On program 1

You can use this code verbatim if you like; it is the contents of my macheps.m file.

```
function mu = MACHEPS
%
% MEPS approximates the machine precision, mu.
%
% No input data needed.
%
% mu is defined to be the smallest positive float that satisfies
%
%             fl(1 + mu) > 1,
%
% We let m be a variable that converges to mu from above by halving.
% At each step we will test to see if  fl(1+m) > 1.
% When the test fails (that is, if fl(1+m) == 1) then m is one
% iteration too small, so we approximate mu by 2m.

m = 1.0;          % start m out big
t = 1.0 + m;      % t = fl(1+m)
while t > 1.0,    % this is the test "is (1+m) > 1?"
    m = m/2;        % if it passed the test, half m
    t = 1.0 + m;    % t = fl(1+m)
end
mu = 2*m;         % if we get here, then m failed the test,
                  % which means m < mu, so we set mu = 2m.
```

While we are on the subject of program 1, here is some code which will help you address the question "*what is small?*" for your quadroot routine. I would suggest inserting the following code (or your own variation) into the beginning of your quadroot.m file. (Crunched for space here I wrote multiple statements per line).

```
m = max(abs([a,b,c]));
if m == 0,   errflag = -1;   return;   end
if b < 0,    m = -m;         end
a = a/m;   b = b/m;      c = c/m;
```

Please think about all of these lines and be confident that you understand what they do, and why we do them. The "if $b < 0$" line is not needed, but it simplifies your code later. How?