Type		Meaning	Position in Source Line	Description Method	Example

Comment		Treats everything after as a comment.	Anywhere	`;………`
				`//………`
					`; This is a comment.`
					`// This is a comment,too.`
					`MAIN // Main Routine`
					`    ldm 0xa  ; A=0xa`
					`    xch 0    // exchange`

Number		Numeric Value	Within expressions or standalone	Values are expressed in either decimal or hexadecimal format. Hexadecimal values are prefixed with 0x. The letters A–F in hexadecimal may be written in either uppercase or lowercase. In decimal notation, adding a minus sign (-) at the beginning indicates a negative number. For hexadecimal values, if the most significant bit (MSB) is 1, the value is treated as negative.	`10`
					`-5 ; negative`
					`0xc`
					`0xAB // negative`
					`0x0AB // positive`

Expression		Arithmetic Expression	Within label assignment expressions or within literals	Arithmetic expressions involving numbers and labels.
				You can use the four standard arithmetic operators: +, -, *, and /. Parentheses () are also supported for grouping expressions.
					`10*20`
					`(1+2)*3`
					`LABEL+8`

Label		A label is assigned the starting ROM address corresponding to its line. Labels are referenced as branch destinations or within expressions. Uppercase and lowercase letters in labels are distinguished—they are case-sensitive. Label values fall within the ROM address space, ranging from 0x000 to 0x3FF.	At beginning of line	Labels are represented by non-numeric character strings, and must be followed by a delimiter—either a space, tab, comma, or colon. Reserved keywords (instruction mnemonics and pseudo-instruction mnemonics) cannot be used as labels. Lines containing only a label are also permitted.	`RESET`
					`     ldm 0x4`
					`LOOP jcn tz LOOP`
					`     add 14`

Literal (Constant)		Stores specified literals (constants) starting from the beginning ROM address corresponding to the line. Each literal constant is 8 bits wide.	Start after inserting one or more spaces or tabs at the beginning of the line.	Expressions or numeric values are listed, separated by spaces, tabs, or commas.	`TABLE`
					`    123`
					`    0xab 0xcd`
					`    (1+2)*3,0x12,0x23`
					`    TABLE+4 255`

Pseudo Instruction	Assignment of numeric values to labels	Assign a numeric value to a label using the expression on the right-hand side. Be aware that the valid range of a label's value depends on how it is referenced—whether in a literal or as part of an instruction.	Begin at the start of the line, including the label.	`Label=Expression`
				`Label equ Expression`
				`Label EQU Expression`
					`CONST0=0x12`
					`CONST1 equ CONST0*11`

	Origin (Start Address)	Directly specify the starting ROM address for the next line.	Begin after inserting one or more spaces or tabs at the start of the line.	`  org ROM Address`
				`  ORG ROM Address`
				
				`or`
				
				`(without any label)`
				`  = ROMAddress`
				`  equ ROM Address`
				`  EQU ROM Address`
					`    org 0x000`
					`RESET jun MAIN`
					…
					`    =0x200`
					`MAIN ldm 0xa`
					…
					`    equ 0x300`
					`FUNC ldm 0xb`

	End of Assemble	Ends assembly process at beginning of line	Begin after inserting one or more spaces or tabs at the start of the line.	`  end`
				
				`  END`
					`LOOP`
					…
					`    jun LOOP`
					`    end`

CPU Instructions		Follow the CPU instruction table. Instruction mnemonics may be written in either uppercase or lowercase.			