

```

1  #include <stdio.h>
2  #include <unistd.h>
3  #include <string.h>
4  #include <math.h>
5
6  #define MAX_N 60
7  #define SUCCESS 1
8
9  void part2a (void)
10 {
11     double x_prev;
12     double x_next;
13     double x_0 = 1.0f;
14     x_prev = x_0; //x_0 is 1
15     double n_2poweri = 1.0f; // this is to keep track of the 2^i term
16                                // to do 2^i on doubles since it is not
17                                // allowed in C otherwise and 2^0 = 1
18
19     for(int i = 1; i <= MAX_N; i++)
20     {
21         x_next = 2*n_2poweri*(sqrt(1+(1/n_2poweri)*x_prev) - 1);
22         x_prev = x_next; //update the x_n for the next iteration
23         double diff = x_next - log(2);
24         n_2poweri = 2*n_2poweri;
25         printf("%s%d%s%.8e %s%d%s %.8e \n", "x",i,"=",x_next,"x",i,"-log(x0+1)=",diff);
26     }
27 }
28 void part2b (void)
29 {
30     double x_prev;
31     double x_next;
32     double x_0 = 1.0f;
33     x_prev = x_0; //x_0 is 1
34     double n_2poweri = 1.0f;
35     for(int i = 1; i <= MAX_N; i++)
36     {
37         x_next = 2*x_prev / (sqrt(1+(1/n_2poweri)*x_prev) + 1 );
38         x_prev = x_next;
39         double diff = x_next - log(2);
40         n_2poweri = 2*n_2poweri;
41         printf("%s%d%s%.8e %s%d%s %.8e \n", "x",i,"=",x_next,"x",i,"-log(x0+1)=",diff);
42     }
43 }
44
45 int main(void)
46 {
47     part2a();
48     part2b();
49     return SUCCESS;
50 }

```