Beispiel eines RAM-Programms: Multiplikation zweier Zahlen

Die beiden Eingabewerte, die in `c(1)` und `c(2)` gespeichert sind, sollen multipliziert werden, und das Resultat soll in `c(1)` gespeichert werden. Dies geschieht durch `c(1)`-faches Aufaddieren des Wertes von `c(2)` (oder umgekehrt, s.u.).

```
// Zeilen 1-15: Wir wollen, dass in Register 1 ein positiver Multiplikand steht.
// Darum: Ist c(1) negativ, so werden c(1) und c(2) vertauscht (Zeilen 3-7).
// Ist dann immer noch eine negative Zahl in Register 1, so werden beide Argumente
//  negiert (Zeilen 10-15).
01: LOAD 1            // Lade Operand in Register 1
02: IF >=0 GOTO 16    // Falls c(1) positiv ist, fahre fort, sonst:
03: STORE 3           // vertausche c(1) und c(2)
04: LOAD 2
05: STORE 1
06: LOAD 3
07: STORE 2
08: LOAD 1            // Lade Register 1 nochmals
09: IF >=0 GOTO 16    // Falls c(1) jetzt positiv, fahre fort, sonst:
10: LOAD #0           // negiere Register 1
11: SUB 1
12: STORE 1
13: LOAD #0           // negiere Register 2
14: SUB 2
15: STORE 2
// Addiere den Inhalt von Register 2 so oft auf, wie der Wert von Register 1
// angibt (c(1) ist also Schleifenzaehler). Das Zwischenergebnis liegt in
// Register 3.
16: LOAD #0          // loesche Register 3 (Zwischenergebnis)
17: STORE 3
18: LOAD 1           // Hauptschleife
19: IF =0 GOTO 26
20: SUB #1                // dekrementiere Schleifenzaehler
21: STORE 1
22: LOAD 3                // c(3):=c(3)+c(2)
23: ADD 2
24: STORE 3
25: GOTO 18
// Speichere das Ergebnis in Register 1 (wo es erwartet wird)
26: LOAD 3            // Resultat aus c(3) in c(1) kopieren
27: STORE 1
28: END
```