Segundo Exercício Programa

MAC0110 - BMAC Noturno 2011

Professor Roberto M Cesar Jr
Entrega até 26/05/2011

Raízes de Equações Quadráticas

Escreva um programa em C que calcule as raízes de equações quadráticas. Seu programa deve ler um real $\epsilon > 0$ que define a precisão para o cálculo de raízes quadradas, um inteiro $n \geq 1$ e os coeficientes reais a, b e c de n equações do segundo grau ($ax^2 + bx + c = 0$, com $a \neq 0$). O programa deve calcular as raízes de cada equação e imprimir os resultados da maneira especificada abaixo.

Entrada e impressão dos resultados

Para cada equação, deve-se ler os seus coeficientes e imprimir o tipo de suas raízes (reais simples, real dupla, ou complexas) e as raízes. No caso em que as raízes forem complexas, deve-se imprimir a parte real e a parte imaginária. Veja um exemplo de entrada e saída do programa:

```
Digite a precisao: 10e-3
Numero de equacoes: 5

Coeficientes da eq de segundo grau > a * x^2 + b * x + c
 - EQUACAO 1
 > a: 1
 > b: -2
 > c: -3
 > raizes simples: 3.000000 ; -1.000000
 - EQUACAO 2
 > a: 0
 > b: 10
 > c: 20
 *** ERRO: equacao nao e do segundo grau! ***
 - EQUACAO 3
 > a: 7.18
 > b: 43.75
 > c: -31.12
 > raizes simples: 0.643381 ; -6.736696
 - EQUACAO 4
 > a: 1
 > b: -2
 > c: 10
 > raizes complexas: 1.000000 + 3.000000i ; 1.000000 - 3.000000i
 - EQUACAO 5
 > a: 1
 > b: -2
 > c: 1
 > raiz dupla: 1.000000
```

Figura 1: Exemplo de entrada e saída para o programa de cálculo de raízes de equações quadráticas.

Cálculo da raiz quadrada de um número não negativo

Neste exercício você **não** usará a função `sqrt(x)` da biblioteca `math.h`. Seu programa deve ter **obrigatoriamente** uma função com o seguinte protótipo:

```
float raiz_quadrada (float x, float epsilon);
```

, que calcula a raiz quadrada de `x` usando método de Newton, descrito a seguir.

Suponha que desejamos extrair a raiz quadrada de um número real $x > 0$. Escolhe-se como chute inicial para $\sqrt{x}$ o número $r_0 = x$ e se calcula a seguinte sequência de números:

$$r_{n+1} = \frac{1}{2}\left(r_n + \frac{x}{r_n}\right)$$

, com $n = 0, 1, 2...$

Ou seja:

$$r_1 = \frac{1}{2}\left(x + \frac{x}{x}\right) = \frac{x+1}{2}$$

, e a partir de r_1 obtemos r_2 e assim por diante.

Esse processo deve ser repetido enquanto $|r_{n+1} - r_n| \geq \epsilon$, onde ϵ é um número positivo dado que representa a precisão do cálculo da raiz. A aproximação de $\sqrt{x}$ será o primeiro valor r_{n+1} para o qual $|r_{n+1} - r_n| < \epsilon$. Utilize como ϵ o valor o parâmetro `epsilon` passado à função `raiz_quadrada`.

Os tipos `double` e `long double`

Foi visto em classe apenas um tipo de dados para armazenar um número "real", o `float`, que é uma representação em ponto flutuante com precisão simples. A linguagem C tem outros dois tipos de "reais" que podem ser usados quando se precisa de maior precisão: `double` (precisão dupla) e `long double` (precisão extendida). Veja exemplos de utilização:

```
/* Declaracao de variaveis do tipo float, double e long double */
float f;
double d;
long double ld;

/* Constantes float, double e long double: */
f = 3.14f;
d = 3.14;       /* Ausencia de aufixo ''f'' ou ''L'' indica double */
ld = 3.14 L;

/* Impressao de float, double e long double */
printf (''%f %f %Lf'', f, d, ld);
```

Note que a leitura e a impressão de `double`'s são feitas com especificações de forma diferentes: a leitura é com "`%lf`" e a impressão é com "`%f`". Repare também que a mesma especificação de formato ("`%f`") é usada para imprimir tanto `float`'s quanto `double`'s. (A impressão com "`%f`" é, na verdade, sempre de `double`'s. Quando se imprime um `float` com "`%f`", o `float` é automaticamente "promovido" para `double` antes de ser passado como parâmetro à função `printf`.)

Faz diferença usar `float`, `double` ou `long double`?

Para verificar a influência do tipo "real" utilizado, faça os experimentos descritos a seguir. Use inicialmente uma versão do seu programa na qual todas as variáveis "reais" são do tipo `float`.

1. Rode o seu programa com valores cada vez menores do parâmetro ϵ. Comece com $\epsilon = 10^{-3}$, como na figura 1, depois tente usar potências de 10 cada vez menores: $\epsilon = 10^{-3}, 10^{-4}, 10^{-5}, ...$ Até qual precisão o programa funciona? O que acontece quando o programa deixa de funcionar?

