

**bc** [ **-hlwsqv** ] [ **long-options** ] [ *file ...* ]

**-h, --help**

Kiírja a használatát és kilép.

**-l, --mathlib**

A matematikai könyvtárat betölti és 20-ra állítja a scale-t.

**Számok:** Tetszőleges pontosságú decimális.

**length** a szám szignifikáns jegyeinek száma.

**scale** a tizedespont utáni jegyek száma.

**Változók:** egyszerű és tömb. Neve: kisbetűvel kezdődik, majd akárhány kisbetű, szám vagy aláhúzás. A tömbnév után [ ] jön. Négy speciális változó:

**scale** kiírt pontosság beállítása (0, **bc -1** esetén 20)

**ibase** input számrendszer alapszáma (2 - 16)

**obase** output szr. alapszáma (2 - BC\_BASE\_MAX)

**last** vagy . az előző eredmény

**Megjegyzés (comment):**

*/\** megjegyzés *\*/* vagy **#**-től a sor végéig.

**Kifejezések:**

**var** egyszerű vagy tömbváltozó: name, name[expr]

*expr* kifejezés

a kifejezés scale értéke a benne szereplő legnagyobbé.

**- expr**

ellentettje

**++ var**

megnöveli eggyel, és ez a kifejezés értéke

**-- var**

csökkenti eggyel, és ez a kifejezés értéke

**var ++**

a kifejezés értéke a változó értéke, majd növeli a változó értékét eggyel

**var --**

a kifejezés értéke a változó értéke, majd csökkenti a változó értékét eggyel

**expr + expr**

**expr - expr**

**expr \* expr**

**expr / expr**

**expr % expr**

maradék: ha  $scale=0$ , és  $a$  és  $b$  egészek, akkor  $a\%b$  a szokásos maradék. Egyébként  $a\%b$  először  $a/b$  scale jegy pontosan,  $a-(a/b)*b$  a  $scale+scale(b)$  és  $scale(a)$  maximuma pontossággal

**expr ^ expr**

hatványozás, a második kifejezés egész.  
 $(scale(a^b) = \min(scale(a)*b, \max(scale, scale(a)))$   
 $expr^0$  mindig 1.

**( expr )**

precedencia megváltoztatása

**var = expr**

**var <op>= expr**

ugyanaz, mint  $var = var <op> expr$   
pl.  $x += 3$  ugyanaz, mint  $x = x + 3$

**Relációk:** két kifejezés közé rakható relációjelek

**<, >, <=, >=, ==, !=**

**Logikai kifejezések:** 0, ha hamis, 1 (inputnál  $\neq 0$ ), ha igaz.

**!expr** tagadás

**expr && expr** logikai és

**expr || expr** logikai vagy

**Precendencia:** BA, JA, NA = bal, jobb, illetve nem asszociatív, op = operátor

**|| op, BA > && op, BA > ! op, NA >** relációk, **BA >** értékadó op, **JA > +** és **- op, BA > \*, /** és **% op, BA > ^ op, JA >** unér **- op, NA > ++** és **-- op, NA**

**Speciális kifejezések:**

**length ( expr )** a kif. jegyeinek száma

**read ( )** beolvas egy számot

**scale ( expr )** tizedesjegyek száma

**sqrt ( expr )** négyzetgyök

**Utasítások:** Utasítás elválasztók a **;** és az újsor. Sortörés **\n** karakterrel.

**expression**

Ha értékadó elvégzi, egyébként kiértékeli kifejezést és kiírja az értékét. A hosszú sorok megtörését **"** jelzi.

**string**

Karakterlánc, **'** jelek közt.

**print list**

A **print** utasítás kiírja a vesszővel elválasztott kifejezések kiértékelt listáját a megadott sorrendben. Speciális karakterek **\**-vel kezdődnek. A **bc** ismeri a **"\a"** (alert or bell), **"\b"** (backspace), **"\f"** (form feed), **"\n"** (newline), **"\r"** (carriage return), **"\q"** (double quote), **"\t"** (tab), and **"\e"** (backslash).

**{ statement\_list }**

Utasításlista, egy utasításnak tekintti.

**if ( expression ) statement1 [else statement2]**

Ha **expression** nem 0, **statement1**, ha 0, **statement2** hajtódik végre.

**while ( expression ) statement**

vége, ha **expression** értéke a kiértékeléskor 0, vagy ha végrehajtódik egy **break** utasítás.

**for ( [expr1] ; [expr2] ; [expr3] ) statement**

A legelső ciklus előtt végrehajtódik **expr1**, minden ciklus elején **expr2**, és mindegyik végén **expr3**. Kiugrik a ciklusból, ha **expr2** 0 (hamis), vagy ha végrehajt egy **break** utasítást. A 3 kifejezés bármelyike elhagyható. Ekvivalens:

**expr1; while (expr2) { statement; expr3; }**

**break**

**continue**

A legutolsó **for** ciklusban a következő iteráció.

**halt**

Leállítja a **bc**-t, ha végrehajtódik.

**if ( 0 == 1 ) halt** nem állítja le.

**return**

0-t ad vissza.

**return ( expression )**

**expression** értékét visszaszítja, a **( )** nem kell.

**Függvények:**

**define name ( params ) { auto\_list statement\_list }**

**name** a függvény neve

**params** 0 vagy több paraméter vesszővel elválasztva, **auto\_list** opcionális **auto name1, name2...**; parancs a lokális változók jelölésére,

**statement\_list** utasításlista, a nyitó kapcsos előtt vagy után is lehet sort törni. A fv hívása: **name ( params )**

**Matematikai könyvtári függvények:**

**s ( x ), c ( x ), a ( x ), l ( x ), e ( x ), j ( n, x )**  
**sin ( x ) cos ( x ) arctg ( x ) ln ( x ) exp ( x )** Bessel-fv