

MC202GH - Estrutura de Dados - Turmas G e H

Laboratório 1 - Programando em C

Docente: Marcelo da Silva Reis

Monitor PED: Matheus Abrantes Cerqueira

Monitor PAD: Andreas Cisi Ramos

Monitor PAD: Wallace Gustavo Santos Lima

25 de agosto de 2022

Data de entrega: 31/08/2022

Entrega no codePost¹

Informações gerais

Este laboratório tem como objetivo a primeira prática em C. Para isso serão fornecidos os arquivos protótipos, os quais devem ser modificados e enviados para a plataforma codePost. Cada questão tem seu próprio arquivo com função *main*, porém cada questão também tem uma função a ser desenvolvida dentro do arquivo *funcoes.c*, as quais não devem produzir saídas de texto, reduzindo as entradas e saídas de texto à função *main()* de cada questão.

Importante: nesse laboratório não será permitido o uso da biblioteca *math.h*. Além disso, para compilar os códigos usando a biblioteca *funcoes.h* use o seguinte:

```
gcc funcoes.c <program>.c -o <program> -I. funcoes.h
```

dentro do diretório que contenha esses arquivos, sendo que *<program>* corresponde ao arquivo de cada questão.

Questão 1 (1 pt.) - Verificador de números primos

Números primos são aqueles que são divisíveis apenas por si mesmo e por 1, por exemplo: 2,3,5,7 e 11. O objetivo dessa questão é completar o programa *primo.c*, que irá pedir por um número e avaliar se ele é primo ou não, apresentando "Numero primo!" ou "Numero nao primo!", assim como mostra a figura 1.

Importante, para isso complete o protótipo

```
int is_primo(int numero);
```

o qual recebe um inteiro e retorna um outro inteiro cujo valor é "1" se o valor da variável *numero* for um numero primo e "0" caso contrário.

¹<https://codepost.io/signup/join?code=ZW239C3IID>

```

root@1e18f9a9cd9a:/app/det# ./primo
Digite um numero:1
Numero nao primo!
root@1e18f9a9cd9a:/app/det# ./primo
Digite um numero:4
Numero nao primo!
root@1e18f9a9cd9a:/app/det# ./primo
Digite um numero:7
Numero primo!
root@1e18f9a9cd9a:/app/det# █

```

Figura 1: Execução do código que verifica se um número é primo.

Questão 2 (3 pts.) - Calculadora de hipotenusa

Complete o programa *hipotenusa.c* que calcula a hipotenusa dada a entrada de seus dois catetos. Aqui diferente da questão anterior é esperado que o programa calcule múltiplas vezes até o usuário cancelar (por meio de um cateto negativo, qualquer que seja):

Complete o protótipo

float hipotenusa(float c1, float c2);

que recebe dois catetos (c1 e c2) na forma de variáveis em ponto flutuante (*float*), retornando o valor da hipotenusa em outro *float*. Além disso é esperado que a hipotenusa tenha dois dígitos de precisão de saída.

```

matheus lab1 $ ./hipotenusa
Por favor, digite os dois catetos (ou dois números negativos para parar):
3 4
hipotenusa eh 5.00
5 5
hipotenusa eh 7.07
-1 1

```

Figura 2: Exemplo de execução da calculadora de hipotenusas.

Questão 3 (6 pts.) - Determinante de Matrizes

Escreva um programa (em *determinant.c*) que recebe 4 inteiros pela entrada padrão, os aloca em uma matriz quadrada de tamanho 2x2 e computa seu determinante, verificando se o mesmo é positivo, negativo ou nulo. Para os casos positivos o programa deve também calcular a hipotenusa do triângulo que possui área igual a metade do determinante obtido.

A figura 3 apresenta o conceito do triângulo contido dentro do quadrado de área equivalente ao determinante positivo. Já as figuras 4, 5 e 6 apresentam exemplos de execuções do código.

Complete o protótipo

float cateto_from_matrix(int M[2][2]);

que recebe a matriz 2x2 na forma de um vetor 2d de inteiros. A função deve então computar o determinante da matriz e a hipotenusa para os casos de determinante positivo. No fim a função retorna um *float* com o valor da hipotenusa (para os casos positivos), um valor qualquer negativo para determinantes negativos e zero para determinantes nulos.

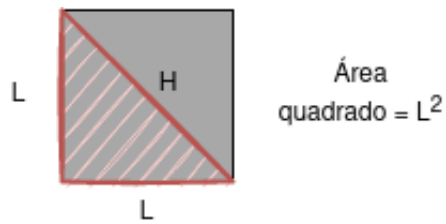


Figura 3: Ilustração do triângulo retângulo contido no quadrado de área equivalente ao determinante.

```
root@1e18f9a9cd9a:/app/det# ./determinant
Digite os 4 numeros:3 0 1 4
Determinante Positivo!
Hipotenusa do triangulo retangulo igual a 4.90
root@1e18f9a9cd9a:/app/det#
```

Figura 4: Execução do programa para uma matriz 2x2 de determinante positivo.

```
root@1e18f9a9cd9a:/app/det# ./determinant
Digite os 4 numeros:0 0 0 0
Determinante Nulo!
root@1e18f9a9cd9a:/app/det#
```

Figura 5: Execução do programa para uma matriz 2x2 de determinante nulo.

```
root@1e18f9a9cd9a:/app/det# ./determinant
Digite os 4 numeros:1 3 4 0
Determinante Negativo!
root@1e18f9a9cd9a:/app/det#
```

Figura 6: Execução do programa para uma matriz 2x2 de determinante negativo.