

# Introdução à Programação - 1.o semestre de 2021

## Segundo Exercício Programa

### Cálculo da Raiz Quadrada

Neste EP2, você deve escrever um programa que leia  $n$  e  $\epsilon$  (valores do tipo double, sendo  $n \geq 1$  e  $\epsilon > 0$ ), e calcule a raiz quadrada de  $n$  com precisão  $\epsilon$ . Para este EP não podem ser usadas funções prontas para realizar o cálculo da raiz (como a função *sqrt* declarada em *math.h*). Sendo  $x$  a raiz de  $n$ , idealmente gostaríamos de determinar  $x$  de modo que:

$$x^2 - n = 0$$

Entretanto, na prática, iremos nos contentar em determinar um  $x$  tal que:

$$|x^2 - n| < \epsilon$$

Ou seja, se  $x^2$  é *aproximadamente* igual a  $n$ , então  $x$  é uma aproximação razoável de  $\sqrt{n}$ .

Para realizar o cálculo da raiz, você deve implementar o método da bissecção (ou dicotomia). Este método consiste em determinar uma solução candidata inicial e, iterativamente, refinar tal solução. A cada iteração, a solução candidata fica mais próxima da solução esperada, e após uma certa quantidade de iterações, a solução candidata estará próxima o suficiente do resultado desejado.

Inicialmente tomamos  $x = n / 2$  como solução candidata. Por que  $n / 2$ ? Por que temos certeza que a raiz quadrada de  $n$  é um valor entre 0 e  $n$  (observe que isto não é válido para  $n < 1$ ). Logo nosso primeiro “chute” de solução é um valor que está exatamente no meio deste intervalo. A partir deste “chute inicial” podemos ter 3 situações:

- 1) se  $x^2 - n \sim 0$  (ou seja,  $|x^2 - n| < \epsilon$ ), então tomamos  $x$  como aproximação boa o suficiente da raiz de  $n$ .
- 2) se  $x^2 - n > 0$  (ou seja  $x^2 > n$ ), isto significa que nosso chute está alto demais. Logo o valor real da raiz deve ser menor que  $x$ . Mais especificamente, a raiz certamente estará entre 0 e  $x$ .
- 3) se  $x^2 - n < 0$  (ou seja  $x^2 < n$ ), isto significa que nosso chute está baixo demais. Logo o valor real da raiz deve ser maior que  $x$ . Mais especificamente, a raiz certamente estará entre  $x$  e  $n$ .

Este processo ilustra a primeira iteração do método. Perceba que ao final desta primeira iteração, mesmo que a raiz de  $n$  não tenha sido encontrada, conseguimos reduzir pela metade nosso espaço de busca (isto é, o intervalo onde a raiz deve se encontrar). Desta forma, na iteração seguinte, o valor candidato a ser avaliado estará em um dos intervalos:  $[0, n/2]$  ou  $[n/2, n]$ .

Aplicando o processo repetidas vezes (ou seja, realizando várias iterações dos passos descritos, considerando sempre o novo intervalo de procura para determinar o próximo valor candidato), o intervalo onde certamente se encontra a raiz de  $n$  é reduzido cada vez mais. Após uma certa quantidade de iterações, ficará pequeno o suficiente de modo que o valor médio do intervalo resultante seja uma boa aproximação da raiz.

Para ilustrar o funcionamento completo do método, considere  $n = 15$ , e  $\epsilon = 0.1$ . Os intervalos considerados e os valores candidatos da raiz, calculados a cada iteração, serão:

```
000: intervalo = [ 0.00000000, 15.00000000]    x = 7.50000000
001: intervalo = [ 0.00000000,  7.50000000]    x = 3.75000000
002: intervalo = [ 3.75000000,  7.50000000]    x = 5.62500000
003: intervalo = [ 3.75000000,  5.62500000]    x = 4.68750000
004: intervalo = [ 3.75000000,  4.68750000]    x = 4.21875000
005: intervalo = [ 3.75000000,  4.21875000]    x = 3.98437500
006: intervalo = [ 3.75000000,  3.98437500]    x = 3.86718750
```

Na iteração número 6 (ou seja, na 7.<sup>a</sup> iteração, uma vez que a primeira é a iteração zero), para  $x = 3.86718750$  temos que  $|x^2 - n| = 0.0448608398$ , ou seja, a diferença absoluta é menor do que  $\epsilon$ . Portanto, tomamos este valor como a raiz quadrada aproximada de 15, dentro da margem de erro escolhida.

O programa desenvolvido por você deve gerar a seguinte saída, ao final da execução (linhas geradas antes destas três linhas finais serão ignoradas na validação da saída. Note também que não devem ser usados caracteres acentuados):

```
Numero de iteracoes: 7
Diferenca absoluta: 0.0448608398
Raiz quadrada: 3.8671875000
```

## Entrega

A entrega deste EP deve ser composta apenas pelo arquivo fonte (.c) do programa desenvolvido. A entrega deverá ser feita até o dia **17/07 às 23:59** pelo eDisciplinas. Não se esqueça de indicar (através de um comentário) seu nome e número USP no início do arquivo. Fique livre para adicionar outros comentários que julgar relevantes para o bom entendimento do código. Este EP deve ser feito de forma individual.

## Bônus

Tente escrever seu programa de modo que ele também consiga calcular corretamente a raiz para valores menores que 1. Ou seja, faça seu programa funcionar para  **$0 \leq n < 1$** .

Boa diversão! ;)