

Ko ste vstavili v program odgovarjajoč ukaz DEF, lahko v izrazih uporabljate svoje lastne funkcije enako svobodno kot računalnikove.

Pozor: v nekaterih dijalektih BASIC-a morate postaviti v oklepaje celo argumente računalnikovih funkcij, kar pa ni v BASIC-u, ki ga uporablja ZX Spectrum.

ITN vedno zaokroža "navzdol". Da bi zaokrožili na najbližje celo število, številu dodajte 0.5. Da bi to storili, sedaj napišite svojo lastno funkcijo:
 20 DEF FN r(x)=ITN (x+0.5):REM da celo število, ki je najbližje številu x.
 Sedaj boste dobili npr.:

```
FN r(2.9)=3          FN r(2.4)=2
FN r(-2.9)=-3       FN r(-2.4)=-2
```

Primerjajte z odgovori, ki jih dobite, če koristite samo INT (namesto FN r).
 Vtipkajte in izvršite naslednji program:

```
10 LET x=0: LET y=0: LET a=10
20 DEF FN p(x,y)=a+x*y
30 DEF FN q( )=a+x*y
40 PRINT FN p(2,3),FNq( )
```

V tem programu imamo precej suptilnih dejstev. Prvo, funkcija ni omejena samo na en argument - lahko jih ima več ali pa nobenega, vendar mora oklepaje zadržati. Drugo, ni važno kam v program postavite ukaz DEF. Ko je računalnik izvršil vrsto 10, on enostavno preskoči vrsti 20 in 30, da bi prišel do vrste 40. Vseeno pa ukaz DEF ne more biti v direktnem ukazu, temveč se mora nahajati nekje v programu. Tretje, x in y so imena variabel v programu kot celoti, so pa tudi argumenti funkcije FN p. FN p začasno pozabi na variable x in y (iz vrste 10) ter glede na to, da nima argumenta z imenom a, uporabi variabla a (iz vrste 10). Medtem, ko je FN p (2,3) ocenjevano, je za a uporabljena vrednost 10, ker je to variabla; za x je vzeta vrednost 2 (ker je to prvo argument), za y pa vrednost 3 (ker je to drugi argument). Rezultat je torej $10+2*3=16$. Ko je ocenjevana FN q(), ki nima argumenta, so a, x in y prevedeni kot variable z vrednostmi 10, 0 in 0 (respektivno). Rešitev je v tem primeru $10+10*0=10$. Sedaj spremenite vrsto 20 v:

```
20 DEF FN p(x,y) = FN q( )
```