

```

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

/* 関数 */
double f1(double x){
    return pow(x,4.0) -3*x +1; }
double f2(double x){
    return cos(x) - x; }
double f3(double x){
    return pow(M_E,x) -1/x; }
double f4(double x){
    return (pow(x,3.0) - pow(x,2.0)*3.0 +9.0*x - 8.0); }

/* 関数を設定 */
#define f(x) f1(x)

/* 差分 */
/* 中心差分    ちゅうしん さぶん    central difference */
double central_df(double x, double h) {
    return (f(x+h) - f(x-h))/(2.0*h);
}
/* 前進差分    ぜんしんさぶん    forward difference */
double forward_df(double x, double h) {
    return (f(x+h) - f(x))/h;
}
/* 後退差分    こうたいさぶん    backward difference */
double backward_df(double x, double h) {
    return (f(x) - f(x-h))/h;
}

/* 差分の設定*/
#define EPSILON 0.0000001
#define df(x) central_df(x, EPSILON)    /* 中心差分 */
// #define df(x) forward_df(x, EPSILON) /* 前進差分 */
// #define df(x) backward_df(x, EPSILON) /* 後退差分 */

/* ニュートン法 */
double newton(double np){
    int i = 0;
    double xk = np, xk1;

    while(1){
        i++;
        xk1 = xk - f(xk)/df(xk);
        printf("Try %d Solution %.7f \n", i, xk1);
        if(fabs(xk1 - xk) < EPSILON){
            break;
        }
        xk = xk1;
    }
    return xk;
}

```

```
/* main 関数*/
int main(int argc, char** argv){
    double initial_value = 2.0;
    printf(" d f(%.6f) = %lf \n", initial_value, df(initial_value));

    double sol = newton(initial_value);
    printf("Final Solution f(%.6f) = %lf \n", sol, f(sol));
    return 0;
}
```