LD D,00
8123 19            ADD HL,DE
8124 7E            LD A,(HL)
8125 C9            RET
              ;
              ;LED all blink
8126 060A     ALBLNK:LD B,0A;=10
8128 3EEF     ALBLNK1:LD A,EF
812A D398        OUT (98),A
812C CD5F81        CALL TM01S
812F 3EFF          LD A,FF
8131 D398        OUT (98),A
8133 CD5F81        CALL TM01S
8136 10F0          DJNZ *ALBLNK1
8138 C9            RET
              ;
```

```
                ;start data disp
8139 214581     STDP:LD HL,STDT
813C 11F8FF          LD DE,LED1
813F 010800          LD BC,$0008
8142 EDB0            LDIR
8144 C9              RET
8145 40         STDT:DB 40
8146 40              DB 40
8147 40              DB 40
8148 40              DB 40
8149 5C              DB 5C
814A 5C              DB 5C
814B 5C              DB 5C
814C 5C              DB 5C
                ;
814D 5C         SEGDT:DB 5C;0
814E 06              DB 06;1
814F 5B              DB 5B;2
8150 4F              DB 4F;3
8151 66              DB 66;4
8152 6D              DB 6D;5
8153 7D              DB 7D;6
8154 27              DB 27;7
8155 7F              DB 7F;8
8156 6F              DB 6F;9
                ;
                ;KEYIN with DE,BC save
8157 C5         KEYIN1S:PUSH BC
8158 D5            PUSH DE
8159 CD1606        CALL KEYIN1
815C D1            POP DE
815D C1            POP BC
815E C9            RET
                ;
                ;0.1sec timer
815F D5         TM01S:PUSH DE
8160 1E64            LD E,64;=100
8162 CD6B81     TM01S2:CALL TM1M
8165 1D              DEC E
8166 C26281          JP NZ,TM01S2
8169 D1              POP DE
816A C9              RET
816B D5         TM1M:PUSH DE
816C 1608          LD D,08
816E CDDF02     TM1M_2:CALL D1_1
8171 D1            POP DE
8172 C9            RET
                ;
8173 F5         SOUND:PUSH AF
8174 E5            PUSH HL
8175 D5            PUSH DE
8176 C5            PUSH BC
8177 21A681        LD HL,SNDTBL
817A 85            ADD A,L
817B 6F            LD L,A
817C 46            LD B,(HL)
817D 1E1A          LD E,1A
817F 50         SNDS1:LD D,B
```

```
8180 3EFF        LD A,FF;SP OUT=H
8182 D398        OUT (98),A
8184 2A0000   SNDS2:LD HL,($0000);DUMMY:16clk
8187 210000      LD HL,$0000;DUMMY:10clk
818A 15          DEC D;4clk
818B C28481      JP NZ,SNDS2;10clk
818E 50          LD D,B
818F 3EDF        LD A,DF;SP OUT=L
8191 D398        OUT (98),A
8193 2A0000   SNDS3:LD HL,($0000);DUMMY:16clk
8196 210000      LD HL,$0000;DUMMY:10clk
8199 15          DEC D;4clk
819A C29381      JP NZ,SNDS3;10clk
819D 1D          DEC E
819E C27F81      JP NZ,SNDS1
81A1 C1          POP BC
81A2 D1          POP DE
81A3 E1          POP HL
81A4 F1          POP AF
81A5 C9          RET
              ; SOUND TABLE
81A6 7F       SNDTBL:DB 7F;so4
81A7 77          DB 77;so#4
81A8 71          DB 71;ra4
81A9 6A          DB 6A;ra#4
81AA 5F          DB 5F;do5
81AB 59          DB 59;do#5
81AC 54          DB 54;re5
81AD 4F          DB 4F;re#
81AE 47          DB 47;fA5
81AF 43          DB 43;fa#5
81B0 3F          DB 3F;so5
81B1 3B          DB 3B;so5#
81B2 35          DB 35;ra#5
81B3 32          DB 32;si5
81B4 2F          DB 2F;do6
81B5 2C          DB 2C;do#6
81B6 25          DB 25;mi6
81B7 27          DB 27;re#6
81B8 2A          DB 2A;re6
81B9 4B          DB 4B;mi5
81BA 38          DB 38;ra5
81BB 64          DB 64;si4
81BC 23          DB 23;fa6
81BD 21          DB 21;fa#6
              ;;
ALBLNK       =8126  ALBLNK1      =8128  CNTR         =FFF7
CNVT         =811D  COMP         =806E  COMP1        =808D
COMP2        =809F  COMP3        =80A8  COMP4        =80B8
COMP5        =80C2  COMP6        =80C7  COMP8        =80CE
D1_1         =02DF  DP1          =FFF4  DP3          =FFF6
DTBF         =E800  DTBF2        =E802  DTDPS        =05C0
DTST1        =800B  DTST2        =801D  DTST3        =8025
DTST4        =8031  GIVUP        =80D6  IX           =0000
KEY          =804A  KEYIN1       =0616  KEYIN1S      =8157
LED1         =FFF8  LED3         =FFFA  LED5         =FFFC
LED8         =FFFF  OK           =80E6  OK1          =80EA
RND          =80FF  RNDDT        =E810  SEGDT        =814D
```

