

```

0096 04
97 BA PHI RA ;Set RA = first byte Source Listing
98 F8 LDI
99 00
9A AA PLO RA
9B 0A LDN RA ;Get first character of line;begin continue
9C FB XRI ;2nd pass
9D 3B ;Test if = ";"
9E 32 BZ ;If so, branch to Carriage Return @ 00EB
9F EB ;(Comment lines-ignored) do not increment R9-
00A0 D4 SEP R4 ;Object Pointer--Do not test for overflow
A1 02 ;Call Point to Instruction
A2 A6
A3 0A LDN RA ;Get first character instruction
A4 FE XRI
A5 20 ;= 20 (ASCII space)?
A6 32 BZ ;If so, branch to Carriage Return @ 00EB
A7 EB ;(Space lines-ignored)
A8 0A LDN RA
A9 FB XRI
AA 52 ;= "R" for Reserve Memory bytes
AB 3A BNZ ;If not, continue
AC B4
AD 1A INC RA
AE 1A INC RA ;Point to # bytes after the "RM"
AF D4 SEP R4
00B0 02 ;Call Reserve Memory - R9 incremented
B1 42 ;To reserve bytes needed
B2 30 BN ;Branch to Carriage Return
B3 EB ;(No instruction to output)
B4 D4 SEP R4
B5 01 ;Call Put Instruction in Memory
B6 00
B7 4A LDA RA ;Get byte just after instruction (RA+1)
B8 FB XRI
B9 3B ;= ";"
BA 32 BZ ;If so, branch to set R9 to next Object
BB DE ;Code slot & Carriage Return (Ignore comments)
BC 0A LDN RA ;Else get first byte of a possible label
BD FB XRI
BE 20 ;= "space"?
BF 32 BZ ;If so, branch to set R9 to next Object
00C0 DE ;Code slot & Carriage Return (Ignore spaces)
C1 93 GHI R3 ;(Get current PC page address (=00))
C2 BD PHI RD ;→RD.1
C3 F8 LDI
C4 CB ;Load address byte string ("Link") @ 00CB
C5 AD PLO RD ;RD points to "link" string stored @ 00CB
C6 D4 SEP R4

```