

ko zanka doseže NEXT stavek, števec naraste za vrednost koraka (STEP) v zanki; v primeru ko je zadnja vrednost (zanke) že dosežena, pa NEXT zaključi zanko in program se nadaljuje z naslednjim stavkom.

Primeri: 1Ø FOR J = 1 TO 5: FOR K = 10 TO 2Ø: FOR N = 5 TO
- 5 STEP - 1

2Ø NEXT N, K, J

(Zaključujemo zanke, ki so druga v drugi)

1Ø FOR L = 1 TO 1ØØ

2Ø FOR M = 1 TO 1Ø

3Ø NEXT M

4Ø NEXT L

(Zanke ne smejo križati druga drugo)

1Ø FOR A = 1 TO 1Ø

2Ø FOR B = 1 TO 2Ø

3Ø NEXT

4Ø NEXT

(Imena spremenljivk v NEXT stavkih ne potrebujemo.)

NOT

Tip: Logični operator

Format: NOT <izraz>

Delovanje: NOT je logični komplement vrednosti vsakega bita ter nam izračuna dvojiški komplement. Če uporabljamo floating-point števila, se operandi najprej pretvorijo v integer, decimalni del je v tem primeru izgubljen. Operator NOT uporabljamo tudi v primerjavah pri testiranju pravilne/neppravilne vrednosti izraza.

Primeri: 1Ø IF NOT AA = BB AND NOT (BB = CC) THEN ...

NN% = NOT 96: PRINT NN%

- 97

POZOR! Za izračun vrednosti NOT uporabljamo izraz $X = -(X+1)$.
(Dvojiški komplement kateregakoli integer števila je komplement bita plus 1.)