```c
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <math.h>

#define MAX_N 60
#define SUCCESS 1

void part2a (void)
{
    double x_prev;
    double x_next;
    double x_0 = 1.0f;
    x_prev = x_0; //x_0 is 1
    double n_2poweri = 1.0f; // this is to keep track of the 2^i term
                             // to do 2^i on doubles since it is not
                             // allowed in C otherwise and 2^0 = 1

    for(int i = 1; i <= MAX_N; i++)
    {
        x_next = 2*n_2poweri*(sqrt(1+(1/n_2poweri)*x_prev) - 1);
        x_prev = x_next; //update the x_n for the next iteration
        double diff = x_next - log(2);
        n_2poweri = 2*n_2poweri;
        printf("%s%d%s%.8e %s%d%s %.8e \n", "x",i,"=",x_next,"x",i,"-log(x0+1)=",diff);
    }
}
void part2b (void)
{
    double x_prev;
    double x_next;
    double x_0 = 1.0f;
    x_prev = x_0; //x_0 is 1
    double n_2poweri = 1.0f;
    for(int i = 1; i <= MAX_N; i++)
    {
        x_next = 2*x_prev / (sqrt(1+(1/n_2poweri)*x_prev) + 1 );
        x_prev = x_next;
        double diff = x_next - log(2);
        n_2poweri = 2*n_2poweri;
        printf("%s%d%s%.8e %s%d%s %.8e \n", "x",i,"=",x_next,"x",i,"-log(x0+1)=",diff);
    }
}

int main(void)
{
    part2a();
    part2b();
    return SUCCESS;
}
```