

Podatki o številčnih spremenljivkah obsegajo 8 zlogov. Prvi zlog vsebuje ime spremenljivke, drugi ima vedno vrednost OE, zadnji pa 2C oz. 29H. Ti trije zlogi so za nas nepomembni. Vaznih je preostalih 5 zlogov. V njih je shranjena vrednost spremenljivke. Če je to decimalno število, je zapisano v plavajoči vejici (podrobneje o tem v poglavju Aritmetika s plavajočo vejico). Če pa je vrednost celo število, so 3., 4. in 7. zlog enaki 0, v 5. zlogu je nizki, v 6. pa visoki zlog števila. Podatki o znakovnih spremenljivkah zasedajo 9 zlogov. V prvem in drugem je ime spremenljivke, prav tako so za nas neuporabni 3., 4. in 9. zlog. Vazni pa so preostali 4 zlogi. V 5. in 6. je naslov znakovne spremenljivke (se pravi: naslov besede) v 5. nizki, v 6. visoki zlog naslova. V zlogih 7 in 8 pa je dolžina besede (število znakov) - v 7. nizki, v 8. visoki zlog.

Bistvo in uporabnost vsega tega? Če v basicu napišete DEF FN F(X) = USR naslov in nato uporabite FN F(spremenljivka), boste s tem pognali strojni program. Ta pa bo lahko uporabil spremenljivko, ki ste jo navedli v oklepaju. Prebral jo bo z zaporedjem ukazov

```

ZA5COB LD HL, (DEFADD) ;DEFADD => 23563
23 INC HL
23 INC HL
23 INC HL
23 INC HL
4E LD C, (HL)
23 INC HL
46 LD B, (HL)

```

Zdaj, ko smo se seznanili z zamisljivo posredovanja podatkov strojnemu programu si oglejmo še drugi način, ki je pravzaprav veliko boljše možnost kot pravkar opisani. Prva vrstica programa v basicu naj bo DIM X(N), kjer je N število podatkov, ki jih moramo prenesti programu v strojnem jeziku. V sistemski spremenljivki VARS bo v tem primeru shranjen začetni naslov polja X, naslov prve vrednosti pa bo za šest večji od začetnega naslova. Če je polje znakovno (tj. če smo definirali DIM Xf(N)), bo vsaka vrednost zasedala en zlog. Če je polje številčno (tj. DIM X(N)), pa je vsaka vrednost zapisana v 5-zložni obliki (glej poglavje "Aritmetika s plavajočo vejico"). Izračun naslova posamezne vrednosti in dostop do nje je tako lažji in lepši. Če moramo posredovati večje število vrednosti, pa je uporaba polja tudi dosti manj zamudna.

IZPISOVANJE NA ZASLON

Prav gotovo že nestrpno čakate, da bodo vaši napori v strojnem jeziku obrอดili sad. Rezultati doslej obravnavanih programov so bile v glavnem zgolj številke - in verjamem, da vam zanje kljub vsemu ni dosti mar. Zato se bomo zdaj lotili naloge, ki bo dala VIDNE plodove: izpisovanja na zaslon.

Pisanje na ekran je v strojnem jeziku precej enostavno, celo nekoliko podobno kot v basicu. Uporabljamo ukaz RST 10, ki izpiše znak, katerega kod je v registru A. Preskusite naslednji program:

```

AF XOR A ;= LD A, 0
323C5C LD (TV FLAG), A
3E2A Zanka LD A, 2A
D7 RST 10
18FB JR Zanka

```

Kot vidite smo morali v sistemsko spremenljivko TV FLAG (najdete jo v 25. poglavju priročnika za Spectrum) na začetku naložiti vrednost 0 (kaj če tega ne bi naredili? Preskusite program brez prvih dveh vrstic!). Ko boste pognali program, se bo ekran napolnil z zvezdicami in to zelo zelo hitro (mnogo hitreje kot z "10 PRINT "*";GO TO 10" v basicu).

Vendar je to šele prvi korak. Mi pa želimo imeti (pod)program, ki bi nam izpisal kakršnokoli besedilo - od "DA" do "Kdor visoko leta, nizko pade", ki bi še rumeno-rdeče utripalo. Recimo, da takšen podprogram že obstaja in se imenuje S_PRINT. Uporabljati ga želimo v obliki: CALL S_PRINT s podatki "PAPIR rumen CRNILO rdeče UTRIPANJE vključeno Kdor visoko leta, nizko pade" (besede z velikimi tiskanimi črkami uporabljamo namesto ustreznih ukazov v basicu). Poglejmo primer:

```

CDXXX CALL S_PRINT
SA647261766F DEFM Zdravo
OO DEFB 00

```

DEFM (DEFINE MESSAGE = označi sporočilo) in DEFB (DEFINE BYTE = določi zlog) nista prava strojna ukaza. Uporabljamo ju le v