```
                        CONVERT/STORE ASCII IN TEXT

021F   9F   GHI  RF    ;Get byte for conversion passed in RF.1
0220   F6   SHR         ;Shift first digit to right
  21   F6   SHR         ;  "      "      "    "    "
  22   F6   SHR         ;  "      "      "    "    "
  23   F6   SHR         ;  "      "      "    "    "
  24   AE   PLO  RE    ;Put in RE.0 for passing to sub
  25   D4   SEP  R4
  26   02               ;Call ASCII conversion (Returns ASCII
  27   11               ;In RE.0)
  28   8E   GLO  RE    ;Get the ASCII conversion
  29   59   STR  R9    ;Store in text at R9 (First digit)
  2A   19   INC  R9    ;R9 + 01
  2B   9F   GHI  RF    ;Get the same byte for conversion
  2C   FA   ANI        ;"AND" with 0F for second digit
  2D   0F
  2E   AE   PLO  RE    ;Put in RE.0 for passing to sub
  2F   D4   SEP  R4
0230   02               ;Call ASCII conversion
  31   11
  32   8E   GLO  RE    ;Get the ASCII conversion
  33   59   STR  R9    ;Store in text at R9 (Second digit)
  34   19   INC  R9    ;R9 + 01
  35   D5   SEP  R5    ;Return

                        INDEX RC TO ASCII STRING

0236   F8   LDI
  37   08
  38   BC   PHI  RC
  39   F8   LDI         ;Initialize RC to point to
  3A   00
  3B   AC   PLO  RC    ;Beginning mnemonic table @ 0800
  3C   E2   SEX  2     ;X = 2
  3D   97   GHI  R7    ;Get the decoded instruction in R7.1
  3E   52   STR  R2    ;Push for upcoming comparison
  3F   4C   LDA  RC    ;Get byte from a table entry
0240   F3   XOR         ;Compare with byte on stack
  41   3A   BNZ
  42   44               ;If not equal, continue
  43   D5   SEP  R5    ;Else return -- RC points to instruction mnemonic
  44   4C   LDA  RC
  45   3A   BNZ         ;Increment RC to next table entry by
  46   44               ;Advancing, testing for null (00) byte
  47   9C   GHI  RC
  48   FB   XRI         ;Test if RC went too far -- past table end
  49   0A               ;(00 byte at 09FF prevents runaway search)
  4A   3A   BNZ
```