

```

METST BIT 2, (HL)
JR Z, P1
LD A, 1
LD (TEST1), A
P1 BIT 5, (HL)
JR Z, P2
LD A, 1
LD (TEST1), A
P2 BIT 3, (HL)
CALL NZ, MINUS
RET

```

Če metulj "pobere" ključ, naj preostane igralcu za kazen manj časa. Da pa bo igra tekla kot je prav, bomo ob tem povečali vrednost zloga KLJUC, ki hrani število že pobranih ključev.

```

MINUS PUSH BC
LD HL, KLJUC
INC (HL)
LD B, 30
PUSH BC
LD BC, (VISURE)
CALL NASLOV
DEC B
JR NZ, NAPR

```

:Shrani koordinate.  
:Povečaj števec pobranih ključev.  
:Preostali čas naj se zmanjša za 30. Pripravi zanko in spravi števec v registru B na sklad.  
:Koordinate stolpca ure.  
:Izračunaj naslov vrhnjega zloga ure.  
:Zmanjšaj višino stolpca.  
:Če višina še ni 0 (tj. ura še ni potekla), skoči na naslov NAPR.

Tu pa se srečamo s težavo. Kaj storiti, če se na tem mestu izkaže, da je igralcu čas potekel? Takoj bi bilo potrebno končati igro in se vrniti v basic, a zdi se, da bi bil to precej zapleten posel. Če se spomnite, smo iz glavne zanke klicali podprogram METULJ (in tam zamenjali registrski niz), od tam KAM, iz podprograma KAM podprogram METST ter iz njega MINUS, v katerem smo zdaj. Vendar je naše znanje kos na videz zelo zapletenemu vozilu. Najprej bomo dali registrskemu paru zamenljivo vrednost 10072, ki je potrebna za...

```

LD HL, 10072
EXX
LD BC, (TOCKE)

```

:...nemoteno vrnitev v basic  
:ter zamenjali registrski niz.  
:V par BC nalozijo dosežene točke.

Ob klicih podprogramov so se na skladu nabirali naslovi za povratke. Da bi prišli do pravega naslova za vrnitev v basic, bi torej morali te naslove odstraniti. Lahko pa si pomagamo na preprostejši način. Shranimo na začetku programa naslov, na katerem je naslov za vrnitev v basic, v rezervirana zloga STKPT! Ko se bomo želeli vrniti, bomo dali kazalcu sklada to vrednost:

```

LD SP, (STKPT)
RET

```

:ter se brez težav vrnili.

Če pa ura še ni potekla, naj se program nadaljuje na NAPR:

```

NAPR LD (VISURE), BC
LD (HL), 129
POP BC
DUNZ VIBA
XOR A
LD (23672), A
POP BC
RET

```

:Shrani zmanjšano višino ter izbriši zgornjo vrstico ure.  
:Vzemi števec s sklada in če še ni nič, ponovi zanko.  
:Daj uri začetno vrednost,  
:vzemi koordinato in  
:se vrni.

Zdaj pa smo že blizu konca. Manjka nam le še nekaj podprogramov, ki smo jih že "uporabili", nimamo pa jih še zapisanih. Prvi je podprogram ZADET, ki ga bomo klicali, kadar bo junak pobral ključ. KONEC bomo klicali, kadar se bo Krpan zadel v metulja ali obratno, OBRAC pa za obratun točk, kadar se bo igra uspešno končala. Vendar se še prej za trenutek ustavimo pri dveh programih iz ROM-a, ki ju bomo uporabili. Pri obratunu bomo klicali podprogram z zgovornim naslovom "HL=HL\*DE" na naslovu 30A9. Ta nam vrne v registrskem paru HL zmožek vrednosti, ki ju pred klicem vsebujeta HL in DE. Seveda morata biti števili takšni, da njun zmožek ni večji od 65535. Če pa se vseeno zgodi, da sta vrednosti preveliki, je ob povratku zastavica prenosa dvignjena. "HL=HL\*DE" spremeni vrednost v paru BC. Drugi podprogram, ki ga kanimo klicati iz ROM-a, pa je BEEPER na naslovu 03B5. Ta nadzoruje zvočnik in daje zvok, ki ga opredelimo z vrednostima v parih DE in HL. V DE mora biti zmožek frekvence in trajanja zelenega tona, v HL pa število potrebnih T stanj, deljeno s štiri (če vam ni povsem jasno, se ne vznemirjajte: ni pomembno).

Podprogram ZADET (igrallec je pobral ključ):