

**Instituto de Computação da UNICAMP****Disciplina MC102: Algoritmos e Programação de Computadores - Turmas EF****Laboratório Nº 07. Peso 1.**

Prazo de entrega: **10/04/2015 às 23:59:59**

PROFESSOR: Alexandre Xavier Falcão

MONITORES: João do Monte Gomes Duarte

MONITORES: Jadisha Yarif Ramírez Cornejo

MONITORES: Takeo Akabane

MONITORES: Eduardo Spagnol Rossi

MONITORES: Guilherme Augusto Sakai Yoshike

---

**Calculadora de Equações e Propriedades**

Neste laboratório vamos exercitar o uso de funções.

É importante aprender a utilidade das funções pois isto vai tornar seus programas mais organizados e evitar erros no futuro. Esta também é uma boa forma de reutilizar pedaços de códigos que você já escreveu (e já verificou que estão corretos) em outros programas.

Vamos organizar uma calculadora que calcula as funções seno, cosseno ou raiz quadrada usando sequências, não vamos usar a biblioteca `math.h`. O seu programa deve receber uma ordem entre as seguintes possibilidades e calcular o que for pedido:

- Comando Seno (indicado pela letra 'S'): deve-se calcular a função seno em um dado ponto. Esta aproximação deve ser feita usando a série de Maclaurin (Série de Taylor no ponto 0). Deverá ser lido o ponto 'x', do tipo **double**, em que a função deve ser calculada e o índice 'n' que limita até onde a série deve ser calculada. Para simplificar vamos definir a série como a figura abaixo. Com o valor calculado, você deve exibir a mensagem "O valor aproximado de seno no ponto 'x' eh 'y'.", onde 'y' é o valor calculado.

$$\text{sen}(x) = \sum_{i=0}^n \frac{(-1)^i}{(2i+1)!} x^{2i+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots$$

- Comando Cosseno (indicado pela letra 'C'): deve-se calcular a função cosseno em um dado ponto. Esta aproximação deve ser feita usando a série de Maclaurin (Série de Taylor no ponto 0). Deverá ser lido o ponto 'x', do tipo **double**, em que a função deve ser calculada e o índice 'n' que limita até onde a série deve ser calculada. Para simplificar vamos definir a série como a figura abaixo. Com o valor calculado, você deve exibir a mensagem "O valor aproximado de cosseno no ponto 'x' eh 'y'.", onde 'y' é o valor calculado.

$$\text{cos}(x) = \sum_{i=0}^n \frac{(-1)^i}{(2i)!} x^{2i} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots$$

- Comando Raiz Quadrada (indicado pela letra 'R'): deve-se calcular uma aproximação da raiz quadrada de um número usando o método de Newton. A função recebe como entrada o valor 'x', do tipo **double**, depois um índice 'n', a raiz quadrada de 'x' deve ser calculada usando 'n' iterações do método de Newton. A cada iteração o método de Newton calcula aproximações repetidas para a raiz quadrada. A primeira aproximação deverá ser  $y_0 = 1$ . As demais são calculadas até a 'n'-ésima aproximação, conforme a fórmula abaixo, até o índice 'n' dado. Com o valor calculado, você deve exibir a mensagem "O valor aproximado da raiz de 'x' eh 'y'.", onde 'y' é o valor calculado.

$$y_{n+1} = y_n - \frac{y_n^2 - x}{2y_n}$$

Cada um destes cálculos devem ser implementados em uma função separada que será descrita a seguir. A sua calculadora deve funcionar, recebendo comandos e fazendo cálculos, até que o usuário entre com o comando 'f', ou 'F'. Caso o usuário entre com algum comando inválido, o programa deve avisar "Comando invalido. Entre com outro comando, ou 'F' para terminar."

Além destes comandos, você deve também implementar uma função para calcular potências inteiras não negativas de um número qualquer e uma para calcular o fatorial de um número. Você pode implementar também qualquer outra função que julgar necessária. Estas funções vão ajudar o seu programa a ficar mais organizado.

