(250 pts) CS3843 Computer Organization Exam #2 Name/abc123:_____
(150 pts) Part 1

Fast Answer: (80 pts, 4 pts each)

1. The following instruction reserves stack space for (**parameters, local variables, global variables, registers**).

   ```
   sub esp, 0x10
   ```

2. When a value is pushed to the stack, the data is written (**before, after**) esp is (**incremented, decremented**).

3. Given the stack frame set up we have seen and discussed in class, mov eax, [ebp + 8] will move what into eax? (**global variable, parameter, local variable**)

4. Given that same stack frame set up, what will be in edx after executing this instruction:

   `mov edx, [ebp + 4]`? _____ Hint: It is NOT any of the selections above.

5. There are two instructions that efficiently set a register to zero. List one of them. _____

6. After executing either of those instructions, what is the value of the ZERO flag? _____

7. Given that cl = 0x94, show the value of eax after this instruction: **`movzx eax,cl`**

8. Given that cl = 0x8F, show the value of eax after this instruction: **`movsx eax,cl`**

9. A "pop ecx" instruction uses which register implicitly? _____

10. Ecx is implicitly used by which instructions? _____

11. The flag used to control whether string instructions increment or decrement the implicit registers is called _____.

12. Exactly how many bytes are in 1 MB of memory (Express in power of 2) ? _____

13. Given a 1 byte operand, what is the range of signed displacement values? _____

14. The stack is always aligned to a _____ byte boundary.

15. List the registers implicitly used by the "string" instructions?

16. In which one of the 3 types of memory are static variables stored? (heap, stack, program)

17. There are two things the NOP instruction accomplishes while doing nothing – what are they?

18. Given that cl=0x9F, what is the minimum value that when subtracted, would set the OVERFLOW flag?

19. Afterwards, what is the value of the SIGN flag?

20. What is the difference between a compare instruction and a subtract instruction? _____

Short Answer (50 pts)

21. (6 pts) Given the memory shown below and esp = 0x12F458. What is value of eax after executing a "pop eax" instruction? What is esp after executing the pop instruction?

```
0012F454   FE CA B0 CA AD BE CE D1     eax = __________  esp = __________
```

22. (19 pts) Given that [ebp + 0x14] refers to a parameter named "tmpi" = 0x9FEC5, ebp = esp = 0x19000.

```
CODE A:                              CODE B:
lea ecx, [ebp + 0x14]       vs.      mov ecx, [ebp + 0x14]
push ecx                             push ecx
call func1                           call func1
```

a. (5 pts) Briefly describe the difference between the CODE A instructions and the CODE B instructions.

b. (4 pts) For each one, show the value of ecx as it appears on the stack.

Stack	CODE A:		CODE B:
0x18FFC			
0x19000	Prior ebp		Prior ebp

c. (5 pts) What is the stack address that will hold the return address when func1 is called?

d. (5 pts) At what address is tmpi stored?

23. (5 pts) What 2 operations are performed by a return instruction?

24. (5 pts) What 2 operations are performed by a call instruction?

25. (15 pts) Examine the following assembly instructions, and answer the subsequent questions.

```
0020 mov eax, [ebp + 0x0C] ; value here = 0x000000FF
0023 mov cl,  [ebp + 0x08] ; value here = 0x82
0026 cmp al, cl
0028 jl  label       ; label is at address 003C, jl is signed
002A nop
002B
...
003C label: sub al, cl
```

a. (3 pts) Inside a function, assuming that ebp is used for the stack frame, what is at the address ebp+0x0C with respect to a C function call?

b. (3 pts) Given a signed operation, al is (**greater than, less than, equal to**) cl?

c. (3 pts) What is the address from which the offset to the jl is calculated? _____

d. (3 pts) How would you determine the value of the offset for the jl instruction? _____

e. (3 pts) Suppose ecx = 0xABCD1234 prior to executing the code above. What is the new value after executing the code at address 23.

ecx = 0x_____

Reading Code (20 pts)

```
004010C0:     push    ebp
004010C1:     mov     ebp, esp

004010C3:     sub     esp, 44h
004010C6:     push    esi
004010C7:     push    edi

004010C8:     mov     ecx, 06h
004010CD:     mov     esi, offset aMsg ; "Money can't buy happiness"
004010D2:     lea     edi, [ebp-34]
004010D5:     rep movsd
004010D7:     movsw
```

26. (2 pts) Briefly explain what the first 2 instructions are doing.

27. (2 pts) Why are we then subtracting 0x44 from esp?

28. (2 pts) Why do we need to push esi/edi?

29. (4 pts) The length of the string is 25 bytes, why do we load ecx (the counter) with a 6?

30. (4 pts) What does the "rep" prefix do for the instruction at address 0x4010D5? There are several parts to this answer.

31. (2 pts) What is the purpose of the movsw instruction?

32. (4 pts) What is the "big picture" function of the code shown here?