

```

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
double funcao(double t){ // função dada para a tensao do cabo.  $h(t) = 20t * \sinh(3/2t) - 32$ . Retorna valor da função  $h(t)$ .
    double x= (3/(2*t));
    double h=((20*t)*(sinh(x)))-32);
    return h;
}
double dfuncao(double t){ // funcao derivada da função  $h(t) = 20t * \sinh(3/2t) - 32$ . Retorna valor da função  $h'(t)$ .
    double x= (3/(2*t));
    double h=(20*(sinh(x)));
    double y= ((30/t) * cosh(x));
    double v= h-y;
    return v;
}
double newton(double a){ // função que aplica o metodo de newton
    double x= a - (funcao(a)/dfuncao(a));
    return x;
}
double main(){
    double e=0.000000001, x0=0, a=2, fx0=0, mfx0=0; int i=1; // inicializando as variaveis com os valores.
    // erro, a é o chute inicial, x0 variavel que recebe o valor das raiz, f(x0) variavel para receber valor f(x0), mf(x0) é o modulo f(x0)
    x0=newton(a); // faz a primeira iteração gerando xo como primeira raiz
    fx0=funcao(x0); // retorna o valor f(x0);
    printf("Raiz= %.11f\n",x0); // imprimi o valor da primeira raiz gerada
    printf("fx0= %.11f\n",fx0); // imprimi o valor de f(x0)
    if(fx0 < 0){ // gera o modulo do valor de f(x0)
        mfx0= fx0 * -1;
    }
    else{
        mfx0= fx0;
    }
    printf("modulo f(x0)= %.11f\n\n",mfx0); // imprimi o valor do modulo de f(x0)
    while(mfx0 > e){ //repete enquanto condição de parada do metodo nao é satisfeita
        i++; // conta numero de interações
        a=0; a=x0; x0=0,fx0=0; // zera as variaveis para evitar lixo na memoria
        x0=newton(a); // chama a função metodo de newton para gerar a nova raiz x0
        fx0=funcao(x0); // chama a função que retorna o valor de f(x0)
        if(fx0 < 0){ // gera o modulo do valor de f(x0)
            mfx0= fx0 * -1;
        }
        else{
            mfx0= fx0;
        }
        printf("Raiz= %.11f\n",x0); // imprimi os valores da raiz, f(x0) e modulo de f(x0) a cada iterações realizada
        printf("fx0= %.11f\n",fx0);
        printf("modulo f(x0)= %.11f\n\n",mfx0);
    }
    printf("Precisao do erro= %.9f\n",e); // imprimi o valor do erro dado como entrada
    printf("Raiz= %.11f\n",x0); // imprimi o valor da raiz final
    printf("%d Numero de iteracoes\n",i); // imprimi o valor do numero de iterações realizadas pelo metodo.
}

```