

# Programação Estruturada (A1 e A2 diurno)

[Página inicial](#) / [Meus cursos](#) / [PE2018\\_DA1DA2](#) / [Semana 07](#) / [Lab 07a - DON'T PANIC](#) / [Descrição](#)

Descrição

[Visualizar envios](#)

## Lab 07a - DON'T PANIC

Disponível a partir de: quinta, 1 Nov 2018, 07:00

Disponível até: segunda, 12 Nov 2018, 02:00

Arquivos requeridos: lab07a.c ([Baixar](#))

Tipo de trabalho: Trabalho individual

UFABC - CMCC

MCTA028-15 - Programação Estruturada

## Laboratório 07a - DON'T PANIC

Prazo de entrega: 11/11/2018

Peso: 20

### Descrição

*A razão de o Guia do Mochileiro das Galáxias ser o livro de maior sucesso da galáxia é, em primeiro lugar, por ser menor e ligeiramente mais barato do que a Enciclopédia Galáctica e, em segundo lugar, por trazer a frase "Não Entre em Pânico" em letras garrafais e amigáveis escrita na capa.*

Você está prestes a começar uma das maiores aventuras de sua vida: se tornar um mochileiro das galáxias.

Obviamente, sua mochila contém um exemplar do livro mais importante que você irá precisar.

É claro que ela também contém uma toalha, né?

*A toalha é um dos objetos mais úteis para um mochileiro interestelar. Em parte devido a seu valor prático: você pode usar a toalha como agasalho; pode deitar-se sobre ela (como uma canga); pode dormir debaixo dela (como cobertor); pode umedecê-la e utilizá-la como arma; enrolá-la em torno da cabeça e proteger do sol, ou nos olhos, protegendo da areia em locais desérticos; pode agitar a toalha em situações de emergência para pedir socorro, etc. E entre inúmeras outras funções, naturalmente, pode usá-la para enxugar-se se ainda estiver razoavelmente limpa.*

Eventualmente, será interessante conseguir um peixe babel também.

*O Peixe Babel é pequeno, amarelo e semelhante a uma sanguessuga, e é provavelmente a criatura mais estranha em todo o Universo. Alimenta-se de energia mental, não daquele que o hospeda, mas das criaturas ao redor dele. Absorve todas as frequências mentais inconscientes desta energia mental e se alimenta delas, e depois expela na mente de seu hospedeiro uma matriz telepática formada pela combinação das frequências mentais conscientes com os impulsos nervosos captados dos centros cerebrais responsáveis pela fala do cérebro que os emitiu. Na prática, o efeito disto é o seguinte: se você introduz no ouvido um peixe-babel, você compreende imediatamente tudo o que lhe for dito em qualquer língua. Os padrões sonoros que você ouve decodificam a matriz de energia mental que o seu Peixe Babel transmitiu para sua mente.*

Agora, você deveria estar verdadeiramente preocupado com as medidas.

*Ao longo do tempo, foi registrado o uso de diversas formas de medidas utilizadas pelos povos antigos. Os egípcios, por exemplo, utilizavam o palmo e o cúbito há 4 mil anos. Porém, nos diferentes territórios e países, os meios e as medidas usadas no dia a dia eram diferentes, assim dificultando o comércio internacional. Com o passar do tempo, e com a evidente necessidade de facilitar o comércio entre as pessoas e as nações, foi criado apenas em 1960, depois de inúmeras convenções internacionais com representantes de diversos países, o Sistema Internacional de Unidades (SI).*

Não existe padrão [nem na Terra](#), quem dirá na Galáxia?

Você provavelmente começará a ter que medir comprimento em anos-luz. E as amigas que fizer provavelmente usam unidades como milhas, fahrenheit ou barris!

Mas, NÃO ENTRE EM PÂNICO!

Basta que você utilize todos os conhecimentos que você viu na sua disciplina favorita e escrever um programa para te ajudar com isso!

## Implementação

Você já viu que `stdio`, `math` e `string` são *bibliotecas* da linguagem C.

Isso significa que alguma alma bondosa, em algum momento de sua vida, escreveu as funções `printf`, `scanf`, `sqrt`, `strlen` e tantas outras.

Essas funções foram definidas em arquivos `.c` e os seus protótipos foram colocados em arquivos `.h`.

Ou seja, quando fazemos `#include <stdio.h>`, o que estamos fazendo é indicar ao compilador que todos os protótipos de funções que estiverem no arquivo de nome `stdio.h` devem ser incluídos no nosso programa.

Durante a compilação, ele se encarrega de encontrar as definições dessas funções para poder executá-las.

O objetivo deste laboratório é criar a sua biblioteca de funções em C para realizar operações de conversão de unidades de medida.