```
SNDS1       =817F  SNDS2       =8184  SNDS3       =8193
SNDTBL      =81A6  SOUND       =8173  START       =8000
START1      =8045  STDP        =8139  STDT        =8145
TM01S       =815F  TM01S2      =8162  TM1M        =816B
TM1M_2      =816E  WAIT        =80F7
```

プログラム№15　SWAP2（SWAP2．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。

```
2010/9/17  14:34  swap2b.txt
END=8162
              ; SWAP2 for NDZ
              ; 03/05/03 5/5 5/15
              ;10/6/4 for ND80Z3
              ;
                   ORG $8000
                   RNDDT=$E810;$E811
                   LED1=$FFF8
                   DP1=$FFF4
                   DP3=$FFF6
                   CNTR=$FFF7;=DP4
              ;
                   DTDPS=$05C0
                   KEYIN1=$0616
                   KEYIN2=$0623
                   D1_1=$02DF;about 125microsec
              ;
              ;data buffer clear
8000 3EEF     START:LD A,EF
8002 D398        OUT (98),A
8004 3EFF          LD A,FF
8006 0608          LD B,08
8008 21F8FF        LD HL,LED1
800B 77       DTCLR:LD (HL),A
800C 23            INC HL
800D 10FC          DJNZ *DTCLR
              ;space position select
800F CD0E81        CALL RND
8012 E607          AND 07
8014 21F8FF        LD HL,LED1
8017 85            ADD A,L
8018 6F            LD L,A
8019 3600          LD (HL),00
              ;"UP" No. select
```

```
801B CD0E81   DTST1:CALL RND
801E E607        AND 07
8020 FE06          CP 06
8022 D21B80        JP NC,DTST1
8025 3C            INC A;from 1 to 6 select
8026 47            LD B,A
              ;"UP" data position select
8027 CD0E81   DTST2:CALL RND
802A E607          AND 07
802C 21F8FF        LD HL,LED1
802F 85            ADD A,L
8030 6F            LD L,A
8031 7E            LD A,(HL)
8032 B7            OR A
8033 CA2780        JP Z,DTST2;space position
8036 3C            INC A
8037 C22780        JP NZ,DTST2;already used
803A 3663          LD (HL),63;"UP"
803C 10E9          DJNZ *DTST2
              ;"down" data set
803E 21F8FF        LD HL,LED1
8041 0608          LD B,08
8043 7E       DTST3:LD A,(HL)
8044 B7            OR A
8045 CA4E80        JP Z,DTST4;space position
8048 3C            INC A
8049 C24E80        JP NZ,DTST4;alredy used
804C 365C          LD (HL),5C;"DOWN"
804E 23       DTST4:INC HL
804F 10F2          DJNZ *DTST3
              ;
8051 210000        LD HL,$0000
8054 22F6FF        LD (DP3),HL;DECIMAL COUNTER clear
8057 22F4FF        LD (DP1),HL
805A 3EFF          LD A,FF
805C D398        OUT (98),A
              ;key entry
805E CD3F81   KEY:CALL KEYIN1S;1-8 input
8061 FE09          CP 09
8063 D25E80        JP NC,KEY
8066 3D            DEC A
8067 FA5E80        JP M,KEY
              ;LED blink and wait WINC in
806A 21F8FF   BLNK0:LD HL,LED1
806D 85            ADD A,L
806E 6F            LD L,A
806F 7E            LD A,(HL)
8070 B7            OR A
8071 CA5E80        JP Z,KEY;space position
8074 57            LD D,A;LED data save
8075 47            LD B,A
8076 0E00          LD C,00
8078 71       BLNK1:LD (HL),C
8079 CD4F81        CALL TM01S
807C CD4781        CALL KEYIN2S;1-8,15 input
807F FE09          CP 09
8081 D28C80        JP NC,BLNK2
8084 3D            DEC A
```

