

Tracy Michaels

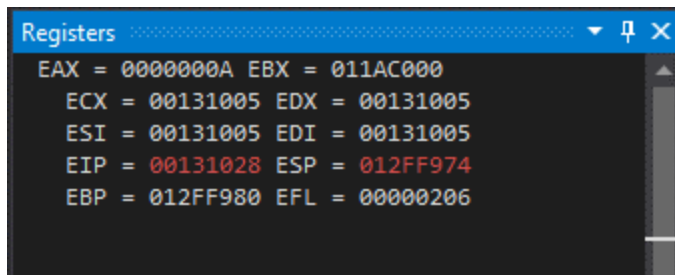
Computer Organization and Programming

CSC 3210

Assignment 5

Part A

```
main.asm*  P X
1  .386
2  .model flat, stdcall
3  .stack 4096
4
5  ;Tracy Michaels
6  ;Class: CSC 3210
7  ;Assignment#: 5 part A
8  ;This program recursively calls a procedure to sum the integers from 1 to a given value n
9
10 ExitProcess proto, dwExitCode:DWORD
11 .code
12 main proc
13     mov eax, 0
14     push 4          ;n = 4
15
16     call SumInts
17
18
19     invoke ExitProcess, 0 ;1ms elapsed
20 main endp
21
22 ;-----
23 ;This procedure sums the integers between 1 and given value n
24 ;Receives:
25 ; [ebp + 8] = n contains the value of n
26 ;Returns:
27 ; EAX which contains the sum of integers
28 ;Requires: Nothing
29 ;-----
30 SumInts proc
31     push ebp
32     mov ebp, esp
33     mov eax, [ebp+8]
34     cmp eax, 1
35     ja L1
36     jmp L2
37 L1:
38     dec eax
39     push eax
40     call SumInts
41
42     add eax, [ebp+8]
43 L2:
44     pop ebp
45     ret 4
46 SumInts endp
47 end main
```

A screenshot of a 'Registers' window from a debugger. The window has a blue title bar with the text 'Registers' and standard window controls. The background is black with text in white and red. The registers and their values are: EAX = 0000000A, EBX = 011AC000, ECX = 00131005, EDX = 00131005, ESI = 00131005, EDI = 00131005, EIP = 00131028 (in red), ESP = 012FF974 (in red), EBP = 012FF980, and EFL = 00000206.

```
Registers :
EAX = 0000000A EBX = 011AC000
ECX = 00131005 EDX = 00131005
ESI = 00131005 EDI = 00131005
EIP = 00131028 ESP = 012FF974
EBP = 012FF980 EFL = 00000206
```

## Part B

For recursive example from book:

48 Bytes when  $n = 3$

$(n+1) * (\text{eax}(4 \text{ Bytes}) + \text{return address}(4\text{bytes}) + \text{ebp}(4 \text{ Bytes}))$

\*\*\* $(n+1)$  since exit condition is 0

For answer from Part A:

36 Bytes when  $n = 3$

$(n) * (\text{eax}(4 \text{ Bytes}) + \text{return address}(4\text{bytes}) + \text{ebp}(4 \text{ Bytes}))$

\*\*\* $(n)$  since exit condition is 1