CSE 200 Spring 2008, HW 9 – due date: April 1, in or before class.
(just to remind you the hw submission policy

```
1) 30% if submitted after the class but on the same due day (on
Tuesday)
2) 50% if submitted the next day (on Wednesday)
3) no homework will be accepted after Wednesday
```

Study the lecture notes on recursion (lectures 8 and 9) for this set of problems.

Problem 1. (5 points) Given two positive integers m and n such that m<n, the greatest common divisor (GCD) of m and n is the same as the greatest common divisor of m and (n-m), and the GCD of 0 and any nonzero integer n is n. Use this fact to write a recursive function GCD(m, n) to compute the greatest common divisor of m and n.

```
function GCD=GCDsoln1(a,b)
if (b==0)
   GCD=a
else
   GCDsoln1(b,mod(a,b)) %you can also use "rem"
end

function GCD=GCDsoln2(m,n)
if m>n
   big=m;
   little=n;
else
   big=n;
   little=m;
end
if little==0;
   GCD=big;
else
   GCD=GCDsoln2(little,big-little);
end
%Note: Watch out to make sure that your function is ending instead
%of calling itself again when m or n equals 0!
```

Problem 2. (8 points) Given a contiguous function f(x) and two values a and b, we know f(a) < 0 and f(b) > 0. Design a **divide-and-conquer** strategy and write a **recursive** function to find a root of f(x), i.e., to find an x such that f(x) = 0 to certain precision, where the precision can be a parameter to your function. Hint: you shouldn't use Newton's method – in fact, it has nothing to do with Newton's as we don't use derivative of f(x) at all. if you still have problem with it, study the square root problem in lecture 9.

```
function Root=Rootsoln1(func,a,b)
avg=(a+b)/2;
s=feval(func,avg);%function must be inputed as 'function'
if abs(s)<=0.001
   Root=avg;
elseif s>0
   Root=Rootsoln1(func,a,avg);
elseif s<0
   Root=Rootsoln1(func,avg,b);
end

function Root=Rootsoln2(func,a,b,precision)
avg=(a+b)/2;
s=inline(func); %function must be inputed as 'function(x)'
s(avg);
if s(avg)<=precision %if you use a number here you don't need the extra input argument
   Root=avg;
elseif s(avg)>0
   Root=Rootsoln2(func,a,avg,precision);
elseif s(avg)<0
   Root=Rootsoln2(func,avg,b,precision);
end

function Root=Rootsoln3(func,a,b)
avg=(a+b)/2;
handle=@ (x) eval(func); %function must be inputed as 'function(x)'
s=handle(avg);
if abs(s)<=0.001
   Root=avg;
elseif s>0
   Rootsoln3(func,a,avg);
elseif s<0
   Rootsoln3(func,avg,b);
end
```

Problem 3. (8 points) Give a positive number a, use divide-and-conquer and write a recursive function to compute log(a), with base 2, to a certain precision (which is a parameter to your program). Here, a >= 1. You cannot use Matlab built-in *logarithmic* functions, but only simple operations of +, -, *, / and *power* function. Hint: think about the definition of log function. Let log(a) = x, then 2^x = a. To start, you need to identify a range where the solution may appear, this will give you upper and lower bounds to start divide-and-conquer.

```
function logsoln=Logsoln1(a,lowerbound,upperbound,precision)
guess=(lowerbound+upperbound)/2
if abs(2^guess-a)<precision
   logsoln=guess
elseif 2^guess>a
      logsoln=Logsoln1(a,lowerbound,guess,precision)
else
   logsoln=Logsoln1(a,guess,upperbound,precision)
end

% Note: call the function with Logsol1(a, 1, a, 0.001) for example
```