```
8085 FA9180        JP M,BLNK3
8088 72            LD (HL),D
8089 C36A80        JP BLNK0
808C FE15      BLNK2:CP 15;WRITE INC
808E CA9780        JP Z,CHECK
8091 79        BLNK3:LD A,C
8092 48            LD C,B
8093 47            LD B,A
8094 C37880        JP BLNK1
               ;check
8097 7D        CHECK:LD A,L
8098 E5            PUSH HL;save HL(current position)
8099 E607          AND 07
809B CAB480        JP Z,CHECKR;current position=LED1
809E 2B            DEC HL
809F 7E            LD A,(HL)
80A0 B7            OR A
80A1 CAD080        JP Z,REP;left=space
80A4 BA            CP D
80A5 CAB480        JP Z,CHECKR;left=same
80A8 7D            LD A,L
80A9 E607          AND 07
80AB CAB480        JP Z,CHECKR;left=LED1
80AE 2B            DEC HL
80AF 7E            LD A,(HL)
80B0 B7            OR A
80B1 CAD080        JP Z,REP;left of left=space
80B4 E1        CHECKR:POP HL
80B5 E5            PUSH HL
80B6 7D            LD A,L
80B7 3C            INC A
80B8 CA0A81        JP Z,CANT;position=L8,cannot replace
80BB 23            INC HL
80BC 7E            LD A,(HL)
80BD B7            OR A
80BE CAD080        JP Z,REP;right=space
80C1 BA            CP D
80C2 CA0A81        JP Z,CANT;right=same,cannot replace
80C5 7D            LD A,L
80C6 3C            INC A
80C7 CA0A81        JP Z,CANT;right=L8,cannot replace
80CA 23            INC HL
80CB 7E            LD A,(HL)
80CC B7            OR A
80CD C20A81        JP NZ,CANT;right of right is not space,cannot replace
               ;replace
80D0 72        REP:LD (HL),D
80D1 E1            POP HL
80D2 3600          LD (HL),00
80D4 3AF7FF        LD A,(CNTR);DECIMAL COUNTER up
80D7 C601          ADD A,01
80D9 27            DAA
80DA 32F7FF        LD (CNTR),A
               ;all check
80DD 21F8FF        LD HL,LED1
80E0 0608          LD B,08
80E2 7E        ALCHK1:LD A,(HL)
80E3 FE5C          CP 5C;"DOWN"
```

