Teh pet vrstic bomo dodali na koncu programa, zdaj pa se vrnimo k podprogramu GOR.

```
        PUSH  BC        ;Shrani koordinate,
        INC   B         ;izračunaj
        INC   B         ;          naslov 3.
        INC   B         ;                   zloga
        CALL  NASLOV    ;                         nad junakom
        CALL  BIT40     ;in preveri ali ni zadel metulja
                        ;ali ključa.
        INC   HL        ;Naslov sosednjega zloga (preveriti
        CALL  BIT40     ;moramo oba zloga, ker sta tako
        POP   BC        ;Krpan kot metulj široka 16 bitov).
```

Da bo gibanje naše figure gladko, se mora v vse smeri premikati bit za bitom, nikoli za več bitov naenkrat. Da pa bo program enostavnejši, bomo premik vsakič ponovili osemkrat. To pomeni, da se bo na vsak pritisk tipke figura pomaknila, bit za bitom, za osem bitov. V ta namen bomo naredili zanko. Če ne bi uporabljali para BC za shranjevanje koordinat, bi si lahko lepo pomagali z ukazom DJNZ. Tako pa bomo števce shranjevali v rezervirani pomnilniški celici, ki ju bomo označevali STEVC1 in STEVC2.

```
        LD    A, 8
GR      LD    (STEVC1), A   ;Prvi števec.
        INC   B             ;To je zdaj koordinata 1. zloga nad
        PUSH  BC            ;figuro. Shrani vrednost.
        LD    A, 16
ZAN1    LD    (STEVC2), A   ;Drugi števec.
        CALL  NASLOV        ;izračunaj naslov zloga...
        PUSH  HL            ;...in ga shrani.
        DEC   B
        CALL  NASLOV        ;Izračunaj naslov zloga pod njim,
        POP   DE            ;naslov zgornjega zloga pa daj v
                            ;par DE.
        LD    A, (HL)       ;Prenesi vrednost spodnjega zloga..
        LD    (DE), A       ;...v zgornji zlog.
        INC   HL            ;To sta zdaj naslova...
        INC   DE            ;          ...sosednjih dveh zlogov.
        LD    A, (HL)       ;Prenesi tudi tu...
```



```
        LD    (DE), A       ;                ...vrednost navzgor.
        LD    A, (STEVC2)   ;Se je zanka že ponovila 16 krat ?
        DEC   A             ;(se pravi: se je že premaknila vsa
        JR    NZ, ZAN1      ;figura?). Če se ni, ponovi ZAN1,
        POP   BC
        LD    A, (STEVC1)   ;sicer zmanjšaj prvi števec in po-
        DEC   A             ;novi veliko zanko. Če se je velika
        JR    NZ, GR        ;zanka že ponovila osemkrat,
        RET                 ;se vrni.
```

To je bil podprogram za pomik navzgor. Zdaj pa premik v levo:

```
LEVO    LD    A, 8          ;Če je figura na levem robu (ob
        CP    C             ;stolpcu ure), ne more na levo. Za-
        RET   Z             ;to se takoj vrni v glavno zanko.
        PUSH  BC
        LD    A, C          ;Najprej bomo preverili, ali ni na
        SUB   8             ;levi strani lika metulj ali ključ.
        LD    C, A
        LD    A, B
        SUB   5             ;Za preverjanje bomo uporabili isti
        LD    B, A          ;zlog kot pri premiku navzgor. Nje-
        CALL  NASLOV        ;gove koordinate so zdaj v paru BC.
        CALL  BIT40
        LD    A, B
        SUB   8
        LD    B, A          ;Kontroliramo tudi zlog, ki je osem
        CALL  NASLOV        ;vrstic niže (ker je figura visoka
        CALL  BIT40         ;16 vrstic).
        POP   BC
        LD    A, 8
LV      LD    (STEVC1), A   ;Prvi števec.
        PUSH  BC
        LD    A, C          ;Ker bo pomik v levo, potrebujemo
        ADD   A, 15         ;koordinati desnega roba figure.
        LD    C, A          ;Ti sta zdaj v paru BC.
        LD    A, 16
ZAN4    LD    (STEVC2), A   ;Drugi števec.
        CALL  NASLOV
        SLA   (HL)          ;Pomakni zlog v levo. Skrajni levi
                            ;bit (bit 7) gre v bit prenosa.
```