metulja, ki leta. Dovolj bodo štiri, ki naj izgledajo takole:

```
.................   .................
.................   .................
.................   .......■..■......
...■.........■...   .....■■..■■.....
...■■.......■■...   ....■■■..■■■....
....■■....■■.....   ....■■■..■■■....
....■■■..■■■.....   .......■..■......
........■■.......   ........■■.......
......■■■■.......   ......■■■■.......
.....■■■■■■......   .....■■■■■■......
....■..■■..■.....   ....■..■■..■.....
..■■..■..■..■■...   ..■■..■..■..■■...
.....■.....■.....   .....■.....■.....
...■■.......■■...   ...■■.......■■...
.................   .................
.................   .................
```

```
.................   .................
.................   .................
.................   .................
.................   .................
..■■.........■■..   .................
...■■.......■■...   ...■■■.....■■■...
...■■■.....■■■...   ..■■■■■...■■■■■..
....■■■...■■■....   ....■■■...■■■....
........■■.......   ........■■.......
......■■■■.......   ......■■■■.......
....■■■■■■■■.....   ....■■■■■■■■.....
...■...■■...■....   ..■■...■■...■■...
..■...■■...■■...■..   .....■■...■■.....
...■■.......■■...   ...■■.......■■...
.................   .................
.................   .................
```

Zdaj že imamo junaka, metulja in ključe; manjka le še ura. Naj za uro služi stolpec na levi strani zaslona, ki se bo nižal, ko bo čas tekel.

Pretehtajmo zdaj, kateri deli programa bodo zapisani v strojnem jeziku in kateri v basicu. Ključi morajo biti razpostavljeni naključno. Žal funkcija RND, ki jo v ta namen uporabljamo v basicu, v ROM-u ni zapisana kot podprogram temveč kot del večje zanke. Zato bomo ključe razpostavili iz basica. Prav tako bomo iz basica narisali stolpec ure. Vmesne dele, pri katerih je bistvena hitrost, bomo napisali v strojnem jeziku. Na koncu pa se bomo spet vrnili v basic in z njim zapisali zaključek.

Preden se lotimo programa, še dve stvari, s katerima bi se sicer srečali kasneje, a je zaradi lažje razlage bolje da ju razjasnimo zdaj. Prva je vprašanje, kako bomo uravnavali tek ure. Najenostavneje bo, če uporabimo vrednost, shranjeno v treh zlogih med sistemskimi spremenljivkami. Ta vrednost se ob vsaki prekinitvi (se pravi vsako 1/50 sekunde) poveča za eno. Mi bomo uporabili najnižji zlog na naslovu 23672 (ki se najhitreje spreminja) in po njem uravnali uro. Drugo vprašanje zadeva označevanje položaja figur in njihovo premikanje. Razporeditev naslovov v zaslonski datoteki ni povsem preprosta - in ker sta Krpan in metulj velika 16 * 16 bitov, je videti to vprašanje še za stopnjo bolj zapleteno. Lepo bi bilo, če bi mogli uporabljati običajne koordinate - denimo takšne kot z ukazom PLOT v basicu. Zaslon (natančneje: zgornjih 22 vrst) je razdeljen na 176 vrstic in 256 stolpcev. Koordinati v ukazu PLOT imata zato lahko vrednost od 0 do 175 (višina = koordinata y) oz. od 0 do 255 (širina = koordinata x). No, sreča je na naši strani: tudi v strojnih programih lahko uprabljamo te koordinate s pomočjo podprograma PIXEL_ADD, ki ga najdemo v ROM-u na naslovu 22AAH (= 8874). V register B naložimo koordinato y (oddaljenost od spodnjega roba zaslona) neke točke (bita), v register C pa pa koordinato x (oddaljenost od levega roba). Po klicu podprograma PIXEL_ADD dobimo v paru HL naslov zloga, v katerem je ta točka, v registru A pa njen položaj v zlogu (od 0 do 7). Podprogram spremeni vrednosti v parih BC in DE. Uporaba podprograma PIXEL_ADD je zelo prijetna možnost, zato se kar odločimo zanj. Zaradi tega bo najbolj priročno, če bomo imeli koordinate vseskozi spravljene v parih BC: v paru BC višino in odmik Krpanovega položaja, v paru BC izmenjalnega registrskega niza pa višino in odmik metulja. To pa nam nalaga posebno skrb za vrednosti v obeh parih BC: spraviti ju bomo morali, kadarkoli bo nevarnost, da se uničita. Zato se kar takoj odločimo za podprogramček z imenom NASLOV: