

```

8009 CD 15 80      R_JMP: CALL  WORM_MON      ; W O R M コントロール部を呼び出す
800C E1           POP      H          ; スタックから戻り番地を取り出し
800D 7D           MOV      A,L
800E 93           SUB      E          ; 後戻りさせたい番地だけひいて
800F 6F           MOV      L,A
8010 7C           MOV      A,H
8011 9A           SBB     D
8012 67           MOV      H,A
8013 E5           PUSH   H          ; スタックに戻り番地をセットする
8014 C9           RET              ; そしてリターン
    
```

; W O R M コントロール部  
 WORM\_MON:

```

8015 F5           PUSH   PSW
8016 C5           PUSH   B
8017 D5           PUSH   D
8018 E5           PUSH   H

8019 7D           MOV      A,L      ; HLレジスタの中身を見て
801A FE 00       CPI     00H      ; 終了かどうかを判定
801C C2 2A 80    JNZ     NEXT
801F 7C           MOV      A,H
8020 FE F0       CPI     0F0H
8022 C2 2A 80    JNZ     NEXT

8025 E1           POP     H        ; 終点ならスタックを元に戻して
8026 D1           POP     D        ; 終了する。
8027 C1           POP     B
8028 F1           POP     PSW
8029 FF           RST     7

802A CD D5 02    NEXT: CALL  DISP4H    ; W O R Mの現在位置を表示
802D CD 00 03    CALL  DERAY     ; 時間稼ぎ
8030 E1           POP     H
8031 D1           POP     D
8032 C1           POP     B
8033 F1           POP     PSW
8034 C9           RET
    
```

; 使用するメモリーは作業用の2バイトだけ  
 ; 必ずW O R Mよりも下の番地に確保すること

8035 00 81 COP\_ADR:DW WORM

; W O R M ルーチン

```

8100           ORG     8100H
8100 06 19       WORM: MVI     B,END-WORM+3 ; メモリーのコピー回数をセット
8102 2A 35 80   LOOP: LHLD   COP_ADR
8105 7E         MOV     A,M      ; コピー元から内容を読みだし
8106 11 19 00   LXI     D,END-WORM+3 ; W O R Mの長さを足して
8109 19         DAD     D
810A 77         MOV     M,A      ; コピー先に内容に移す
810B 2A 35 80   LHLD   COP_ADR
810E 23         INX     H
810F 22 35 80   SHLD   COP_ADR
    
```

```

8112 11 17 00   LXI     D,END-LOOP+3 ; 相対番地の距離をセット
8115 05         DCR     B
8116 C4 09 80   END:   CNZ     R_JMP
    
```

; これでプログラムは終わり。読んでくれてありがとう。

This file is well assembled.