1. (10 pts) Write a short descrip3on of the x86 architecture that would be suitable to provide a programmer who is unfamiliar with x86\_65. (No need to describe the FP or SIMD features)

2. (5 pts) What is an “implied operand” and how does it relate to the x86\_64 architecture?

3. (5 pts) Describe either the GNU or Microsoft calling convention for the C language and the x86\_64 architecture. (Specify which conven3on you’re describing)   
4. (5 pts) Briefly describe the function of the following:   
a. Assembler   
b. Linker   
c. Loader

5. (5 pts) What is the size of a double word in x86\_64?

6. (5 pts) What does the RIP do in the x86\_64 architecture? What does it point to?

7. (5 pts) To reference memory, what four inputs can be used and how do they relate to the memory address generated?

8. (5 pts) What does SIMD mean in computer architecture.

9. (10 pts) As a computer programmer, would you prefer to write code for the AArch32 or x86\_64 architecture? Why?

10. Given the following C program, write assembly rou3nes for:   
a. (15 pts) int isOdd(int a)   
b. (15 pts) int someCalcs(int a, int b, int c)   
c. (15 pts) int arrRef1d(int \*arr, int element)   
   
#include <stdio.h>   
   
/\*   
 Function isOdd returns 1 if the argument a is an   
 odd integer, 0 otherwise   
\*/   
extern void isOdd (int a) asm ("intMean\_");   
   
/\*   
 Function someCalcs returns (a + b) / c   
 using integer division.   
 If c == 0, returns –1   
\*/   
extern int someCalcs (int a, int b, int c)asm

(“someCalcs\_”)   
   
/\*   
 Function arrRef1d returns arr[element]   
\*/   
extern int arrRef1d) (int\* arr, int element) asm   
 (“arrReff1d\_”)   
   
   
   
void main () {   
 int arr[10] = {1, 2, 3, 4, 5};   
 int x = 10, y = 20, z = 30; a = 40;   
   
 printf(“Results of isOdd(5): %d \n”, isOdd(5));   
 printf(“Results of isOdd(10): %d \n”, isOdd(10));   
   
 printf(“a = %d, b = %d, c = %d, result = %d\n”,   
 x, y, z, someCalcs(x, y, z));   
 printf(“a = %d, b = %d, c = %d, result = %d\n”,   
 1, 2, 0, someCalcs(1, 2, 0));   
   
 printf(“arr[2] = %d\n”, arrRef1d(arr, 2));   
 printf(“arr[7] = %d\n”, arrRef1d(arr, 7));   
   
 printf(“Thanks for a great summer semester!\n”);   
   
}