```
80E5 C2EB80        JP NZ,ALCHK2
80E8 23            INC HL
80E9 10F7          DJNZ *ALCHK1
80EB B7       ALCHK2:OR A
80EC C25E80        JP NZ,KEY;not success
80EF 23            INC HL
80F0 05            DEC B
80F1 7E       ALCHK3:LD A,(HL)
80F2 FE63          CP 63;"UP"
80F4 C25E80        JP NZ,KEY;not success
80F7 23            INC HL
80F8 10F7          DJNZ *ALCHK3
              ;OK! ALL LED blink and COUNTER disp
80FA CD2C81        CALL ALBLNK
80FD CDC005        CALL DTDPS
8100 0614          LD B,14;=20 ,2sec wait
8102 CD4F81   WAIT:CALL TM01S
8105 10FB          DJNZ WAIT
8107 C30080        JP START
              ;cannot replace
810A E1       CANT:POP HL
810B C39180        JP BLNK3
              ;
              ;ransu
810E E5       RND:PUSH HL
810F D5            PUSH DE
8110 2A10E8        LD HL,(RNDDT)
8113 ED5F          LD A,R
8115 5F            LD E,A
8116 19            ADD HL,DE
8117 7C            LD A,H
8118 E603          AND 03
811A 67            LD H,A
811B 7E            LD A,(HL)
811C 23            INC HL
811D 86            ADD A,(HL)
811E 5F            LD E,A
811F 23            INC HL
8120 86            ADD A,(HL)
8121 23            INC HL
8122 86            ADD A,(HL)
8123 57            LD D,A
8124 ED5310E8      LD (RNDDT),DE
8128 7A            LD A,D
8129 D1            POP DE
812A E1            POP HL
812B C9            RET
              ;
              ;LED all blink
812C 060A     ALBLNK:LD B,0A;=10
812E 3EEF     ALBLNK1:LD A,EF
8130 D398        OUT (98),A
8132 CD4F81        CALL TM01S
8135 3EFF          LD A,FF
8137 D398        OUT (98),A
8139 CD4F81        CALL TM01S
813C 10F0          DJNZ *ALBLNK1
813E C9            RET
```

```
                ;
                ;KEYIN with DE,BC save
813F C5         KEYIN1S:PUSH BC
8140 D5            PUSH DE
8141 CD1606        CALL KEYIN1
8144 D1            POP DE
8145 C1            POP BC
8146 C9            RET
                ;
8147 C5         KEYIN2S:PUSH BC
8148 D5            PUSH DE
8149 CD2306        CALL KEYIN2
814C D1            POP DE
814D C1            POP BC
814E C9            RET
                ;
                ;0.1sec timer
814F D5         TM01S:PUSH DE
8150 1E64            LD E,64;=100
8152 CD5B81     TM01S2:CALL TM1M
8155 1D              DEC E
8156 C25281          JP NZ,TM01S2
8159 D1              POP DE
815A C9              RET
815B D5         TM1M:PUSH DE
815C 1608          LD D,08
815E CDDF02     TM1M_2:CALL D1_1
8161 D1            POP DE
8162 C9            RET
                ;
ALBLNK        =812C  ALBLNK1      =812E  ALCHK1       =80E2
ALCHK2        =80EB  ALCHK3       =80F1  BLNK0        =806A
BLNK1         =8078  BLNK2        =808C  BLNK3        =8091
CANT          =810A  CHECK        =8097  CHECKR       =80B4
CNTR          =FFF7  D1_1         =02DF  DP1          =FFF4
DP3           =FFF6  DTCLR        =800B  DTDPS        =05C0
DTST1         =801B  DTST2        =8027  DTST3        =8043
DTST4         =804E  KEY          =805E  KEYIN1       =0616
KEYIN1S       =813F  KEYIN2       =0623  KEYIN2S      =8147
LED1          =FFF8  REP          =80D0  RND          =810E
RNDDT         =E810  START        =8000  TM01S        =814F
TM01S2        =8152  TM1M         =815B  TM1M_2       =815E
WAIT          =8102
```

第2部　電子オルゴール

1．　電子オルゴールプログラム

　MUSIC．BTKは自動演奏プログラムです。
　電子オルガンプログラム　SOUND．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、MUSIC．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

nd80z3フォルダにあるMUSIC．BTKを、ND80ZⅢ（ND80Zモニタ）に、HIDWR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、nd80z3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HIDWRコマンドでND80ZⅢ（ND80Zモニタ）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8001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8001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8001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(約10μsec)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、ひく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EFFE(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。

