

pomnilnika. Prav tako izbriše vrednosti v prostorih 0 in 1 ukaz RST 10, kadar izpisuje grafične znake. Zaradi tega je najbolj zanesljivo uporabljati le prostore 3, 4 in 5. Poglejmo zdaj primer: želeli bi izračunati izraz  $\sin X + \cos X$ . Recimo, da je številko X že na vrhu računskega sklada. Uporabimo zaporedje

```

EF      RST 28
CS      DEFB CS (spravi_5) ;spravi X v prostor 5. X ob tem
                                ostaja tudi na vrhu sklada.
1F      DEFB 1F (sin)      ;izračunaj SIN X. Rezultat nad-
                                mesti vrednost na vrhu sklada
                                (t.j. odstrani X).
ES      DEFB ES (vzemi_5)  ;porini vrednost iz prostora 5 na
                                sklad. Na vrhu sklada je zdaj
                                X, pod njim SIN X.
20      DEFB 20 (cos)      ;izračunaj COS X. Rezultat naj
                                nadomesti vrednost na vrhu skla-
                                da (t.j. odstrani X).
OF      DEFB OF (seštej)   ;odstrani zgornji dve vrednosti
                                (SIN X in COS X) s sklada in ju
                                nadomesti z njuno vsoto.
38      DEFB 38 (končaj)

```

Vsoto dveh kotnih funkcij smo tako izračunali v samo sedmih zlogih! Podobno je tudi z rabo drugih operacij. Opozoriti pa velja na operacije VAL, VAL\$ in STR\$. Njihova raba je bolj zapletena, zato naj bi se jih lotil le tisti, ki dobro pozna Spectrumov nadzorni program.

Toliko za okus. Raziskovanje številnih preostalih možnosti prepuščam vam, da boste imeli od učenja res pravo korist.

## KAR SE JANEZEK NAUČI . . .

V zadnjem poglavju bomo pridobljeno znanje uporabili za izdelavo programa v strojnem jeziku. In česa bi se v tem programu lotili? Gotovo je najprivlačnejša misel o igrici. Pri oblikovanju takšnega programa smo povsem svobodni, edina omejitev je naša iznajdljivost. Odlotimo se torej za igro, ki bo zahtevala nekaj programerske veštine, računalnikove hitrosti in - na koncu - igralčeve spretnosti. Program bomo vnašali z urejevalnikom, zato bomo skrbeli, da ne bo predolg.

Predlagam igro, ki bo miroljubnejše narave. Odpovejmo se streljanju in pobijanju nasprotnikov, ki brzijo čez zaslon. Raje naj igralec npr. zbira določene predmete, pri tem pa naj mu bo na voljo le omejen čas. Več predmetov ko bo zbral, večje bo število točk; hitreje ko jih bo zbral, toliko boljši bo uspeh. Da pa ne bo vse odvisno le od hitrosti, pošljimo v igro še nasprotnika. Ta naj preganja našega junaka. Če ga bo ujel, bo to pomenilo takojšen neuspešen zaključek.

Obrise igre zdaj že imamo. Lotimo se najprej izdelave načrta, nato pa bomo postopoma zapisali program. Za začetek moramo pripraviti vse, kar bomo v igri potrebovali:

Začetek      Razpostavi predmete.  
 Pripravi uro.  
 Narizi junaka.  
 Narizi nasprotnika.

Še kaj? No dai vsakič bo treba preveriti, ali ima junak še kakšno življenje, treba bo šteti, ali so razpostavljeni že vsi predmeti... in še kaj. A to so že podrobnosti - in posebej smo poudarili, da podrobnosti vedno prihranimo za na konec. Torej lahko rečemo, da je načrt začetka igre narejen. Zdaj je na vrsti glavni, bolj zapleteni del - igra sama. Ta naj ima eno glavno zanko, iz katere bomo po potrebi klicali stranske zanke, t.j. podprograme. V glavni zanki bo tekla ura, spremljali bomo premike igralca, premikali nasprotnika in kontrolirali, ali so že pobrani vsi predmeti. Takoles:

Ura            Zmanjšaj preostali čas.  
 Je čas že potekel? Če je, končaj igro.