2. Crie uma nova versão do seu programa que usa precião dupla (`double`) para cacular as raízes quadráticas. Nesta versão do programa, a função `raiz_quadrada` terá protótipo:

   ```
   float raiz_quadrada (double x, double epsilon);
   ```

   Empora essa versão da função devolva raiz quadrada como um `float`, internamente ela trabalha só com `double`. Você pode continuar usando `float`'s no restante do programa, exceto quando estiver lendo o valor de ϵ da entrada. Esse valor deve ser lido agora como `double`.

3. Repita o ítem 1 com a segunda versão do programa. Até que precisão essa versão do programa funciona?

4. Crie uma terceira versão do seu programa que usa precisão extendida (`long double`) para calcular raízes quadradas. Nesta versão do rograma a função `raiz_quadrada` terá protótipo:

   ```
   float raiz_quadrada (long double x, long double epsilon);
   ```

   Embora essa versão da função devolva a raiz quadrada como um `float`, internamente ela trabalha só com `long double`'s. Você pode continuar usando `float`'s no restante do programa, exceto quando estiver lendo o valor de ϵ da entrada. Esse valor deve ser lido agora como `long double`.

5. Repita o ítem 1 com a terceira versão do programa e descubra até qual precisão essa versão do programa funciona.

Que versão do programa deve ser entregue?

Entregue apenas a versão que usa a precisão dupla (`double`) para calcular raízes quadradas. Mas coloque no comentário no início do programa, logo após o cabeçalho com seu nome e número USP, suas respostas para as seguintes questões:

1. Até que precisão funcionou o programa só com `float`'s?

2. Até que precisão funcionou o programa que usa `double`'s para calcular as raízes quadradas?

3. Até que precisão funcionou o programa que usa `long double`'s para calcular as raízes quadradas?

4. Para cada uma das três versões do programa, explique também o que acontece quando a precisão é excessiva e o programa deixa de funcionar.

Instruções

Sobre a Elaboração

O EP pode ser elabarado por equipes de dois alunos, desde que as seguintes regras sejam respeitadas:

- Os alunos devem trabalhar sempre juntos, buscando a cooperação.
- Caso exista em um grupo um aluno com maior facilidade, este deve explicar as decisões tomadas, e o seu par deve participar e se esforçar para entender o desenvolvimento do programa. (denominamos isso de *programação em pares*, uma excelente prática que vocês devem se esforçar para adotar).
- Mesmo a digitação do EP deve ser feita em grupo, enquanto um digita, o outro deve acompanha.
- Recomendamos fortemente que o exercício seja desenvolvido conforme a descrição nos ítens acima, mas ele também pode ser feito individualmente.

Sobre a Avaliação

- É sua responsabilidade manter o código do seu EP em sigilo, ou seja, apenas você e seu par podem ter acesso ao código.
- No caso de EPs feitos em dupla, a mesma nota será atribuída aos dois alunos do grupo.
- **Não serão toleradas cópias!** Exercícios copiados (com eventuais disfarces) levarão à reprovação da disciplina e ao encaminhamento do caso para a Comissão de Graduação.
- Exercícios atrasados não serão aceitos.
- Exercícios com erro de sintaxe (ou seja, erros de compilação) receberão nota ZERO.
- É muito importante que seu programa tenha comentários e esteja bem indentado, ou seja, digitado de maneira a ressaltar a estrutura de subordinação dos comandos do programa. Isto afetará a sua nota.
- Caso o programa apresente resultados "estranhos" (inesperados) para eventuais dados de entrada "incorretos", haverá descontado de nota.
- É importante que o exercício programa siga as instruções do enunciado e faça tudo da maneira que ele pede. Caso isso não aconteça, haverá desconto de nota.
- Uma regra básica é a seguinte: do ponto de vista do monitor responsável pela correção dos trabalhos, quanto mais convenientemente apresentado estiver o seu programa, melhor avaliado ele será.

Sobre a entrega

- O prazo de entrega é até o dia 26/05/2011.
- Caso feito em dupla: **Ambos devem submeter a versão final do código!**
- Entregar apenas um arquivo com o nome *raizes.c*.

- Para a entrega, utilize o Paca. Você pode entregar várias versões de um mesmo EP até o término do prazo, mas somente a última versão que permanecerá armazenada pelo sistema.
- Não serão aceitas submissões por e-mail ou atrasadas. Não deixe para a última hora, pois o sistema pode ficar congestionado, e você corre o risco de não conseguir enviar.
- Guarde uma cópia do seu EP pelo menos até o final do semestre.
- No início do arquivo, acrescente o seguinte cabeçalho:

```
/*****************************************************************/
/** MAC0110 BMAC IME Noturno 2011                                **/
/** Prof Roberto Cesar                                           **/
/**                                                              **/
/** Segundo Exercicio Programa -- Eq Segundo Grau                **/
/**                                                              **/
/** <nome do(a) aluno(a)> <numero USP>                           **/
/** <nome do(a) aluno(a)> <numero USP>                           **/
/**                                                              **/
/** Informacoes sobre Desenvolvimento:                           **/
/**  <Ambiente (CodeBlocks, Gedit, Dev-C++, ...)>                **/
/**  <Sistema Operacional (Windows, Linux, ...)>                 **/
/**                                                              **/
/** Repostas:                                                    **/
/** 1)...                                                        **/
/** 2)...                                                        **/
/** 3)...                                                        **/
/** 4)...                                                        **/
/**                                                              **/
/*****************************************************************/
```