```
2010/9/24  14:14  MUSIC4.TXT
END=80E5
              ;;; music 09/10/9 10/10 10/12 10/15
              ;;;
              ;;;MUSIC from MYCPU80 MUSIC
              ;10/6/4 6/5 6/15 for ND80Z3 CLK=6MHz
              ;7/20 9/24
              ;
                 ORG $8000
              ;
                 SPEED=$EFFF
                 REPEAT=$EFFE
                 MAE=$EFFD
                 DATA=$8100
                 END=$080F;MONITOR START
              ;
8000 3E03        LD A,03
8002 32FFEF      LD (SPEED),A
8005 210081      LD HL,DATA
8008 3AFDEF   SND1:LD A,(MAE)
800B 4F          LD C,A
800C 7E       SND11:LD A,(HL)
800D FEFF        CP FF
800F CA0F08      JP Z,END
```

```
8012 B7          OR A
8013 CA6F80      JP Z,YASUMI
8016 FE1E        CP 1E
8018 DA2080      JP C,SND12
801B 23          INC HL
801C 23          INC HL
801D C30C80      JP SND11
8020 F5       SND12:PUSH AF
8021 B9          CP C
8022 CC8D80      CALL Z,T5MS
8025 23          INC HL
8026 46          LD B,(HL)         ;NAGASA
8027 23          INC HL
8028 F1          POP AF
8029 E5          PUSH HL
802A 21AA80      LD HL,SNDTBL
802D 87          ADD A,A
802E 5F          LD E,A
802F 1600        LD D,00
8031 19          ADD HL,DE
8032 56          LD D,(HL)
8033 23          INC HL
8034 5E          LD E,(HL);KURIKAESI KAISU
8035 3AFFEF      LD A,(SPEED)
8038 6F          LD L,A
8039 60          LD H,B
803A 7B          LD A,E
803B 32FEEF      LD (REPEAT),A
              ;
803E 4A       SNDS1:LD C,D
803F 3EEF        LD A,EF          ;sp out=H,DMAoff
8041 D398        OUT (98),A
8043 E5       SNDS2:PUSH HL            ;DUMMY CLK=11
8044 E5          PUSH HL          ;DUMMY CLK=11
8045 E1          POP HL           ;DUMMY CLK=10
8046 E1          POP HL           ;CLK=10
8047 00          NOP              ;CLK=4
8048 0D          DEC C            ;CLK=4
8049 C24380      JP NZ,SNDS2      ;CLK=10
804C 4A          LD C,D
804D 3ECF        LD A,CF           ;sp out=L,DMAoff
804F D398        OUT (98),A
8051 E5       SNDS3:PUSH HL                ;DUMMY CLK=11
8052 E5          PUSH HL          ;DUMMY CLK=11
8053 E1          POP HL           ;DUMMY CLK=10
8054 E1          POP HL           ;CLK=10
8055 00          NOP              ;CLK=4
8056 0D          DEC C            ;CLK=4
8057 C25180      JP NZ,SNDS3      ;CLK=10
805A 1D          DEC E
805B C23E80      JP NZ,SNDS1
805E 3AFEEF      LD A,(REPEAT)
8061 5F          LD E,A
8062 05          DEC B
8063 C23E80      JP NZ,SNDS1
8066 44          LD B,H
8067 2D          DEC L
8068 C23E80      JP NZ,SNDS1
```