A biblioteca que você irá implementar deve conter implementações para funções que convertam medidas de peso, comprimento, temperatura, volume e velocidade.

Cada função irá receber um indicador para a unidade atual. A função deve converter o valor recebido para todas as outras unidades. Por exemplo, se a função de temperatura receber o indicativo de Celsius, significa que há um valor em graus Celsius que deve ser convertido para Fahrenheit e para Kelvin.

Mais especificamente, teremos as funções:

- peso: para converter entre as unidades
  - quilograma [*kilogram*]
  - libra [*pound (troy)*]
  - onça [*ounce (US food nutrition labelling)*]
- comprimento: para converter entre as unidades
  - quilômetro [*1000 metre (SI base unit)*]
  - milha [*mile (international)*]
  - pé [*foot (International)*]
  - légua [*league (land)*]
- temperatura: para converter entre as unidades
  - Celsius [*degree Celsius*]
  - Fahrenheit [*degree Fahrenheit*]
  - kelvin [*kelvin (SI base unit)*]
- volume: para converter entre as unidades
  - metro cúbico [*cubic metre (SI unit)*]
  - litro [*litre*]
  - galão [*gallon (imperial)*]
  - barril [*barrel (imperial)*]
  - onça fluída [*ounce (fluid US food nutrition labeling)*]
- velocidade: para converter entre as unidades
  - km/h (quilômetros por hora) [*kilometre per hour*]
  - m/s (metros por segundo) [*metre per second (SI unit)*]
  - mph (milhas por hora) [*mile per hour*]

A tabela de conversões que deve ser usada como referência é [essa](#). Os nomes entre [ e ]

acima devem ser usados para acessar a tabela. Utilize os números exatamente como eles estão nessa tabela para não haver problemas de precisão. Dízimas periódicas que aparecem com notação de traço superior devem ser utilizadas como estão também: sem repetições (isto é, apenas desconsidere o traço). Por exemplo, para indicar a correspondência entre 1 km/h e m/s, utilize o valor  $2.7 \times 10^{-1}$ , e não  $2.77 \times 10^{-1}$  ou  $2.777777 \times 10^{-1}$ .

OBS: Os protótipos exatos dessas funções e a explicação de cada parâmetro estão no arquivo `lab07a.h`. O arquivo `main.c` é fornecido para você e faz uso dessas funções (você não deve modificá-lo). Você deve apenas implementar as funções em um arquivo chamado `lab07a.c`, que é o único arquivo que deve ser submetido. Seu arquivo não deve conter uma função `main()`.

## Múltiplos Arquivos e Função Principal

Neste laboratório vamos utilizar o conceito de dividir o código em múltiplos arquivos.

Quando se implementa programas grandes é comum separar o código em vários arquivos com a extensão `.c`, onde cada arquivo implementa um conjunto de funções relacionadas entre si e apenas um deles contém uma função `main()`.

Isto facilita a manutenção e a leitura do código.

Para compilar um código organizado dessa forma, basta passar todos os arquivos na linha de comando para o compilador.

Nesse laboratório, em particular, você deve usar a seguinte linha de comando:

```
gcc -ansi -pedantic -Wall -Werror -o lab07a lab07a.c main.c
```

Nesse laboratório, no entanto, para compilar o seu programa basta adicionar o arquivo extra que provemos (`lab07a_main.c`) ao final da linha de comando, como no exemplo a seguir:

```
gcc -o lab07a lab07a.c lab07a_main.c
```

Você encontra os arquivos auxiliares aqui: [http://professor.ufabc.edu.br/~e.francesquini/pe/files/testes\\_abertos/](http://professor.ufabc.edu.br/~e.francesquini/pe/files/testes_abertos/).

A organização do conteúdo de cada arquivo é a seguinte:

- `lab07a.c` :
  - funções auxiliares que você queira escrever;
  - `peso(...)` ;
  - `comprimento(...)` ;
  - `temperatura(...)` ;
  - `volume(...)` ;
  - `velocidade(...)` ;
- `main.c` :
  - funções auxiliares para a main;
  - `main()` .

# Reforçando

Neste laboratório você não precisará se preocupar em ler a entrada a partir da entrada padrão, nem em escrever a saída.

Seu trabalho é apenas implementar as funções descritas. A função `main()` que é fornecida no arquivo `main.c` se encarrega dessa parte.

Você também não deve submeter o arquivo `main.c`, somente o arquivo `lab07a.c`.

As sessões abaixo, de [Entrada](#) e [Saída](#), descrevem os formatos de entrada e saída, mas você não precisa se preocupar com eles.

## Entrada

A entrada consiste de várias operações de conversão entre unidades de medidas.

