

```

#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 falsap(double a, double b){ // Metodo da falsa posicao, retorna um valor x
para intervalo [a,b]
    double fa=funcao(a);
    double fb=funcao(b);
    double x= ((a*(fb))-(b*(fa)))/((fb)-(fa)); ;
    return x;
}
double main(){
    double e=0.000000001,a=2,b=3,x0=0,fx0=0,mfx0=0; int i=1; // inicializando as
variaveis com os valores.
    // Erro, intervalo [a,b] variavel x0 que é a raiz, variavel fx0 que recebe o
valor da função f(x0), mfx0 modulo de f(x0)
    x0=falsap(a,b); // faz a primeira iteração gerando x0 como primeira raiz
    fx0=funcao(x0); // retorna o valor f(x0);
    printf("Raiz= %.11f\n",x0); // imprimir o valor da raiz e f(x0)
    printf("fx0= %.11f\n",fx0);
    if(fx0 < 0){ // gera o modulo do valor de f(x0)
        mfx0= fx0 * -1;
    }
    else{ // gera o modulo do valor de f(x0)
        mfx0= fx0;
    }
    printf("modulo fx0= %.11f\n\n",mfx0); // imprimir o valor do modulos de f(x0)

    while(mfx0 > e){ //repete enquanto o modulo do valor x0 aplicado na função
seja maior que o erro, condição de parada do metodo.
        i++;
        if(fx0 < 0 && (funcao(a)) < 0 || fx0 > 0 && (funcao(a)) > 0 ){ // f(x0) tem
sinal igual a f(a)?
            a=x0; // troca o intervalo a=x0
        }
        else{ // caso f(x0) tenha sinal igual f(b)
            b=x0; // troca o intervalo b=x0
        }
        x0=0,fx0=0; // zera as variaveis para evitar lixo na memoria
        x0=falsap(a,b); // chama a função falsa posicao para 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 f(x0), que é a condição de
parada |f(x0)| < Erro.
            mfx0= fx0 * -1;
        }
        else{
            mfx0= fx0;
        }
        printf("Raiz= %.11f\n",x0); // a cada iteração o valor da raiz, f
(x0) e modulo de f(x0) são impressos
        printf("fx0= %.11f\n",fx0);
        printf("modulo fx0= %.11f\n\n",mfx0);
    }
    printf("Precisao do erro= %.9f\n",e); // imprimir o erro dado como entrada.
    printf("Raiz= %.11f\n",x0); // imprimir o valor da raiz final
    printf("%d Numero de iteracoes\n",i); // imprimir o numero de iterações
realizadas pelo metodo.
}

```