

## P O G L A V J E 6

READ, DATA, RESTORE

Kratka vsebina: READ, DATA, RESTORE

V nekaterih prejšnjih programih smo videli, da lahko informacije in podatki direktno prihajajo v kompjuter z uporabo ukaza INPUT. To je lahko zamudno, posebno še, če se ponavlja množica podatkov, ko program deluje. Z uporabo ukazov READ, DATA in RESTORE lahko prihranite precej šasa. Na primer:

```
1Ø READ a, b, c
2Ø PRINT a, b, c
3Ø DATA 1Ø, 2Ø, 3Ø
4Ø STOP
```

Ukaz READ je sestavljen iz READ, ki mu sledi seznam imen variabel, medsebojno ločenih z vejicami. Ta ukaz deluje podobno kot INPUT, le da morate vtipkati vrednosti variabel; računalnik zahteva te vrednosti v ukazu DATA.

Vsak ukaz DATA je seznam številčnih ali string izrazov (zaporedje znakov), ki jih deli vejica. Te izraze računalnik zanemarja (razen pri izvrševanju READ-a), zato jih lahko postavite kamorkoli v program. Zamislite si izraze, zbrane za vsemi ukazi READ, ki tvorijo dolg seznam: DATA seznam. Ko se računalnik pripravlja, da bo prvič prebral (READ) vrednosti, vzame najprej prvi drugi, tretji, ... izraz ter preko ukaza READ prehaja skozi seznam podatke (DATA seznam). (Če ga poskušate obiti, da napako). Vstavljanje ukazov DATA v direktno komando je izguba časa, ker jih READ ne bo našel. Ukazi DATA morajo iti v program.

Poglejmo, kako ti ukazi odgovarjajo eni drugim v programu, ki ste ga ravnokar vtipkali. Vrsta 1Ø pove računalniku, naj prečita tri podatke ter jim pridruži variable a, b in c. Vrsta 2Ø ukaže PRINT (izpiši) te podatke. Ukaz DATA v tretji vrsti, da vrednost za a, b in c. Vrsta 4Ø program ustavi. Da bi videli vrstni red, s katerim se vse to odvija, spremenite vrsto 2Ø v: