```
014E    0F                  ;Strip first 4 bits to assure "0-N" form
 4F     F1      OR          ;"OR" with top of stack to combine byte
0150    59      STR    R9   ;Store back via R9
 51     19      INC    R9   ;R9 + 1 -2nd half Chip-8 Instruction
 52     0D      LDN    RD   ;Get low part address in table
 53     59      STR    R9   ;Store as 2nd half Chip-8 Instruction
 54     29      DEC    R9   ;Reset to point to same Object Code
 55     D5      SEP    R5   ;Return
```

                    SYMBOL TABLE REARRANGER

```
0156    D4      SEP    R4
 57     01                  ;Call Clear Text
 58     CA
 59     F8      LDI
 5A     04
 5B     BA      PHI    RA
 5C     F8      LDI
 5D     00
 5E     AA      PLO    RA   ;RA = 0400 Text
 5F     F8      LDI
0160    40
 61     52      STR    R2   ;Push ASCII"@" (one less than "A")
 62     F8      LDI
 63     0A
 64     BD      PHI    RD
 65     F8      LDI
 66     00
 67     AD      PLO    RD   ;RD = 0A00 Table
 68     02      LDN    R2
 69     FC      ADI
 6A     01
 6B     52      STR    R2   ;ASCII on stack + 01
 6C     FB      XRI
 6D     5B
 6E     3A      BNZ
 6F     71
0170    D5      SEP    R5   ;Return after ASCII = 5B "["
 71     0D      LDN    RD
 72     32      BZ
 73     62                  ;Check next letter after table searched
 74     E2      SEX    2    ;X = 2
 75     F3      XOR         ;Else compare with stack
 76     3A      BNZ
 77     7F                  ;Branch to check next in table if ≠
 78     22      DEC    R2   ;Preserve stack
 79     D4      SEP    R4
 7A     01                  ;Call Output Symbol to Text
 7B     89
```