CSCI 3320
Fall 2022
Exam 2

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, calculator, the lobster.

1. Write the x86_64 assembly language statements that correspond to each of the following high-level language statements. The variables `i`, `j`, `k`, and `n` are all 32-bit signed integers.

   (a) (4 pts) `if (i <= n && j <= n) { k= k + n; }`

   (b) (8 pts) `k= 0; i= 1; do { i= i * 2; k++; } while (i<n);`

2. Suppose that `a` is an array of `n` null-terminated strings, each of which has reserved 24 bytes (i.e., longest possible string is 23 characters). Write a section of `x86_64` assembly code to accomplish these tasks. You may assume that `n` is a 32-bit signed integer.

   (a) (4 pts) Show declaration that will reserve space to up to 10,000 strings at the label `a`. Also, reserve swap space for holding a single string at the label `junk`.

   (b) (8 pts) Use the MOVSx command to swap the contents of `a[i]` and `a[j]`. Use the memory reserved at `junk` for the swap space. Assume `i` and `j` are 32-bit integers.

   (c) (12 pts) Write a section of code that will display the length of each string in the array with one length per line. NOTE: To determine the length of the string, find the position of the null-terminating characters (a zero-byte).

3. Suppose you have an array of references to structured data with each element containing the following information about a sugar glider: int id, int wingspan (both are 64-bit signed integers).

   (a) (4 pts) Show the commands you would need to issue in NASM to reserve (at the label `sg`) enough memory to store 500 references to sugar glider records.

   (b) (4 pts) Show how you would use `%define` to give named offsets to the contents of a sugar glider record.

   (c) (16 pts) Suppose the `sg` array has been initialized and that all 500 addresses are pointing to already allocated sugar glider records. Further suppose that the entries are supposed to be ordered with largest wingspans coming first. Write necessary statements that will determine whether the entries have been ordered properly or not. Print 0 if they are *not* properly ordered and 1 if they are properly ordered.