CIS240 / CNET220 EXERCISE 7

NAME

ASCII TO HEX AND HEX TO ASCII SUBROUTINES

In this exercise you will build on the basic input and display from exercise 6. However in this exercise you will translate the ASCII characters to HEX by subtracting a fixed conversion value, then save the information to the memory storage buffer. Then, before displaying the message you will translate the stored hex number values back to ASCII one character at a time then send to the output.

The ASCII to HEX routine is given to you, however you will have to create the HEX TO ASCII routine which essentially is the reverse operation of the ASCII TO HEX routine. In performing the HEX TO ASCII operation it will be tempting to use some of the same Label names as the ASCII TO HEX routine; however label names for the time being cannot be duplicated so altering old label names for the new routine may be useful such as CONVERT_LOWER to CONVERT_LOWER1 etc..

```
.MODEL          small

ASCII_TO_HEX        PROTO
HEX_TO_ASCII        PROTO

        .STACK      200

        .DATA

;Define data variables for program
VarWord         WORD 0
VarByte         BYTE  0
VarArray        BYTE  200 DUP(?)


;
; Note add more variables if needed
;

;Define Constants

cstCR           = 0Dh
cstLF           = 0Ah
cstSPACE        = 20h
cstEOL          = 24h


        .CODE

        .STARTUP
```

```
MAIN    PROC

        MOV     AX,2000h
        MOV     DS,AX
        MOV     SI,0

WAITFORLF:
        MOV     AH,1
        INT     21H
        CMP     AL,0DH
        JE      DATAENTERED
        INVOKE          ASCII_TO_HEX
;       CMP     AH,1
;       INVOKE          ERROR_HANDLER
        MOV     [SI],AL
        INC     SI
        JMP     WAITFORLF

DATAENTERED:
        MOV     AH,2
        MOV     DL,0AH
        INT     21H

        MOV     AL,20H
        MOV     [SI],AL
        INC     SI

        MOV     AH,0
        INT     16H
        CMP     AL,'D'
        JE      DISPLAYMESSAGE
        JMP     WAITFORLF

DISPLAYMESSAGE:
        MOV     AL,24H
        MOV     [SI],AL
        MOV     SI,0            ;Point to beginning of outbuffer
CONV_LOOP:
        MOV     AL,[SI]
        CMP     AL,24H
        JE      CONV_DONE
        INVOKE          HEX_TO_ASCII
        MOV     [SI],AL
        INC     SI
        JMP     CONV_LOOP
CONV_DONE:
        MOV     AH,9H
        MOV     DX,0
        INT     21H

MAIN    ENDP
        .EXIT
```

```
;*************** SUBROUTINES *********************

;ascii to hex conversion routine
;inputs   AL input ascii character between 0 to 9, A to F, or a to f
;outputs AL, contains hex number between 0 and F
;          AH, input number out of range AH=1, otherwise AH=0
ASCII_TO_HEX  PROC
          MOV     AH,0              ;Clear return error

          CMP     AL,39H            ;test for 0 to 9
          JG      ALPHABET
          CMP     AL,30H
          JL      CONVERT_ERROR
          SUB     AL,30H
          JMP     CONVERT_DONE
ALPHABET:
          CMP     AL,46H            ;test for A to F
          JG      CONVERT_LOWER
          CMP     AL,41H
          JL      CONVERT_ERROR
          SUB     AL,37H
          JMP     CONVERT_DONE
CONVERT_LOWER:
          CMP     AL,66H            ;test for a to f
          JG      CONVERT_ERROR
          CMP     AL,61H
          JL      CONVERT_ERROR
          SUB     AL,57H
          JMP     CONVERT_DONE
CONVERT_ERROR:
          MOV     AH,1              ;set invalid char error
CONVERT_DONE:
          RET
ASCII_TO_HEX  ENDP

;hex to ascii conversion routine
;inputs AL input number between 0 and 0Fh
;outputs AL ascii character between 0 and F
;          AH input number out of range AH=1, otherwise AH=0

HEX_TO_ASCII PROC
;Add your new code here
HEX_TO_ASCII ENDP

          END               ;End of program
```

InstructorVerification