CSL211
TUTORIAL SHEET Sep 12^{th} week

1) Assume that you want to divide A by B. A is saved in register R0, and B is saved in register R1. The quotient needs to be saved in R7, and the remainder needs to be saved in R8.
   1. Write code in ARM assembly for restoring division.
   2. Write code in ARM assembly for non-restoring division.

2) Assume that you are multiplying two numbers, A and B. A is characterized by having fairly long runs of ones. For example, A can have the form 00111000111111001111. Can we optimize the traditional bit by bit O(n log(n)) time, right shift based multiplication algorithm to perform faster with such kind of inputs? If so, how ?

3) How can you extend the bit by bit multiplication algorithm for signed multiplication?

4) What is the range of positive and negative denormal numbers?

5) Numerical values A and B, are saved in the computer as A' and B'. Neglecting any further truncation or roundoff errors, show that the relative error of the product is approximately the sum of the relative errors of the factors.

6) You are given two floating point numbers, A and B, of the form (1+x) and (1-y), where 0 <= (x,y) < 0.5. What is a quick way of to compute A / B, using a sequence of multiplications, and additions using an iterative algorithm?

7)
```
int func1() {
   float dd = a + b;
   float d  = dd - c;
   if(d > e)
      return 1;
   return 0;
}
int func2() {
   float d  = a + b - c;
   if(d > e)
      return 1;
   return 0;
}
int main (){
   a = 1;
   b = pow(2, -50);
   c = 1;
   e = pow(2, -51);
   printf("%d\n",func1());
   printf("%d\n",func2());
}
```

What is the output of this program?