

```

ZANKA      PUSH AF
           CALL NASLOV
           LD (HL), A
           INC HL
           LD (HL), A
           DEC B
           POP AF
           DEC A
           JR NZ, ZANKA
           POP BC
           EXX
           PUSH BC
           EXX
           POP DE
           LD A, B
           CP D
           JR Z, HORIZ
           JR C, NAGOR

NADDL     LD A, B
           SUB 16
           LD B, A
           JR HORIZ
           LD A, B
           ADD A, 16
           LD B, A
           LD A, C
           CP E
           JR C, NADESN
           LD A, C
           SUB 16
           LD C, A
           JR VEN
           LD A, C
           ADD A, 16
           LD C, A

```

; Najprej bomo z ZANKO izbrisali
; sliko metulja, tj. ustreznim zlogom na zaslону bomo dali vrednost
; 0. Ker se koordinati vedno nanašata na levi bit (bit 7), ima A po
; klicu NASLOVA vrednost 0.
; Naslednja vrstica.
; Je zanka že končana?
; Če še ni, jo ponovi.
; Daj koordinatama začetno vrednost.
; Zamenjaj registre,
; shrani koordinate Krpana na sklad
; in ponovno zamenjaj registre.
; Spravi koordinate Krpana v DE.
; Primerjaj koordinate Y junaka in
; metulja.
; Če sta enaki, skoči naprej.
; Če je Krpan višje od metulja,
; mora metulj navzgor,
; sicer pa navzdol. V tem primeru
; zmanjšaj koordinato Y metulja
; in skoči naprej.
; Za dvig je treba višino povečati.
; Zdaj pa še primerjava koordinat X.
; Če je koordinata X metulja manjša
; od Krpanove, naj gre metulj desno,
; sicer pa levo. V tem primeru
; zmanjšaj oddaljenost od levega
; roba in
; skoči naprej.
; Za pomik v desno...
; ...povečaj oddaljenost od levega
; roba.

Zdaj moramo preveriti, ali ni metulj ob premiku zadel ključa ali junaka igre. Samo testiranje bo tu nekoliko bolj zapleteno, zato ga bomo oblikovali kot podprogram z naslovom METST

```

(METULJEV TEST). Za testiranje bomo uporabljali četrte zloge na posameznih položajih.
VEN       PUSH BC
           LD A, B
           SUB 4
           LD B, A
           CALL NASLOV
           CALL METST
           INC HL
           CALL METST
           LD A, B
           SUB 8
           LD B, A
           CALL NASLOV
           CALL METST
           INC HL
           CALL METST
           POP BC
           LD DE, LIKIZ
           RET

           ;Shrani koordinate.
           ;Izračunaj njen naslov ter klicni
           ;podprogram za testiranje.
           ;Naslov sosednjega zloga -
           ;preveri njegovo vsebino.
           ;Zdaj pa še dva zloga, ki sta
           ;osem vrstic nižje.
           ;Ponovi isti postopek.
           ;Na naslovu LIKIZ so shranjene
           ;slike metulja.
           ;Vrni se v zanko METULJ.

```

S podprogramom METST ugotavljamo, ali je metulj zadel ključ ali junaka še preden se je metulj v resnici premaknil. Seveda želimo na zaslону videti, kako metulj "požre" junaka in bomo šele potem zaključili igro. Zato bomo v podprogramu uporabili rezerviran zlog TESTI. Če bo metulj zadel Krpana, bo TESTI dobil vrednost 1. Ta zlog bomo kontrolirali, ko bo premik že opravljen - in ustrezno ukrepali. Vendar moramo zaradi tega nekoliko dopolniti konec podprograma METULJ, in sicer bomo za ukazom EXX dodali tele vrstice:

```

LD HL, TESTI
DEC (HL)
RET NZ
CALL KONEC

```

;Preveri, ali je metulj zadel
;Krpana. Če ga ni...
;...se vrni.
;Sicer pokliči podprogram, ki bo
;zaključil igro.

Zdaj pa k podprogramu METST. Katere zloge oz. bite testiramo v tem podprogramu boste najlaže razumeli, če boste pogledali slike naših likov.