

CSC 3210 – Assignment #4
Fall 2021
Due 11/18/21, 11:59 PM

1. (9 points) Write an assembly program to implement the following.

```
sum = 0
i = 0
j = 12
var1 = 3
var2 = 3
var3 = 0
while ( (i < 20) and (j > -2)) {
    if (var1 > var3):
        var1 = var1 - i
    else if (var2 < var3):
        var2 = var2 + j
    else:
        var3 = var3 + i

    sum = var1 + var2 + var3
    i = i + 1
    j = j - 1
}
```

- Use short-circuit evaluation
- Assume that A, B, and X are **16-bit signed integers** variables
- You are not allowed to make any logical reduction to the code. You need to implement it the way it is provided.
 - o **Submit the following:**
 - Rename the asm file using your last name as Lastname1.asm and submit it.
 - Screenshot of the code
 - Then run the code until you reach INVOKE ExitProcess, 0
 - Then take a screenshot of the watch window showing Sum variable content.

Screenshot(s):

Registers
EAX = 0000000E EBX = 0000FFFF ECX = 00000000 EDX = 00000014 ESI = 000F1005 EDI = 000F0035 EIP = 000F1090 ESP = 0078FF6C EBP = 0078FF78 EFL = 00000246

100 %

Do1.asm

```

20 .code
21 main proc
22     mov eax, 0                ; putting zeros in the register values
23     mov ebx, 0
24     mov ecx, 0
25     mov edx, 0
26
27     mov ax, i                 ; putting i into eax register
28     mov bx, j                 ; putting j into ebx register
29     mov cx, var1              ; putting var1 into ecx register
30     mov dx, var2              ; putting var2 into edx register
31
32     while_block:              ; start of while loop
33         cmp ax, 20
34         jge endwhile_block    ; jumping to end if i >= 20
35         cmp bx, -2
36         jle endwhile_block    ; jumping to end if j <= -2
37         jmp if_block
38
39     if_block:                  ; if statement
40         cmp cx, var3           ; comparing var1 with var3
41         jle elseif_block
42         sub cx, ax             ; var1 - i
43         jmp continue
44
45     elseif_block:              ; else if statement
46         cmp dx, var3           ; comparing var2 with var3
47         jge else_block
48         add dx, bx             ; var2 + j
49         jmp continue
50
51     else_block:                ; else statement
52         add var3, ax           ; var3 + i
53         jmp continue
54
55     continue:                  ; adding up the vars in edi register
56         mov di, cx
57         add di, dx
58         add di, var3
59
60         add ax, 1              ; i + 1 or incrementing i
61         sub bx, 1              ; j - 1 or decrementing j
62         jmp while_block        ; looping back to beginning of while loop
63
64     endwhile_block:           ; putting edi register into sum
65         mov sum, di
66
67     invoke Exitprocess, 0      ; 1ms elapsed
68 main endp
69 end main

```

Watch 1

Search (Ctrl+E) Search Depth: 3

Name	Value	Type
sum	53	unsigned short
Add item to watch		

2. (6 points) Write an assembly program to test if the MSB in the register **al** is set:

- If it is set, then divide the content of **al** by 8 using the appropriate shift instruction and exit.
- Else, if it is not set, then multiply the content of **al** by 4 using the appropriate shift instruction and exit.
- When checking **al** MSB, do not change **al** content.
- Assume that **al** is equal to the *signed integer* 88h.
- You need to come up with the appropriate *mask* to check the MSB.
- Run your program using the debugger to verify your answers.

o **Submit the following:**

- Rename the asm file using your last name as Lastname2.asm and submit it.

- Screenshot of the code
 - Then run the code until you reach INVOKE ExitProcess, 0
 - Then take a screenshot of the register window.

Screenshot(s):

The screenshot shows a Windows assembly editor window. At the top, the 'Registers' window displays the current state of the CPU registers: EAX = 00000004, EBX = 004AA000, ECX = 00DD100A, EDX = 00DD100A, ESI = 00DD100A, EDI = 00DD100A, EIP = 00DD10D5, ESP = 0073FA5C, EBP = 0073FA68, EFL = 00000202. Below the registers, the assembly code for 'Do2.asm' is displayed. The code includes comments about the student's name (Vivian Do), class (CSC 3210), and assignment (Assignment 4 Question 2). The program tests if the MSB in the AL register is set. If set, it divides AL by 8; if not set, it multiplies AL by 4. The code then invokes ExitProcess with a return code of 0. The status bar at the bottom indicates 'End Results: AL = F1' and '≤ 1ms elapsed'.

```

1  ; Vivian Do
2  ; Class: CSC 3210
3  ; Assignment 4 Question 2
4  ; This program will test if MSB in al register is set
5  ; If set, al is divided by 8 using shift and exiting; if not set, al is multiplied by 4 using shift and exiting
6
7  .386
8  .model flat, stdcall
9  .stack 4096
10 ExitProcess proto, dwExitCode:dword
11
12 .code
13 main proc
14     mov eax, 0                ; putting zeros in the register values
15     mov al, 88h
16
17     test al, 1000000b         ; bit masking to check MSB
18     jnz msb_set               ; msb is set
19     sal al, 2                 ; msb is not set
20
21     msb_set:
22         sar al, 3
23
24     invoke ExitProcess, 0      ; End Results: AL = F1 ≤ 1ms elapsed
25 main endp
26 end main
27

```

Note:

- **Comment header** for .ASM files:

Student Name

Class: CSC3210

Assignment#: 4

Description: This program

- Follow the program standards as presented in your book. Pay more attention to code comments and consistent indentation.
- Create a new project for every question. Do not use one project with multiple .asm files.