COA1 Exam 2 – Fall 2018

Name: ________________________________ Computing ID: __________

Letters go in the boxes unless otherwise specified (e.g., for **C** 8 write "C" not "8").
Write Letters clearly: if we are unsure of what you wrote you will get a zero on that problem.
Bubble and Pledge the exam or you will lose points.
Assume unless otherwise specified:

- the following have been declared:

```
void *malloc(size_t);    void free(void *);
int puts(const char *);  int printf(const char *, ...);
```

- `char`, `short`, `int`, and `long` are 8-, 16-, 32-, and 64-bits long, respectively; and that `float` is 32- and `double` is 64-bits long.
- the compiler pads pointers where it is allowed to do so such that
  - ▷ an `X`-pointer is a multiple of `sizeof(X)` for all types `X`
  - ▷ `sizeof(struct X)`
    - – an even multiple of the size of its largest field
    - – the smallest such multiple big enough to store all its fields
- compilation happens using `clang` on a Linux system

Single-select by default: Multiple select are all clearly marked; answer them by putting 1 or more letters in the box, or writing "`none`" if none should be selected.
Mark clarifications: If you need to clarify an answer, do so, and also add a ⋆ to the top right corner of your answer box.

. . . .

Information for questions 1–4
Suppose the assembly given in each subquestion was inserted at random between two instructions of a function, with all jump targets and other code addresses updated accordingly. Either state that this has no functional impact by writing "`nop`" or describe a scenario where such an insertion could change the behavior of the function.

Question 1 [2 pt]: (see above) What if we insert `addq $0,%rax`?

Answer: ____

____

Question 2 [2 pt]: (see above) What if we insert `movq %rax,%rax`?

Answer: ____

____

Information for questions 3–11

For each of the following questions, assume the first eight registers have the following values prior to the assembly being run:

Register	RAX	RCX	RDX	RBX	RSP	RBP	RSI	RDI
Value (hex)	0	1C3F5678	200400800	FFFF	200	240	20	100

Note: the questions are independant. Do not use the result of one as the input for the next.

Answer by writing a changed register and its new value, like "RDI = 24F2", leaving one or more lines blank if fewer registers change than there are lines.

Question 3 [2 pt]: (see above) Which program registers are modified, and to what values, by `leaq 0x10(%rdi,%rsi,4), %rax`?

____

____

Question 4 [2 pt]: (see above) Which program registers are modified, and to what values, by `pushq %rcx`?

____

____

Question 5 [2 pt]: (see above) Which program registers are modified, and to what values, by `retq`?

____

____

Question 6 [2 pt]: (see above) Which program registers are modified, and to what values, by `addq %rsi, %rdi`?

____

____

Question 7 [2 pt]: (see above) Which program registers are modified, and to what values, by `movl %ecx, %edx`?

____

____

Question 8 [2 pt]: Consider the following assembly:

```
pushq (%rbp)
retq
```

Functionally (ignoring time taken to execute), what does this do?

A the same thing as `retq` without the preceding `pushq`
B the same thing as `retq` without the preceding `pushq`, but after returning the stack is one item larger
C it jumps to an address stored in `%rbp`
D it jumps to an address stored in memory pointed to by `%rbp`
E it depends on the contents of `%rbp`
F it depends on the contents of `(%rbp)`

Answer:

Information for questions 9–17
For each of the following bugs, indicate the stage of compilation that would be find it. If it would not be found until run-time, write "`none`". The stages are

- **L**exing – breaking input into words and related tokens
- **P**arsing – making an abstract syntax tree (AST)
- **T**ype-checking – annotating the AST with data types, etc
- **C**ode generation – creating assembly
- **A**ssembling – turning assembly into machine code
- **L**inking – attaching library files to code

Question 9 [2 pt]: (see above)
Incorrect signature of library function

Answer:

Question 10 [2 pt]: (see above)
Using an undeclared variable

Answer:

Question 11 [2 pt]: (see above)
Having more "`(`" than "`)`" in your program

Answer:

Question 12 [2 pt]: (see above)
Invoking a function you've declared but never defined

Answer:

Question 13 [2 pt]: What value is placed in `x`?

```
#define THING 3 + 2
int x = THING * 2;
```

Answer:

Question 14 [2 pt]: What is `sizeof(float[5])`? See the assumptions on page 1 to compute an exact number.

Answer:

Question 15 [2 pt]: What is the minimum number of bytes of read-only memory needed for the compiler to store the following set of string literals: `"earing"`, `"hearing"`, `"wearing"`?

Answer:

Question 16 [8 pt]: The following program both (a) contains a memory error and (b) has a memory leak. Circle and describe the error, and insert any needed `free` invocations to fix the memory leak.

```
typedef struct { int *data; int capacity; int size; } stack;

// add a value to the stack, increasing its size if necessary
void push(stack s, int val) {

    if (s.size == s.capacity) {
        // stack full; double the capacity of the array before continuing
        int *tmp = (int *)malloc(s.capacity*2);

        for(int i=0; i<s.capacity; i+=1) {
            tmp[i] = s.data[i];

        }

        s.data = tmp;

        s.capacity *= 2;

    }
    // put the data in the stack and increase it's used size

    s.data[s.size] = val;

    s.size += 1;
}

// remove an object from the stack (assume there is something to remove)
int pop(stack s) {

    s.size -= 1;

    return s.data[s.size];
}
```

Question 17 [6 pt]: Re-write the following snippet of C code to have the same behavior without using `goto` or labels.

```
L0:
    y += 1;
    if (x&1) goto L1;
    x >>= 1;
    goto L2;
L1:
    x *= 3;
    x += 1;
L2:
    if (x > 1) goto L0;
```

Pledge:

On my honor as a student, I have neither given nor received aid on this exam.

____

Your signature here