Isti program lahko zapišemo z uporabo FOR zanke:

```
1ØØ FOR L = 1 TO 1Ø
11Ø PRINT L
12Ø NEXT L
13Ø END
```

Kot vidimo iz primera je z uporabo FOR zanke program krajši ter lažje razumljiv.
Kadar izvajamo FOR stavek, se izvrši nekaj operacij. Začetna vrednost postane vrednost spremenljivke števca, ko pridemo nato do NEXT stavka, se vrednost koraka doda vrednosti spremenljivke števca. Če STEP stavka ni, potem je ta korak vedno + 1. Sedaj se vrednost spremenljivke primerja s končno vrednostjo, in če le-ta še ni dosežena, se izvajanje zanke nadaljuje. V primeru, ko vrednost spremenljivke doseže končno vrednost, se zanka zaključi in program se nadaljuje za NEXT stavkom. Če je vrednost koraka pozitivna, mora spremenljivka doseči ali preseči končno vrednost, kadar pa je negativna, pa mora postati enaka ali manjša od končne vrednosti.

Primeri:

```
1ØØ FOR L = 1ØØ TO Ø STEP - 1
1ØØ FOR L = PI TO 6 * PI STEP .Ø1
1ØØ FOR AA = 3 TO 3
```

FRE

Tip: Funkcija
Format: FRE (< spremenljivka >)

Delovanje: FRE funkcija nam pokaže, koliko RAM spomina je še na voljo za program in spremenljivke. Če poskuša program porabiti več spomina, kot ga je še na razpolago, se pojavi OUT OF MEMORY napaka. Število v oklepaju ima lahko kakršnokoli vrednost in se ne uporablja pri računanju.

POZOR! Če je rezultat FRE funkcije negativen, mu je potrebno prišteti 65536, tako da dobimo pravilno število bytov, ki so še na voljo.