

```

1  #include <stdio.h>
2  #include <math.h>
3
4  typedef double fp(double); // 関数ポインタ
5
6  // 関数func と show_func は既に作成したものを利用すること
7  double func(double a){ ...
8  void show_func(fp* func, const double a, const double b, const int n){...
9
10 // 2分法により解を算出する関数
11 int solve_bisection(fp* func, double &ans, double low, double high){
12     const double EPS = 1e-6; // 収束判定値
13     const int limit = 100; // 反復回数の上限
14
15     for(int i=0; i<limit; i++){
16         double x = (low + high)/2.0; // 中点を計算
17         if (func(x)>0){
18             high = x;
19         }else{
20             low = x;
21         }
22         // 収束判定条件
23         if(fabs(high - low) < EPS){
24             ans = x;
25             return i+1; // 解が収束した場合、反復回数を返して関数を抜け出す
26         }
27     }
28     // 解が得られなかった場合、-1 を返して関数を抜け出す
29     return -1;
30 }
31
32 // メイン関数
33 int main(void){
34     double ans = 0.0; // 解を格納する変数
35     double a = 1.0;
36     double b = 2.0;
37
38     show_func(func, a, b, 100);
39     int ret = solve_bisection(func, ans, a, b);
40
41     if(ret > 0){
42         printf("\nThe root_x has been successfully found\n");
43         // 解と計算回数の表示
44         printf("at %9.6f in %d times of iteration. \n",ans, ret);
45     }else{
46         // 求められなければError を表示
47         printf("Error : The root_x cannot be found.\n");
48     }
49     return 0;
50 }

```