

```
4Ø NEXT m
```

```
5Ø PRINT
```

```
6Ø NEXT n
```

}  
} n-zanka

Od dveh FOR-NEXT zank, se mora ena nahajati v drugi ali pa morata biti popolnoma ločeni.

Še eno stvar morate preprečiti: skakanja v FOR-NEXT zanko od zunaj. Kontrolna variabla je postavljena pravilno šele po izvršitvi ukaza FOR, če pa tega ni, bo ukaz NEXT računalnik zmedel. Verjetno boste dobili sporočilo o napaki, ki pravi: NEXT without FOR (NEXT brez FOR) ali pa: variable not found (variabla ni najdena).

FOR in NEXT lahko uporabljate tudi v direktnih ukazih. Poizkusite:

```
FOR m = Ø TO 1Ø; PRINT m: NEXT m
```

Včasih lahko to uporabite tudi kot način (čeprav umeten), da preprečite omejitve v ukazu GO TO, ker ukaz nima rednega števila. Na primer:

```
FOR m = Ø TO 1 STEP Ø; INPUT a: PRINT a: NEXT m
```

S korakom "nič" dosežemo, da se ta ukaz stalno ponavlja.

To pravzaprav ni priporočljivo, saj v primeru, da se pojavi napaka, izgubimo ukaz, ki ga moramo ponovno vtiskati, CONTINUE pa ne bo deloval.

#### VAJE

- 1) Kontrolna variabla ima poleg imena in vrednosti tudi mejno vrednost, korak in navodilo za ukaz, ki sledi ukazu FOR.
- 2) Pustite, da dela tretji program (tretji od začetka poglavja) ter nato vtipkajte:

```
PRINT z
```

Zakaj je odgovor 6 in ne 5?

Odgovor:

Ukaz NEXT v vrsti 6Ø je izvršen petkrat in vsakokrat je povečan z za 1. Zadnjikrat ima z vrednost 6, nakar ukaz NEXT odloči, da se ne vrne tem-