# CS 261 Machine Organization (Spring 2020) – Lab Week 5

***Total – 100 points***

Name:______________________________________ UIN: ____________________

Name:______________________________________ UIN: ____________________

Complete your work in the space provided below. The lab is due at the end of your session via Gradescope. You may work individually or in groups of 2. Only one group member should submit to Gradescope but remember to add all members of the group to the submission. Please ask for assistance if you have any questions.

1. The attached labw5.s file is x86-64 assembly source file and it contains some assembly operations. Compile the code with -g flag on systemsX.cs.uic.edu (where X is either 1 or 2 or 3 or4) using command "**gcc -g labw5.s**"

2. Open the executable obtained after compilation with gdb using command "**gdb a.out**"

3. Set a break point at line 25 that contains the instruction "movabsq $0xabcdef1234567890, %rax" with the command b 25 (this works because you did the gcc -g command earlier).

   Fill in the stack address, value and address expression for top 4 giant words(8 bytes each) of stack at that point of execution. Recall the 64-bit register rsp is the Stack pointer and holds the address that is the top of the stack. First row provides you an example. Use command "**x/4gx $rsp**" in gdb. [Note, you may have different stack addresses]

**(4 points each, total 36 points)**

| **Stack address** | **Value (as hex)** | **Address Expression** |
|---|---|---|
| | **0x20756f7920657241** | **(%rsp)** |
| | | |
| | | |
| | | |

**4.** What is the string message starting at (%rsp)? **(10 points)**

______________________________________________________________________________

**5.** Fill in the space below with any value(s) that would satisfy the C format string that is located at address 0x20(%rsp). **(9 points)**

______________________________________________________________________________

6. Based on the provided source file, give the contents of rax after execution of each instruction below. **(5 points each, total 25 points)**

| **Assembly instruction** | **Value of register rax (as hex)** |
|---|---|
| **movabsq $0xabcdef1234567890, %rax** | |
| **movb $0x5c, %al** | |
| **movl $0x5c, %eax** | |
| **movsbq %cl, %rax** | |
| **movzbq %cl, %rax** | |

**7.** Based on the provided source file, complete the table below. The values should be the results obtained after execution of instructions given below. **(2 points each, total 20 points)**

| **Assembly instruction** | **Destination** | **Value (as hex)** |
|---|---|---|
| **leaq 0x20(, %rcx, 4), %rdx** | | |
| **addq 0x10(%rcx, %rcx, 2), %rax** | | |
| **movl (%rsp, %rdx), %eax** | | |
| **shrq $56, %rbx** | | |
| **sarq $56, %rax** | | |