

stringa "2"+"* 3" ter je ocenjen tako, da da string "2 * 3". Nato se brišejo dvojni narekovaji stringa, to kar ostane pa se tretira kot številka. Tako $2 * 3$ daje številko 6.

V kolikor niste zbrani, vas lahko vse to zmede - na primer:

```
PRINT VAL "VAL""VAL""2""
```

(Zapomnite si, da morajo biti narekovaji izpisani dvojno, če želite postaviti kaj v narekovaje znotraj stringa, če pa greste še globje v string, pa morajo biti narekovaji štirojni ali celo osmerni).

Obstaja še ena funkcija, ki je zelo podobna VAL-u, čeprav manj uporabna, imenuje pa se VAL\$. Njen argument je še vedno string, vendar pa je tudi njen rezultat string. Da bi videli kako to deluje, se spomnite, kako se VAL razdvaja v dva koraka; najprej se njen argument oceni kot string, nato se brišejo narekovaji, ostanek pa se tretira kot številka. Pri VAL\$ je prvi korak isti, vendar se po brisanju narekovajev v drugem koraku, vse kar ostane tretira kot še en string:

```
VAL$ "" Sadni punč "" = " Sadni punč "
```

(Bodite pozorni na ponovno širjenje stringovih narekovajev!)

Napišite:

```
LET a$ = "99"
```

ter izvršite naslednje: VAL a\$, VAL "a\$", VAL ""a\$"", VAL\$a\$, VALS"a\$" in VAL\$""a\$"". Nekaj od tega bo uspelo, nekaj pa ne: poizkusite pojasniti vse odgovore. (ostanite zbrani)

SNG je funkcija signum (funkcija predznaka). To je do sedaj prva funkcija, ki s stringi nima veze, saj sta rezultat in argument števili. Če je argument pozitiven, bo rezultat +1, če pa je negativen, bo rezultat -1, argument 0 pa pomeni, da je rezultat enak nuli.

Tudi ABS je funkcija, pri kateri sta argument in rezultat števili. Ta funkcija pretvarja argument v pozitivno število (ki je rezultat), ne glede na predznak. Primer:

```
ABS - 3.2 = ABS 3.2 = 3.2
```

INT vzame cel del (to je celo število), ki je eventualno negativen. Ta funkcija pretvarja pozitivno decimalno število v celo število, del za decimalno ve-