```
/*To find the roots of the equation f(x) = 0 by successive approximations,
we write it in the form x = p(x). The roots of f(x) = 0 are the same as the points of inter
section of the straight line y = x and the curve representing y = p(x).
Let X = X0 be an initial approximation of the desired root, then the first approximation x1
 is given by X1 = p(X0). Now, treating x1 as the initial value, the second approximation is
 X2 = p(X1). Proceeding in this way the nth approximation is given by Xn = p(Xn-1)
The equation Xk+1 = 1.2/log(xk) is used to find the roots of the equation
xlog(x) = 1.2.

In this method we use X0 as initial value to find X1 using X1 = 1.2/log(X0), then X2 is cal
culated using X2 = 1.2/log(X1) and the process goes on.....

Finally we find the value of X for which x = 1.2/log(x), and that is the point of convergen
ce of the equation f(x) = xlog(x) - 1.2.
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void log_eq(double guess); //Function prototypes

int main(int argc, char **argv)
{
  if(argc != 2)
  {
       fprintf(stderr, "USAGE: %s <guess>\n", argv[0]);
       exit (0);
  }
  double guess;

  guess = atof(argv[1]);
  log_eq(guess); //Invoking the function
  exit (0);
}

//Xk+1 = 1.2/log(Xk)
void log_eq(double guess)
{
  double x, root; //x is xk
  int i = 1; //To record number of iterations

  printf("F(x) = xlog(x) - 1.2\n");
  x = guess;

  //Loop infinitely
  while(1)
  {
      //printf("Checking: %lf\n", x);
      root = 1.2/log10(x); //Substitute the values into the equation Here Xk+1 is root And
Xk is x
      if(root == x) //If loop converges, then break out of the loop
      {
          printf("After %d iteration root %lf found.\n", i, root);
          break;
      }
      //If the loop goes infinitely, break after 100000 iterations
      if(i > 100000)
      {
          printf("Root of F(x) might be around %lf\n", root);
          exit (0);
      }
      x = root; //Update Xk
      i++;
  }
  return;
}
```