



Core War³²

The Ultimate Programming Game

A REDCODE Cheat Sheet for Coders

At DEF CON 24. CoreWar designed by A. K. Dewdney. Contest managed by Leo Tindall.

Special thanks to the PMARS team, <http://corewars.co.uk>, and reddit's [/r/corewars](https://www.reddit.com/r/corewars).

Instructions **ADD.I #4, 3**

Every Core Warrior is just a collection of instructions. These instructions are turned into "machine code", loaded into virtual memory, and executed. Each instruction has an **A-field** and a **B-field**. In the ADD example above, the A-field has a literal number 4 (hence the #) and the B-field has a relative address of +3. The .I is a **modifier suffix** (see below). I is the default (ADD #4, 3 is the same as ADD.I #4, 3).

<p>DAT: data (kills the process if executed; use for storage) MOV: move (copies data from one address to another) ADD: add (adds one number to another) SUB: subtract (subtracts one number from another) MUL: multiply (multiplies one number with another) DIV: divide (divides one number with another) MOD: modulus (divides one number with another and gives the remainder) JMP: jump (continues execution from another address) JMZ: jump if zero (tests a number and jumps if it's 0) JMN: jump if not zero (tests a number and jumps if it isn't 0) DJN: decrement and jump if not zero SPL: split (starts a second process at another address) CMP: compare (same as SEQ) SEQ: skip if equal (compares 2 instructions, skips the next instruction if they are equal) SNE: skip if not equal (compares 2 instructions, skips the next instruction if they aren't equal) SLT: skip if lower than (compares 2 values, and skips the next instruction if first < second) LDP: load from p-space (loads a number from private storage space) STP: save to p-space (saves a number to private storage space) NOP: no operation (does nothing)</p>

<p>Addressing Modes # Immediate \$ Direct (default) * A-indirect @ B-indirect { A-predecrement < B-predecrement } A-postdecrement > B-postdecrement</p>
--

<p>Suffix Mods MOV.A: A -> A MOV.B: B -> B MOV.AB: A -> B MOV.BA: B -> A MOV.F: AB -> AB MOV.X: AB -> BA MOV.I: Whole Instr</p>
--

<p>DAT with 1 arg (DAT 1) defaults to using the B-field.</p>
--

Example

```
; Semicolon starts a comment
; name, author, and strategy are just metadata.
; assert comments let the game ensure that warriors that won't work, aren't executed.
; name Dwarf
; author A.K. Dewdney
; strategy Bombs the core at regular intervals.
; assert CORESIZE % 4 == 0

org loop ; loop is a label (see loop: on the next line). org means, start at the given label

loop:  add.ab #4, bomb ; labels get replaced by their location
        ; this is add.ab #4, 3
        mov.i bomb, @bomb ; move the whole bomb instruction (.i)
        ; to @bomb (the location pointed at by the B-field of bomb)
        jmp     loop
bomb:  dat     #0, #0
```

Resources

Submit your warriors at <https://corewar.silverwingedseraph.net>

For more info on REDCODE, check out <http://vyznev.net/corewar/guide.html>

To learn more about Core War in general, and to compete with people around the world, check out <http://koth.org> and <http://corewar.co.uk>