```
020E   2F   DEC   RF   ;Decrement Loop Count
 0F    8F   GLO   RF
0210   3A   BNZ        ;Loop until done with 5 rows
 11    08
 12    86   GLO   R6
 13    FF   SMI        ;Subtract 2F hex from R6.0 to point to
 14    27
 15    A6   PLO   R6   ;Next byte over for successive displays
 16    D5   SEP   R5   ;Return
```

INTERRUPT ROUTINE

```
0217   7A   42   70   22   78   22   52   C4
021F   E2   F8   00   A0   9B   B0   E2   E2
0227   80   E2   E2   20   A0   E2   20   A0
022F   E2   20   A0   3C   27   98   32   3B
0237   AB   2B   8B   B8   88   32   17   7B
023F   28   30   18
```

RESERVE MEMORY

```
0242   4A   LDA   RA   ;Get first ASCII digit
 43    BE   PHI   RE   ;→ RE.1
 44    0A   LDN   RA   ;Get second ASCII digit
 45    AE   PLO   RE   ;→ RE.0 to pass to sub
 46    D4   SEP   R4
 47    02              ;Call ASCII to Hex Conversion
 48    66              ;(Answer in RF.1)
 49    9F   GHI   RF   ;Get the converted number
 4A    32   BZ        ;If = 00, branch to error
 4B    58
 4C    AF   PLO   RF   ;Else put in RF.0 for Loop Count
 4D    19   INC   R9   ;R9 + 1
 4E    2F   DEC   RF   ;Loop - 01
 4F    8F   GLO   RF
0250   3A   BNZ        ;Loop, incrementing R9 x number bytes required
 51    4D
 52    89   GLO   R9   ;Test if R9 is even or odd
 53    F6   SHR        ;By shifting right
 54    3B   BNF        ;(DF=0=Even/=1=Odd)
 55    57              ;If even, branch to exit
 56    19   INC   R9   ;Else increment R9 to make even
 57    D5   SEP   R5   ;Return
 58    F8   LDI
 59    05
 5A    AE   PLO   RE   ;Put #5 error message (Reserved 00 bytes)
 5B    D4   SEP   R4
 5C    02              ;Call Error Message - halt program
 5D    B0
 5E    D5   SEP   R5   ;Return (In case of later change in Error Routine)
```