

jico (piko) pa odvrže. Primer:

INT 3.9 = 4

SQR izračunava kvadratni koren, rezultat pa je število, ki pomnoženo s samim seboj daje argument. Primer:

SQR 4 = 2      zato ker je  $2 * 2 = 4$

SQR 0.25 = 0.5      zato ker je  $0.5 * 0.5 = 0.25$

SQR 2 = 1.4142136 (približno) zato ker je  $1.4142136 * 1.4142136 = 2.000001$

Če množite katerokoli število (lahko tudi negativno) s samim seboj, je odgovor vedno pozitiven. To pomeni, da kvadratni koreni nimajo negativnih števil, zato dobite v primeru, če uporabite SQR za negativni argument, obvestilo o napaki: An invalid Argument.

Po želji lahko definirate tudi vaše osebne funkcije. Možna imena za to so: FN, za katerim se nahaja črka (če je rezultat število) ali pa FN za katerim se nahaja črka in oznaka \$ (če je rezultat string).

Funkcijo definirajte z ukazom DEF nekje v programu. Primer definicije funkcije FN s, ki ima za rezultati kvadrat argumenta je:

```
10 DEF s(x)=x * x: REM kvadrat od x
```

Ukaz DEF izdajte šele, ko ste računalniku določili E-mod (razširjeni mod), s pritiskom na SYMBOL SHIFT skupaj z 1. Ko to vtipkate, vam bo računalnik avtomatsko dal FN, saj za DEF vedno prihaja FN. Nato se bo ime funkcije dopolnilo z: FN s. X v oklepajih je ime, ki ga želimo dodeliti argumentu funkcije. Za to lahko uporabite črko (ali črko z \$, če je argument string).

Za znakom = prihaja prava definicija funkcije. To je lahko katerikoli izraz, ki se nanaša na argument ter pri tem uporablja ime, ki ste mu ga določili

(v tem primeru x), kot tudi katerokoli navadno variabla. Ko ste vpisali vrsto s katero definirate funkcijo, jo lahko uporabite podobno kot ostale lastne funkcije računalnika, izpisujoč njeno ime (FN s), ki mu sledi argument.

Zapomnite si, da se mora argument nahajati v oklepaju, če ste funkcijo definirali samo. Poizkusite to:

```
PRINT FN s(2)
```

```
PRINT FN s(3+4)
```

```
PRINT 1+INT FN s(LEN "Slon"/2+3)
```