University of Delaware
Computer and Information Science
CISC 260 – A Sample Final Exam

1. This is an open notes exam. You are not allowed to use any electronic devices except standard calculators.
2. Check that you have all pages. There are 4 problems for a total of 100 points.
3. Write you name on every page in the space provided
4. If you need more space, use the back of the same page. But do not feel obligated to "fill up" all the space that is provided.
5. Give clear, concise answers to each question.

Problem	Grade
1	
2	
3	
4	
Total	

1. [30 points] Short Answers

a. Write down the IEEE 754 binary representation of the number -0.75 in single precision.

b. What decimal number does the bit pattern 1010 1101 0001 0000 0000 0000 0000 0010 represent if it is a floating point number in IEEE 754 standard?

c. Is there an overflow for an 8-bit machine when adding a two's complement integer x to a two's complement integer y as given below? Show your work. x = 0100 1011 and y = 0111 0100

d. In linking stage, does the label "else" in the following code require relocation or not?

```
BEQ  else
```

e. Is the recursion tail recursive? If not, convert it to a tailed recursion (only at the high level code, not the assembly code.)

```
int SquareSum(int n) {
     if (n == 1) return 1;
     return n*n + SquareSum(n - 1);
}
```

2. [25 points] Below is a function that inserts a node into a linked list.

```
void insert(node** root, node* a_node) {

   if((a_node)->data <= (*root)->data) {
           (a_node)->next = *root;
       *root = a_node;
   } else if((*root)->next != NULL) {
     if ((a_node)->data <= ((*root)->next)->data){
        (a_node)->next = (*root)->next;
        (*root)->next = a_node;
     } else {
       insert(&(*root)->next, a_node);
     }
   } else {
      (*root)->next = a_node;
  }

}
```

Translate the function into ARM assembly code. You can assume that the two arguments are put in r0 and r1 respectively by the caller.

A node in the linked list is structured as follows

data	Pointer to the next node

3. [20 points]

```
.text
main:
     ldr  r0, =x
     bl   foo
     mov  pc, lr
foo:
L2:  mov  r5, #0
     cmp  r0, #0
     blt  L1
     add  r5, r5, #1
     sub  r0, r0, #1
     b    L2
L1:  mov  r0, r5
     mov  pc, lr
.data
     x: .word 10
```

For the above ARM assembly program, answer the following questions.

a) Show the address next to each instruction according to the ARM conventions for memory allocation.
b) Draw the symbol table listing the labels and their addresses
c) Convert "blt L1" into machine code (written in hex)
d) Sketch a memory map showing where data and instructions are stored.

4. [25 points] Consider the following ARM assembly code

```
        mov   r3, #0
        mov   r7, #10
Loop:   ldr   r1, [r2, #0]
        mul   r1, r1, r4
        add   r2, r2, #4
        add   r3, r3, #1
        cmp   r7, r3
        bne   Loop
```

a) Compute the number of cycles needed for each loop iteration. Assume CPI is 1 for data processing, 5 for data transfer, and 2 for branching.

b) What is the average CPI for the above code?

c) Assume the clock cycle is 200ps, what is the total CPU time of running the above code.

d) What type of locality does this code have for accessing the data in memory? If the instruction **`ldr`** can load four adjacent words into the cache, what is the miss rate? Note that the cache has two sets, each set has a block of 4 words, with memory mapping as shown below, where the values are only placeholder to indicate the numbers of bits in each field.