As operações de conversão são as seguintes:

- `P x u` : fazer conversão do valor `x` que está na unidade de medida `u` para todas as outras unidades de peso.
- `C x u` : fazer conversão do valor `x` que está na unidade de medida `u` para todas as outras unidades de comprimento.
- `T x u` : fazer conversão do valor `x` que está na unidade de medida `u` para todas as outras unidades de temperatura.
- `V x u` : fazer conversão do valor `x` que está na unidade de medida `u` para todas as outras unidades de volume.
- `S x u` : fazer conversão do valor `x` que está na unidade de medida `u` para todas as outras unidades de velocidade.
- `Q` : encerrar o programa.

## Saída

Cada linha da saída do programa contém o resultado da execução de cada operação dada na entrada, de forma que a saída possui uma linha a menos que a quantidade de linhas da entrada.

## Exemplos

Exemplo 1:

Entrada

```
P 500 quilogramas
P 500 libras
P 500 oncas
C 5 quilometros
C 5 milhas
C 5 pes
C 5 leguas
```

T 30 celsius  
T 30 fahrenheit  
T 30 kelvin  
V 4 m3  
V 4 litros  
V 4 galoos  
V 4 barris  
V 4 oncas  
S 80 km/h  
S 80 m/s  
S 80 mph  
Q

## Saída

500.0000 quilos equivalem a 1339.6144 libras, que equivalem a 17857.1429 oncas.  
186.6209 quilos equivalem a 500.0000 libras, que equivalem a 6665.0307 oncas.  
14.0000 quilos equivalem a 37.5092 libras, que equivalem a 500.0000 oncas.  
5.0000 quilometros equivalem a 3.1069 milhas, que equivalem a 16404.1995 pes, que equivalem a 1.0356 leguas.  
8.0467 quilometros equivalem a 5.0000 milhas, que equivalem a 26400.0000 pes, que equivalem a 1.6667 leguas.  
0.0015 quilometros equivalem a 0.0009 milhas, que equivalem a 5.0000 pes, que equivalem a 0.0003 leguas.  
24.1400 quilometros equivalem a 14.9999 milhas, que equivalem a 79199.4751 pes, que equivalem a 5.0000 leguas.  
30.0000 graus Celsius equivalem a 86.0000 graus Fahrenheit, que equivalem a 303.1500 kelvin.  
-1.1111 graus Celsius equivalem a 30.0000 graus Fahrenheit, que equivalem a 272.0389 kelvin.  
-243.1500 graus Celsius equivalem a -405.3700 graus Fahrenheit, que equivalem a 30.0000 kelvin.  
4.0000 metros cubicos equivalem a 4000.0000 litros, que equivalem a 879.8770 galoos, que equivalem a 24.4410 barris, que equivalem a 133333.3333 oncas fluidas.  
0.0040 metros cubicos equivalem a 4.0000 litros, que equivalem a 0.8799 galoos, que equivalem a 0.0244 barris, que equivalem a 133.3333 oncas fluidas.  
0.0182 metros cubicos equivalem a 18.1844 litros, que equivalem a 4.0000 galoos, que equivalem a 0.1111 barris, que equivalem a 606.1453 oncas fluidas.  
0.6546 metros cubicos equivalem a 654.6370 litros, que equivalem a 144.0000 galoos, que equivalem a 4.0000 barris, que equivalem a 21821.2320 oncas fluidas.  
0.0001 metros cubicos equivalem a 0.1200 litros, que equivalem a 0.0264 galoos, que equivalem a 0.0007 barris, que equivalem a 4.0000 oncas fluidas.  
80.0000 km/h equivalem a 21.6000 m/s, que equivalem a 48.3178 mph.  
296.2963 km/h equivalem a 80.0000 m/s, que equivalem a 178.9549 mph.

132.4563 km/h equivalem a 35.7632 m/s, que equivalem a 80.0000 mph.

Para mais exemplos, consulte os testes abertos em [http://professor.ufabc.edu.br/~e.francesquini/pe/files/testes\\_abertos/](http://professor.ufabc.edu.br/~e.francesquini/pe/files/testes_abertos/).

## Observações

- Você não deve submeter o arquivo `main.c`, somente o arquivo `lab07a.c`.
- O número máximo de submissões é 15.
- Para a realização dos testes, a compilação dos programas desenvolvidos em C irá considerar o comando:  
`gcc -ansi -pedantic -Wall -Werror -o lab07a lab07a.c main.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.
- Indente corretamente o seu código e inclua comentários no decorrer do seu programa.

## Critérios importantes

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

- Os únicos headers aceitos para essa tarefa são `stdio.h`, `math.h` e `lab07a.h`.

VPL

Você acessou como [Lucas Fernandes Muniz \(Sair\)](#)  
[PE2018\\_DA1DA2](#)

[Português - Brasil \(pt\\_br\)](#)  
[English \(en\)](#)

[Português - Brasil \(pt\\_br\)](#)