

```

DEC HL      ;Naslov naslednjega zloga na levi.
RL (HL)     ;Pomakni zlog v levo. V desni bit
            ;(bit 0) pride bit prenosa iz prej-
            ;šnjega pomika, tj. bit 7 sosednje-
            ;ga zloga. V bit prenosa se nato
            ;prenese bit 7 zloga, ki ga zdaj
            ;premikamo.
DEC HL      ;Naslednji zlog proti levi
RL (HL)     ;in še en pomik, nato pa...
DEC B       ; ...v naslednjo vrstico navzdol.
LD A,(STVC2)
DEC A       ;Se je že premaknila vsa figura?
JR NZ,ZAN4  ;Če se še ni, ponovi zanko.
POP BC      ;Lik se je pomaknil v levo, zato
DEC C       ;zmanjšaj koordinato x za 1.
LD A,(STVC1) ;Se je pomik že ponovil osemkrat?
DEC A
JR NZ, LV   ;Če se še ni, ga ponovi, sicer...
RET         ; ...se vrni v glavno zanko.

```

Podprograma za premik v desno in navzdol sta sestavljena povsem podobno. Da se ne bomo preveč zamudili, se zato zdaj ob nji ju ne bomo ustavljali. S potrebnimi opombami vred ju bomo vključili v končni zapis. Zdaj pa vzemimo v precep podprogram, ki uravnava gibanje metulja.

```

METULJ     EXX      ;Podatke o metulju smo naložili v
                ;zamenljivi registriški niz.
                PUSH HL ;To vrednost moramo shraniti za
                ;nemoten povratek v basic (gl.
                ;poglavje o menjavi registrov).
LD HL,GIB   ;V rezerviranem zlogu GIB bomo
                ;šteli zamabe kril. Preden se sme
                ;metulj premakniti drugam na ekranu
                ;se morajo izvrstiti vse 4 sličice.
                ;Zlogu GIB bomo dali vrednost 4 in
                ;nato odštevali. Ko bodo opravljeni
                ;vsi štirje gibi (tj. ko bo vred-
                ;nost zloga GIB 0), bomo poklicali
                ;podprogram KAM, ki bo premikal
                ;metulja po zaslonu.

```

Zamah metuljevih kril sestavljajo štiri slike, vsaka ima velikost 16 \* 16 točk oziroma bitov. Če preracunamo, znese to 128 zlogov. Da jih bomo lahko prenašali na zaslon, jih bomo morali shraniti nekje v pomnilniku. Tóčen naslov bomo določili kasneje, za zdaj ga imenujmo LIK12. Čisto na začetku bomo dali vrednost LIK12 v par DE zamenljivega niza.

```

LD A, 16    ;Pripravi števec
PUSH BC     ;in shрани koordinato metulja.
PUSH AF     ;Števec bomo spravili kar na sklad.
PUSH DE     ;Shrani naslov s katerega prenašamo
CALL NASLOV ;metuljevo podobo in izračunaj na-
POP DE      ;slov metuljevega zloga na zaslonu.
LD A,(DE)   ;Prenesi s pomočjo registra A zlog
LD (HL), A  ;slike iz pomnilnika na ekran.
INC DE      ;To sta naslova za...
INC HL      ; ...desni zlog.
LD A,(DE)   ;Prenesi ustrezni zlog iz
LD (HL), A  ;pomnilnika na zaslon.
INC DE
DEC B       ;Naslednji zlog v pomnilniku.
POP AF      ;Koordinata naslednje vrstice.
DEC A       ;Vzemi števec...
JR NZ,RACUN ;...in ga zmanjšaj.
POP BC      ;Če še ni 0, ponovi zanko,
POP HL      ;sicer
EXX         ; "pospravi
RET         ; za seboj"
            ;in se vrni.

```

Naslednji korak je podprogram, ki bo premikal metulja po zaslonu. Da program ne bo postal predolg, se bomo odpovedali gladkemu gibanju; premiki bodo znašali 16 bitov naenkrat. Če pogledate začetek podprograma METULJ, boste videli, da ima ob klicu KAM registriški par HL vrednost GIB, vsebina zloga pa mora doseči 0, da pride do klica KAM. Da se bo program po vrnitvi lahko prav nadaljeval,

```

KAM        LD A, 4      ;moramo najprej zlogu GIB
            LD (HL), A  ;dati vrednost 4.
            PUSH BC    ;Shrani položaj (koordinati).
            LD A, 16

```