## Implementação

Você deve escrever as seguintes funções:

```
/*Esta funcao calcula o fatorial de um numero inteiro.*/
/*Recebe um inteiro e retorna o fatorial deste numero como um numero inteiro.*/
int fatorial(int a);

/*Esta funcao calcula a potencia inteira positiva de um numero float.*/
/*Recebe um float e um inteiro e calcula o valor do float elevado ao inteiro.*/
/*Retorna o valor do calculo em float.*/
float potencia(float x, int n);

/*Esta funcao calcula um valor aproximado para a funcao seno em um ponto proximo de 0.*/
/*Recebe um numero do tipo double e um limite de iteracoes para calcular o valor de*/
/* seno no ponto do double fazendo n+1 (i=0 até i=n) iteracoes na série de Maclaurin.*/
double seno(double x, int n);

/*Esta funcao calcula um valor aproximado para a funcao cosseno em um ponto proximo de 0.*/
/*Recebe um numero do tipo double e um limite de iteracoes para calcular o valor de*/
/* cosseno no ponto do double fazendo n+1 (i=0 até i=n) iteracoes na serie de Maclaurin.*/
double cosseno(double x, int n);

/*Esta funcao calcula um valor aproximado para a raiz quadrada de um numero usando o metodo de Newton*/
/* ate a n-esima iteracao.*/
/*Recebe como parametros um numero cuja raiz deve ser calculada, e o valor de n indicando o numero de*/
/* iteracoes do metodo de Newton*/
/*Utilize como valor inicial para a primeira aproximação da raiz o valor y_0 = 1 */
double raizQuadrada(double x, int n);
```

O seu programa deve ler da entrada padrão um dos comandos listados: 'S' para Seno, 'C' para Cosseno, e 'R' para Raiz quadrada. O programa deve fazer a operação exigida, escrever o resultado na tela e esperar pelo próximo comando. O comando 'f', ou 'F' termina o programa sem nenhuma mensagem. Caso um comando inválido seja pedido, o programa deve dar uma mensagem de erro "Comando invalido. Entre com outro comando, ou 'F' para terminar." e esperar pelo próximo comando.

## Dicas

Lembre-se de testar todas as funções separadamente enquanto escreve o programa. Comece pelas mais fáceis e menos dependentes. Por exemplo, a função para calcular seno usa fatorial, então faça primeiro a função que calcula fatorial, e **teste** esta função. Isto vai evitar erros e economizar o seu tempo.

Ao usar a função `scanf` para ler um caractere (com a máscara `%c`), deve-se acrescentar um espaço antes de `%c` para garantir que somente um caracter válido seja lido (ou seja, para ignorar espaços e quebras de linhas).

- Por exemplo, para um char `x`, o comando `scanf("%c",&x)`; assinalaria para `x` o caractere de espaço ' ' para a entrada " A " (note o espaço na entrada!) enquanto que `scanf(" %c",&x)`; (note o espaço antes de `%c`) assinalaria

'A' para a variável x.

**Observação Importante:** não é permitido o uso da biblioteca `math.h`, só a biblioteca `stdio.h` poderá ser usada.

### Entrada

O programa lê, da entrada padrão, um caractere como comando e uma quantidade de números inteiros que dependem do comando pedido.

### Saída

O programa podem imprimir as seguintes mensagens:

"O valor aproximado de seno no ponto  $\langle x \rangle$  eh  $\langle y \rangle$ ." para o comando 'S', o tipo **double**  $\langle x \rangle$  e o inteiro  $\langle i \rangle$  como entrada.

"O valor aproximado de cosseno no ponto  $\langle x \rangle$  eh  $\langle y \rangle$ ." para o comando 'C', o tipo **double**  $\langle x \rangle$  e o inteiro  $\langle i \rangle$  como entrada.

"O valor aproximado da raiz de  $\langle x \rangle$  eh  $\langle y \rangle$ ." para o comando 'R', o tipo **double**  $\langle x \rangle$  e um inteiro  $\langle n \rangle$  como entrada.

### Exemplos de execução

#### Exemplo 1

```
R
2
10
O valor aproximado da raiz de 2.0000 eh 1.4142.
C
3.14159
5
O valor aproximado de cosseno no ponto 3.1416 eh -1.0018.
S
3.14159
5
O valor aproximado de seno no ponto 3.1416 eh -0.0004.
F
```

#### Exemplo 2

```
p
Comando invalido. Entre com outro comando, ou 'F' para terminar.
a
Comando invalido. Entre com outro comando, ou 'F' para terminar.
f
```

*Nota:* Textos em azul denotam dados de entrada, isto é, a serem lidos pelo programa.

Textos em vermelho denotam dados de saída, ou seja, a serem impressos pelo programa.

### Observações

- Não é permitido o uso da biblioteca `math.h`, a única biblioteca permitida é a `stdio.h`;
- O número máximo de submissões é 15;
- O seu programa deve estar completamente contido em um único arquivo denominado `calculadora.c`;
- Para a realização dos testes do SuSy, a compilação se dará da seguinte forma: `gcc -std=c99 -pedantic -Wall -lm -o calculadora calculadora.c`;

- Você deve incluir, no início do seu programa, uma breve descrição dos objetivos do programa, da entrada e da saída, além do seu nome e do seu RA.

**Critérios importantes**

Independentemente dos resultados dos testes do SuSy, o não cumprimento dos critérios abaixo implicará nota zero nesta tarefa de laboratório.

- Não serão aceitas soluções que usem a biblioteca math.h, a única biblioteca permitida é a stdio.h;

---

[Voltar](#)