

Programação de Software Básico
Lista 1 de exercícios: Introdução à C, Funções
Prof. Marcelo Cohen

20 de agosto de 2016

1. Descubra quantos bits são usados no seu compilador preferido para armazenar os tipos `short int`, `int`, `long int` e `long long int`. Use o operador `sizeof(x)` para isso, onde x pode ser uma variável ou mesmo um tipo de dado.
2. Escreva uma função que faz a conversão de temperaturas de Fahrenheit para Celsius, segundo a fórmula conhecida:

$$C = \frac{5}{9} * (F - 32)$$

Depois de pronta, faça um programa para descobrir em que temperatura as duas escalas coincidem. Use papel, lápis e um pouco de matemática do ensino fundamental e confirme o resultado. Analise um pouco antes de programar e pense nisso: se você escrever a fórmula de conversão (na maioria das linguagens de programação) exatamente como ela foi dada acima, o resultado da conversão será sempre zero. Explique o motivo e corrija.

3. Escreva uma função que recebe um horário (horas, minutos, segundos) e determina quantos segundos já se passaram desde que o dia começou.
4. Crie outra função similar à função do programa do exercício 3 para determinar também quantos segundos ainda faltam até o dia terminar.
5. Um método para cálculo de raízes quadradas de um número N já era conhecido pelos babilônios em... bom, há muito tempo atrás. (Este método também é conhecido como método de Heron, um matemático grego que o descreveu 20 séculos depois dos babilônios, perto do ano 50 DC.) Começando de um valor inicial k (geralmente valendo um) os babilônios geravam um novo valor de k de acordo com a regra:

$$k = \frac{k + N/k}{2}.$$

À medida que o processo é repetido, os novos valores de k se aproximam cada vez mais da raiz de N . Escreva um programa que leia o valor de N e imprima os primeiros doze valores obtidos com esta fórmula, testando-os para ver se eles realmente se aproximam.

6. Com base no programa anterior, faça com que o programa termine a repetição quando uma aproximação aceitável seja atingida. Por exemplo, considere que se desejamos calcular a raiz de 25: se o resultado atingido for 5.001, a repetição pode terminar. Ou seja, o erro desejado deve ser menor ou igual a, por exemplo, 0.001.
7. Escreva uma função para um caixa de banco, que recebe um valor inteiro R e algumas variáveis (por referência) para armazenar o número de notas de 100, 50, 10, 5 e 1 reais, necessárias para pagar a quantia R . Faça de forma que o número de notas usado seja o menor possível.
8. Escreva uma função para calcular a soma a seguir. Crie uma função auxiliar para calcular o fatorial de um número: $n! = n \times n - 1 \times n - 2 \cdots \times 1$

$$S = 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \cdots + \frac{1}{8!}.$$

9. Escreva uma função que recebe um valor inteiro n e retorna 1 se este valor for um número primo (0 se não for). Um número primo é divisível apenas por ele e por 1.
10. Escreva uma função que calcula o valor (aproximado) de $\cos(x)$, dado pela aproximação abaixo:

$$\cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

Faça a soma com três termos, e escreva um programa que recebe valores de x e calcula seu cosseno, calculando **também** a função cosseno disponível em C e apresentando a diferença entre os resultados obtidos. Se você achar o erro grande demais, recalcule com mais termos e compare outra vez. Use novamente a função fatorial já criada.

11. Seus colegas da Biologia criaram uma bactéria que come lixo, consumindo cerca de metade de seu peso em lixo todos os dias. (Para facilitar, vamos supor que uma bactéria pesa cerca de 10^{-11} gramas). Seus colegas tem apenas cinco dessas bactérias, mas elas se dividem em duas todos os dias. A partir daí, escreva um programa que imprime uma tabela a partir do primeiro dia, quando existem apenas cinco bactérias, até dois meses depois (você terá de usar inteiros longos ou números de ponto flutuante para isso). A tabela deve listar o número do dia, quantas bactérias existem até o momento e quanto lixo elas comeram naquele dia. Depois de fazer isso, adapte seu programa para estas situações:

- (a) Suponha que as bactérias vivem apenas cinco dias. Neste caso, as bactérias que morrem devem ser descontadas do total.
- (b) Faça seu programa se adaptar às unidades de peso: de acordo com a quantidade de lixo consumida as unidades devem ser escritas em microgramas (10^{-6} g), miligramas (10^{-3} g), gramas, quilogramas ou toneladas, usando a maior unidade onde o peso seja maior do que um. Por exemplo, se for preciso imprimir 0.0037 gramas, você deve escolher 3.7 miligramas, e não 3700 microgramas.
12. Use seu conhecimento adquirido nos exercícios anteriores para imprimir as primeiras doze linhas do triângulo de Pascal como mostrado abaixo. Faça isso usando funções e não se importe com o tempo gasto.

```
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```