```
806B E1          POP HL
806C C30880      JP SND1
                 ;;;
806F 23       YASUMI:INC HL
8070 46          LD B,(HL)        ;NAGASA
8071 23          INC HL
8072 E5          PUSH HL
8073 3AFFEF      LD A,(SPEED)
8076 6F          LD L,A
8077 60          LD H,B
8078 CD9280   YASUMI2:CALL T40MS
807B 05          DEC B
807C C27880      JP NZ,YASUMI2
807F 44          LD B,H
8080 2D          DEC L
8081 C27880      JP NZ,YASUMI2
8084 E1          POP HL
8085 C30880      JP SND1
                 ;;;
8088 23       ERROR:INC HL
8089 23          INC HL
808A C30880      JP SND1
                 ;
808D 0E05     T5MS:LD C,05
808F C39480      JP T40MS2
                 ;
8092 0E28     T40MS:LD C,28               ;=40
8094 CD9C80   T40MS2:CALL T1MS
8097 0D          DEC C
8098 C29480      JP NZ,T40MS2
809B C9          RET
                 ;
809C F5       T1MS:PUSH AF                ;CK=11....11+7+10+10+10=48/6(8microsec)
809D 3E5F        LD A,5F           ;=99  CK=7
809F E5       T1MS2:PUSH HL;DUMMY CLK=11--------
80A0 E5          PUSH HL;DUMMY CLK=11          |
80A1 E1          POP HL;DUMMY CLK=10           |CK=60/6(10microsec)
80A2 E1          POP HL;DUMMY CLK=10           |
80A3 00          NOP ;CLK=4                    |
80A4 3D          DEC A             ;CK=4       |10x99=990microsec
80A5 C29F80      JP NZ,T1MS2       ;CK=10---
80A8 F1          POP AF            ;CK=10
80A9 C9          RET               ;CK=10 + CALL CK=10
                 ;
                 ; SOUND TABLE
80AA 00       SNDTBL:DB 00;DUMMY
80AB 00          DB 00;DUMMY
80AC 7F          DB 7F;so4
80AD 10          DB 10
80AE 77          DB 77;so#4
80AF 11          DB 11
80B0 71          DB 71;ra4
80B1 12          DB 12
80B2 6A          DB 6A;ra#4
80B3 13          DB 13
80B4 64          DB 64;si4
80B5 14          DB 14
80B6 5F          DB 5F;do5
```

```
80B7 15          DB 15
80B8 59          DB 59;do#5
80B9 16          DB 16
80BA 54          DB 54;re5
80BB 18          DB 18
80BC 4F          DB 4F;re#5
80BD 19          DB 19
80BE 4B          DB 4B;mi5
80BF 1A          DB 1A
80C0 47          DB 47;fa5
80C1 1C          DB 1C
80C2 43          DB 43;fa#5
80C3 1E          DB 1E
80C4 3F          DB 3F;so5
80C5 1F          DB 1F
80C6 3B          DB 3B;so#5
80C7 21          DB 21
80C8 38          DB 38;ra5
80C9 23          DB 23
80CA 35          DB 35;ra#5
80CB 25          DB 25
80CC 32          DB 32;si5
80CD 27          DB 27
80CE 2F          DB 2F;do6
80CF 2A          DB 2A
80D0 2C          DB 2C;do#6
80D1 2C          DB 2C
80D2 2A          DB 2A;re6
80D3 2F          DB 2F
80D4 27          DB 27;re#6
80D5 32          DB 32
80D6 25          DB 25;mi6
80D7 35          DB 35
80D8 23          DB 23;fa6
80D9 38          DB 38
80DA 21          DB 21;fa#6
80DB 3B          DB 3B
80DC 1F          DB 1F;so6
80DD 3F          DB 3F;
80DE 1D          DB 1D;so#6
80DF 43          DB 43;
80E0 1B          DB 1B;ra6
80E1 47          DB 47;
80E2 1A          DB 1A;ra#6
80E3 4A          DB 4A;
80E4 18          DB 18;si6
80E5 50          DB 50;
                ;END
DATA        =8100  END         =080F  ERROR       =8088
MAE         =EFFD  REPEAT      =EFFE  SND1        =8008
SND11       =800C  SND12       =8020  SNDS1       =803E
SNDS2       =8043  SNDS3       =8051  SNDTBL      =80AA
SPEED       =EFFF  T1MS        =809C  T1MS2       =809F
T40MS       =8092  T40MS2      =8094  T5MS        =808D
YASUMI      =806F  YASUMI2     =8078
```

2. 電子オルゴールデータ

①プログラムの8001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(8001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(8001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(8001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(8001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(8001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(8001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(8001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(4001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
```