

in

$$a \uparrow (b * c) = (a \uparrow b) \uparrow c$$

Če se s tem srečujete prvič, se ne trudite, da bi si vse zapomnili; zapomnite si samo naslednje:

$$a \uparrow (-1) = 1/a \quad \text{in}$$

$$a \uparrow (1/2) = \text{SQR } a$$

in ko boste razumeli, bo tudi ostalo mnogo lažje. Eksperimentirajte z vsem tem tako, da poizkusite naslednji program:

```
10 INPUT a, b, c
20 PRINT a ↑ (b+c), a ↑ b * a ↑ c
30 GO TO 10
```

Če pravilo, ki smo vam ga prej podali drži, potem bosta vsakih (ko bo računalnik izpisova), dve števili enaki. (Pazite: zaradi načina, s katerim računalnik izračunava \uparrow , mora biti število na levi strani vedno pozitivno - v našem primeru je to a).

Tipičen primer uporabe funkcije je obrestni račun.

Vzemimo na primer, da imate na banki nekaj denarja, za katerega dobite letno 15 % obresti. Po letu dni, se bo vaša vloga povečala za 15 % in bo znašala 115 % (vsoto ste pomnožili s 1.15). Naslednje leto se bo zgodilo isto, zato sedaj vsota znaša: $1.15 * 1.15 = 1.15 \uparrow 2 = 1.3225$ krat vložena vsota. Torej, po y letih boste imeli 1.15^y krat vložena vsota. Če sedaj poizkusite:

```
FOR y=0 TO 100: PRINT y, 10 * 1.15 ↑ y: NEXT y
```

boste videli, da se vsota viša vse hitreje in hitreje (čeprav boste opažali, da ne drži koraka z inflacijo).

Takšna reakcija, kjer se neka vsota po določenem časovnem intervalu, množi sama s seboj po določenem ključu, se imenuje eksponencialna rast in se izračunava z dviganjem nekega števila na potenco časa. Predstavljajmo si, da ste napravili naslednje:

```
10 DEF FN a(x) = a ↑ x
```

Tu je a več ali manj fiksiran z ukazom LET; njegova vrednost odgovarja ob-