CSCI 3320
Fall 2022
Exam 3

Last Name: _____

First Name: _____

Write all answers on the answer sheets provided. You *may* use the following resources during the exam: *x86 Assembly Quick Reference*, `iomacros.asm` source code.

1. (4 pts) Name two common uses for the system stack.

2. (4 pts) The push command is purported to accomplish two things: 1) it adjusts the stack pointer, and 2) it puts the operand on the stack. Name two differences between the following sets of instructions beside the fact that one of them requires two statements:

```
push the_operand          vs          sub rsp,size_of_the_operand
                                      mov [rsp],the_operand
```

3. (8 pts) Write a macro called `abs` that will accept a 32-bit integer value as an argument (register or memory location) and will return (in EAX) the absolute value of the passed parameter. The macro should use the system stack to save/restore all modified registers (except for RAX). The macro should be written so it can be called multiple times in a single program. NOTE: The absolute value of 40 is 40. The absolute value of -40 is 40.

4. (8 pts) Show how the following section of Java code could be implemented in assembly language. Be sure to show declarations of labels as part of your answer. After the code is complete the label `a` should contain the answer.

```
int x;
float y;
double z,a;
...
a= (x+y) * (z-a) / 4.4;
```

5. (4 pts) List (in order) the first four registers that are used for passing parameters to functions.

6. (8 pts) Consider the following Java function (recall that a variable of type `long` is a 64-bit integer). Write this function in assembly language assuming a register-based calling convention in which registers are saved by the caller.

```
public static long calc(long x, long y, long z)
{
   if (z<=0) return x+y;

   if (z%2==0)
      return calc(x+1,y,z-1);
   else
      return calc(x,y+1,z-1);
}
```

7. (2 pts each) Using words (not assembly language) explain a strategy for bit manipulations to solve each of these problems:

   (a) multiplying an integer by 2

   (b) perform the set operation: A-B (NOTE: A-B means the elements in A that are not also in B)

8. (12 pts) IP (v3) numbers are 32-bit values that are used as unique identifiers for devices on the internet. The numbers are typically written using "dotted notation" in which the the 32-bits are considered in 8-bit chunks that are then displayed as base-10 values. For example, consider the follow 32-bit value that represents an IP number: 00001010 00010001 00001111 00100000. In dotted notation it would be written as: 10.17.15.32

   Write a function called `print_ip` that will accept an IP number as a 32-bit integer) as a parameter and will use the I/O macros along with bit manipulations to display the IP number in dotted notation. The function should follow all the conventions of functions with register-based parameters. You may assume registers are save by the caller.

   To accomplish this you will isolate each group of 8 bits in the number and print them as an integer followed by a period. Your function should use a register-based calling convention and should save/restore any registers it modifies (except for RAX).

9. (2 pts each) Define each of the following terms:

   (a) ripple-carry adder

   (b) full adder

   (c) ALU

10. (6 pts) Draw an exclusive-or circuit using AND, OR, and NOT gates.

11. Suppose a "friend" asks you to design an efficient circuit that given three binary inputs, `a`, `b`, and `c`, will produce the result, `r`, portrayed in the following chart:

a	b	c	r
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(a) (4 pts) Represent this chart as a boolean algebra expression.

(b) (2 pts) How many gates are represented by the expression you generated in problem 11a?

(c) (4 pts) Apply various rules of boolean algebras to simplify the expression as much as you can.

(d) (4 pts) Represent the simplified expression as a circuit diagram.

(e) (2 pts) How many gates are represented by your simplified diagram?

NOTE: In case you find it helpful, the five properties of a Boolean algebra (along with two additional properties) are listed below:

Commutative	$x + y = y + x$	$x \cdot y = y \cdot x$
Associative	$(x + y) + z = x + (y + z)$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
Distributive	$x + (y \cdot z) = (x + y) \cdot (x + z)$	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
Identity	$x + 0 = x$	$x \cdot 1 = x$
Complement	$x + \overline{x} = 1$	$x \cdot \overline{x} = 0$
DeMorgan's	$\overline{x + y} = \overline{x} \cdot \overline{y}$	$\overline{xy} = \overline{x} + \overline{y}$
Involution	$\overline{\overline{x}} = x$	

12. (20 pts) What assembly language command is used as an incantation to summon the lobster?