

```

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

double funcao(double t){ // função dada para velocidade do foguete.  $v(t)=200*\log(1600/(1600-(27*t)))+(9.8*t)-100$ ;
    double x= (1600-(27*t));
    double y=log(1600/x);
    double v=(200*y)+(9.8*t)-(100);
    return v;
}

double secante(double x0, double x1){ // Metodo da secante, retorna um valor x
para intervalo [a,b]
    double fx0=funcao(x0);
    double fx1=funcao(x1);
    double x2= ((x0*(fx1))-(x1*(fx0)))/((fx1)-(fx0));
    return x2;
}

double main(){
    double e=0.000000001,x0=7,x1=7.2,x2=0,fx2=0,mfx2=0; int i=1; // inicializando as
variaveis com os valores.
    // e=0.000000001 a=7, b=8, v=0, x0=0 ,fx0=0 ,mfx2=0 ; int i=1;
    // Erro, intervalo [a,b], variavel para receber valor da função v(t)
    x2=secante(x0,x1); // faz a primeira iteração gerando x2 como primeira raiz
    fx2=funcao(x2); // retorna o valor f(x2);
    printf("Raiz= %.11f\n",x2); // imprimi o valor da primeira raiz
    printf("fx2= %.11f\n",fx2); // imprimi o valor de f(x2)
    if(fx2 < 0){ // gera o modulo do valor de f(x2)
        mfx2= fx2 * -1;
    }
    else{
        mfx2= fx2;
    }
    printf("modulo fx2= %.11f\n\n",mfx2); // imprimi o valor do modulo de f(x2)
    while(mfx2 > e){ //repete enquanto condição de parada do metodo nao é
satisfeita
        i++; // contador de iterações
        x0=0; x0=x1; x1=0; x1=x2; x2=0; fx2=0; mfx2=0; // zera as variaveis para
evitar lixo na memoria
        x2= secante (x0,x1); // chama o metodo da secante para a nova raiz x2
        fx2=funcao(x2); // chama a função que retorna o valor de f(x2)
        if(fx2 < 0){ // gera o modulo do valor f(x0), que é a condição de
parada |f(x0)| < Erro.
            mfx2= fx2 * -1;
        }
        else{
            mfx2= fx2;
        }
        printf("Raiz= %.11f\n",x2); // a cada iteração imprimi os valores da
raiz, f(x2) e modulo de f(x2)
        printf("fx2= %.11f\n",fx2);
        printf("modulo fx2= %.11f\n\n",mfx2);
    }
    printf("Precisao do erro= %.9f\n",e); // imprimi o valor do erro dado como
entrada
    printf("Raiz= %.11f\n",x2); // imprimi o valor da raiz final
    printf("%d Numero de iteracoes\n",i); // imprimi o numero de iterações
realizadas pelo metodo
}

```