CSCI2160Quiz13HexadecimalConversionByte Name: Brandon Buchanan

Staple this quiz sheet as a cover sheet for your hardcopies.

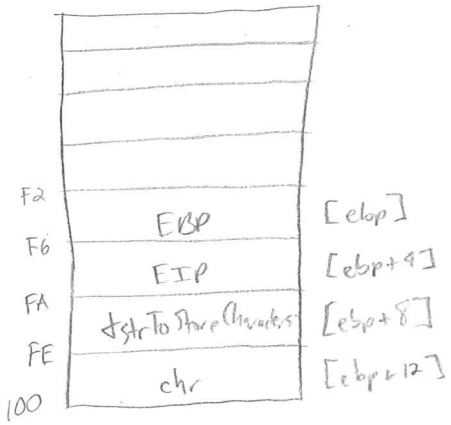
Store in a file named **hexstring.asm** the below method..

void hextochar (&strToStoreCharacters, byte chr)

The method will accept two parameters: the first is the address of a 2-byte string that will hold the 2 hex characters derived from the byte passed to it. For example if the byte has a value **F4** as its hex value, then the string would contain two new bytes **4634** (the character 'F' and the character '4').

Test your external method by writing a driver called Quiz13XXX.asm. Take screenshots of the string named **strHex** to verify that your method has correctly converted the below hex values:

```
strHex  byte    16 dup(?)  ;I deliberately made this the width of one row on the debugger
h1      byte    0F2h
h2      byte    3Ch
h3      byte    0E9h
h4      byte    0BAh
h5      byte    45h
```



```
;Program:   Quiz13XXX.asm
;Name:      Brandon Buchanan
;Date Due:  Nov 10, 2011
;Purpose:   Converting a byte in hex to its equivalent String of hex characters

    .486
    .model Flat
    ExitProcess PROTO NEAR32 stdcall, dwExitCode:DWORD
    .Stack 100h
    extern putchar:near32, hextochar:near32


    .Data
strHex  byte    16 dup(?),0 ; 16 bytes wide for easy viewing in windbg
h1      byte    0F2h
h2      byte    3Ch
h3      byte    0E9h
h4      byte    0BAh
h5      byte    45h

    .code
_start:

main PROC

;h1
    push word ptr h1
    push offset strHex
    call hextochar
    add esp,6

    push offset strHex      ;display
    call putstring          ;  the
    add esp,4               ;    string

;h2
    push word ptr h2
    push offset strHex
    call hextochar
    add esp,6

    push offset strHex      ;display
    call putstring          ;  the
    add esp,4               ;    string

;h3
    push word ptr h3
    push offset strHex
    call hextochar
    add esp,6

    push offset strHex      ;display
```

```
    call putstring          ;  the
    add esp,4               ;    string

;h4
    push word ptr h4
    push offset strHex
    call hextochar
    add esp,6

    push offset strHex      ;display
    call putstring          ;  the
    add esp,4               ;    string

;h5
    push word ptr h5
    push offset strHex
    call hextochar
    add esp,6

    push offset strHex      ;display
    call putstring          ;  the
    add esp,4               ;    string


        Invoke ExitProcess,0
        PUBLIC _start



;Procedure that accepts a null terminated string and outputs the
;string to the console screen
putstring PROC

    push ebp                ;enter
    mov ebp,esp             ;  the
    push ebx                ;    method
    push ecx                ;----------

    mov ebx,[ebp+8]         ;store the string's address

    xor ecx,ecx             ;clear ecx, create a counter to reference
                            ;  within the string
    displayString:
        mov al,[ebx+ecx]    ;store the current character in the al register
        cmp al,0            ;if the contents of al is 0, then the string
        je endOfString      ;  is done and the loop is exited
        push eax            ;pass the character to display to the method
        call putchar        ;display a character to the string
        add esp,4           ;clean up the stack
        add ecx,1           ;increment the counter by 1
        jmp displayString   ;restart the loop
```

```
    endOfString:                ;label so that the displayString loop can be
                                ;  escaped

    pop ecx                     ;----------
    pop ebx                     ;exit
    pop ebp                     ;  the
    ret                         ;    method

putstring ENDP

main ENDP

END
```

```
;Program:   hexstring.asm
;Name:      Brandon Buchanan
;Date Due:  Nov 10, 2011
;Purpose:   --Converting a byte in hex to its equivalent String of hex characters

    .486
    .model Flat
    .code


;  void hextochar(int &strToStoreCharacters, byte chr)
hextochar PROC

    push ebp
    mov ebp,esp
    push ebx

    mov ebx,[ebp+8]             ;address of string that store characters
    mov al, byte ptr [ebp+12]   ;character to convert

    shr al,4                    ;get the high order four bits
    cmp al,0Ah                  ;if it is a hex number
    jl  add30hToFirst           ;  go add 30h
    jmp add37hToFirst           ;if it hex letter, go add 37h

add30hToFirst:
    add al,30h
    jmp addFirstChar
add37hToFirst:
    add al,37h
addFirstChar:
    mov [ebx],al

    mov al, byte ptr [ebp+12]   ;restore original byte
    and al,00001111b            ;zero out 4 high order bits
    cmp al,0Ah                  ;if it is a hex number
    jl  add30hToSecond          ;  go add 30h
    jmp add37hToSecond          ;if it hex letter, go add 37h


add30hToSecond:
    add al,30h
    jmp addSecondChar
add37hToSecond:
    add al,37h
addSecondChar:
    mov [ebx+1],al

    pop ebx
    pop ebp
    ret
```

```
hextochar ENDP

END
```

Console Output: