Devin Hardy

CS 340

Section 8.10.2 #8

SetColor PROC USES eax, forecolor:BYTE, backcolor:BYTE

      Movzx eax, backcolor

      Shl eax, 4

      Add al, forecolor

      Call SetTextColor

      Ret

SetColor ENDP

Section 8.11 #2

Chess Board

```
TITLE Add and Subtract, Version 2          (AddSub2.asm)

; This program adds and subtracts 32-bit integers
; and stores the sum in a variable.

INCLUDE Irvine32.inc

.data
val1     byte  "I like pie", 0Ah, 0D, 0


.code


printRow1 PROC USES ecx
            push ebp ; stackframe!
            mov ebp, esp
            mov ecx, 4
        T1:
            mov eax, white
            call SetTextColor
            mov eax, 219
            call WriteChar
            call WriteChar
            call WriteChar
            call WriteChar
            mov eax, red
            call SetTextColor
            mov eax, 219
```

```
                call WriteChar
                call WriteChar
                call WriteChar
                call WriteChar
                loop T1 ; Loop for white red line
                mov eax, 0Ah ; character return
                call WriteChar
                mov eax, 32
                call WriteChar
                mov esp, ebp
                pop ebp
                ret

printRow1 EndP

printRow2 Proc USES ecx
                push ebp ; stackframe!
                mov ebp, esp
                mov ecx, 4
        T1:
                mov eax, red
                call SetTextColor
                mov eax, 219
                call WriteChar
                call WriteChar
                call WriteChar
                call WriteChar
                mov eax, white
                call SetTextColor
                mov eax, 219
                call WriteChar
                call WriteChar
                call WriteChar
                call WriteChar
                loop T1 ; loop for red white
                mov eax, 0Ah ; character return
                call WriteChar
                mov eax, 32
                call WriteChar
                mov esp, ebp
                pop ebp
                ret
printRow2 EndP




main PROC
        mov eax, 0Ah
        call WriteChar
        mov eax, 32
        call WriteChar
        mov ecx, 4 ; for nested loop 1
        T3:
        call printRow1
        call printRow1
        call printRow2
        call printRow2
```

```asm
        loop T3

        mov eax, 0Ah
        call WriteChar

        exit
main ENDP
END main
```