

```

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
double funcao(double t){ // função dada para a tensão 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 bisseccao(double a, double b){ // função dada para aplicar o metodo da
bisseccao. retorna o valor x;
    double x= ((a+b)/2);
    return x;
}
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=0.001,a=2, b=3, x0=0, fx0=0 ,mab=0, ab=0,mfx0=0; int i=1; //
    inicializando as variaveis com os valores.
    // erro para o primeiro metodo e intervalos [a,b] que tambem serao utilizados,
    o restante sao apenas variaveis que receberam valores
    // das funções chamadas
    x0=bisseccao(a,b); // faz a primeira iteração gerando x0 como primeira raiz
    para o metodo da bisseccao
    fx0=funcao(x0); // retorna o valor f(x0);
    printf("Raiz= %.11f\n",x0); // imprimi o valor da primeira raiz encontrada
    printf("fx0= %.11f\n",fx0); // imprimi o valor de f(x0)
    ab=b-a; // gera b-a que é a condição de parada do metodo
    if(ab < 0){ // gera o modulo de b-a que é mab
        mab= ab * -1;
    }
    else{
        mab=ab;
    }
    printf("modulo |b-a|= %.11f\n\n",mab); // imprimi o valor do modulo b-a
    while(mab > e){ //repete enquanto condição de parada do metodo nao é
    satisfeita
        i++; // conta o numero de iterações
        if(fx0 < 0 && (funcao(a)) < 0 || fx0 > 0 && (funcao(a)) > 0 ){ // f(x0) tem
    sinal igual a f(a)?
            a=0; a=x0; // troca o intervalo a=x0
        }
        else{ // caso f(x0) tenha sinal igual f(b)
            b=0; b=x0; // troca o intervalo b=x0
        }
        x0=0,fx0=0; ab=0; // zera as variaveis para evitar lixo na memoria
        x0=bisseccao(a,b); // chama a função bisseccao para a nova raiz x0
        fx0=funcao(x0); // chama a função que retorna o valor de f(x0)
        ab=b-a; // gera b-a que é a condição de parada do metodo
        if(ab < 0){ // gera o modulo de b-a que é mab
            mab= ab * -1;
        }
        else{
            mab=ab;
        }
        printf("Raiz= %.11f\n",x0); // imprimi o valor da raiz, f(x0) e do |b-a| a

```

```

cada iteracao realizada.
    printf("fx0= %.11f\n",fx0);
    printf("modulo |b-a|= %.11f\n\n",mab);
}
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
iteracoes realizadas pelo metodo
printf("\n\n\n");                             // quebra linha
printf("troca de metodo *newton* \n\n");       // imprimi que agora começa outro
metodo
printf("\n\n\n"); // quebra linha
e=0; e=0.000000001; a=0; a=x0; x0=0; fx0=0; mfx0=0; i=0; i=1;
// inicializa as variaveis necessarias para a realizacao do metodo de newton,
os valores iniciados foram obtidos anteriormente pelo
// metodo da bissecao.
x0=newton(a); // faz a primeira iteracao gerando x0 como primeira raiz
fx0=funcao(x0); // retorna o valor f(x0);
printf("Raiz= %.11f\n",x0); // imprimi o valor da primeira raiz encontrada
pelo metodo de newton
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 condicao de parada do metodo de newton
nao é satisfeita
    i++; // conta numero de iteracoes
    a=0; a=x0; x0=0,fx0=0; // zera as variaveis para evitar lixo na memoria
    x0=newton(a); // chama a funcao do metodo de newton para a nova raiz x0
    fx0=funcao(x0); // chama a funcao 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 a cada iteracao o valor da raiz, f
(x0) e modulo de f(x0)
    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
encontrada
printf("%d Numero de iteracoes\n",i); // imprimi o valor do numero de
iteracoes realizadas pelo metodo
}

```