```
02D2   8E   GLO   RE    ;Get the converted byte
  D3   59   STR   R9    ;Store in text for display
  D4   B8   PHI   R8    ;Also put in R8.1 for passing to sub
  D5   D4   SEP   R4
  D6   03               ;Call Display Character @ R7
  D7   6C
  D8   19   INC   R9    ;Point to next character slot
  D9   89   GLO   R9
  DA   F6   SHR         ;Shift R9.0 right to test if even or odd
  DB   33   BDF
  DC   DE               ;If odd, (DF=01), skip the INC R7 instruction
  DD   17   INC   R7    ;R7 + 1 - Next display byte (there are 2 char. per
  DE   2B   DEC   RB    ;Loop count - 1                             byte)
  DF   8B   GLO   RB
02E0   3A   BNZ         ;Loop till all entries processed and displayed
  E1   C6               ;All entries on stack ending with "FF" stop  byte
  E2   E2   SEX   2     ;X = 2
  E3   F8   LDI         ;Set RF.1 (answer) = 00
  E4   00               ; "   "     "     "  "
  E5   BF   PHI   RF    ; "   "     "     "  "
  E6   60   IRX
  E7   9F   GHI   RF
  E8   AF   PLO   RF    ;Transfer contents RF.1 into RF.0
  E9   72   LDXA        ;Pop and advance pointer (2nd ½ byte)
  EA   BF   PHI   RF    ;Store in RF.1
  EB   F0   LDX         ;Pop next value (first ½ byte)
  EC   FE   SHL         ;Shift left X 4 so digit in left position
  ED   FE   SHL
  EE   FE   SHL
  EF   FE   SHL
02F0   52   STR   R2    ;Re-stack shifted byte
  F1   9F   GHI   RF
  F2   F1   OR          ;OR existing RF.1 with byte on stack in
  F3   BF   PHI   RF    ;Order to combine
  F4   60   IRX
  F5   F0   LDX         ;Test next byte on stack for FF stop byte
  F6   FB   XRI
  F7   FF
  F8   3A   BNZ         ;If not detected, loop for next byte data
  F9   E7
  FA   D5   SEP   R5    ;Else return, stack "right", byte(s) in RF
```