/\*

Newton Raphson's Method

\*/

#include <math.h>

#include <stdio.h>

double f(double x) {

return 6 \* x \* x + 11 \* x - 35;

}

double f0(double x) {

return 12 \* x + 11;

}

int main() {

double xn, xn0;

double slope\_lim = 0.00001;

// calculating tolerance2

int noDigits = 4;

double t = 0.5 \* pow(10.0, -noDigits);

// input

printf("Enter initial approximation:: ");

scanf("%lf", &xn);

xn0 = xn;

printf("x\_i\t| f(x\_i)\n");

while (1) {

if (fabs(f0(xn0) < slope\_lim)) {

printf("Slope is too small. Stoping execution....");

return 0;

}

xn = xn0 - (f(xn0) / f0(xn0));

printf("%lf\t| %lf\n", xn, f(xn));

if (f(xn) == 0) {

printf("Result :: %lf", xn);

break;

} else if (fabs(xn0 - xn) < t) {

printf("Result:: %lf within %lf tolerance.", xn, t);

break;

}

xn0 = xn;

}

return 0;

}