INSTITUTO FEDERAL
Catarinense
Campus Blumenau

Algoritmos

Bacharelado em Ciência da Computação
Prof. Paulo César Rodacki Gomes

Lista de exercícios -Alg-05.2
Exercícios sobre Funções - Parte 2

Estes exercícios devem ser entregues no Google Classroom. Para cada um dos exercícios, crie um arquivo fonte Python com o respectivo nome de acordo com a seguinte regra: SUASINICIAIS-Alg-05-Ex-num.py. Por exemplo, se o professor resolvesse o exercício número 3, o nome do arquivo seria PCRG-Alg-05-Ex-03.py.

Introdução

Funções permitem que o programador quebre um problema em partes de código que podem ser reutilizadas. Elas também ajudam o programador a se concentrar isoladamente em partes menores de um problema maior. Com isso, frequentemente escrever funções é uma parte importante do desenvolvimento de softwares maiores. Os exercícios desta lista tem o objetivo de lhe ajudar a se aprimorar nas seguintes habilidades:

- Definir uma função para usá-la posteriormente
- Passar um ou mais valores para uma função
- Fazer cálculos relativamente complexos usando funções
- Retornar um ou mais resultados re uma função
- Chamar (executar) uma função que você definiu anteriormente.

Questões:

1. **Hipotenusa**. Escreva uma função que recebe como parâmetros os comprimentos dos dois lados menores de um triângulo retângulo. A função deve retornar como resultado o comprimento da hipotenusa, calculado com o Teorema de Pitágoras. Inclua um programa principal (função main) que solicite ao usuário os comprimentos dos lados, utilize sua função para calcular a hipotenusa e imprima o resultado.

2. **Tarifa do táxi**. Em uma determinada cidade, as tarifas de táxi consistem em um valor inicial de R$ 4,00 mais R$ 0,25 a cada 140 metros percorridos. Escreva uma função que recebe como seu único parâmetro a distância percorrida em quilômetros, e retorna como seu único resultado o valor total da corrida. Escreva um programa principal que demonstre o funcionamento de sua função.

   > **Dica**: as tarifas de táxi mudam com tempo. Utilize constantes para representar o preço inicial e a parte variável da tarifa de forma que o programa possa ser facilmente atualizado caso os valores de tarifa sofram aumento.

3. **Calculadora de envio e-commerce**. Uma loja online fornece envio de seus itens pelo preço de R$ 10,95 para o primeiro item e R$ 2,95 para cada um dos demais itens. Escreva uma função que receba a quantidade de ítens de um pedido e retorne o valor total do envio de acordo com essas regras. Inclua um programa principal que leia do usuário o número de itens adquiridos e mostre o custo do envio.

4. **Mediana de três valores.** Escreva uma função que recebe três números como parâmetros e retorna o valor da mediana desses parâmetros como seu resultado. Inclua um programa principal que lê três valores do usuário e exibe a mediana destes valores.

Dica: a mediana é o valor do meio dos três valores quando eles são classificados em ordem crescente. Ela pode ser encontrada usando comandos condicionais ou então com um pouco de criatividade matemática (veja exercício 4 da lista 2).

5. **Números ordinais.** Palavras como *primeiro*, *segundo* e *terceiro* são chamadas de números ordinais. Neste exercício, você deve escrever uma função que recebe um inteiro como seu único parâmetro e retorna uma string contendo o número ordinal em português como seu único resultado. Sua função deve lidar com números inteiros entre 1 e 12 (inclusive). Ela deve retornar uma string vazia se um valor fora desse intervalo for fornecido como um parâmetro. Inclua um programa principal que demonstra sua função exibindo cada inteiro de 1 a 12 e seu respectivo número ordinal.

6. **Centralizando uma string**. Escreva uma função que recebe uma string como seu primeiro parâmetro e a largura de uma linha do terminal em caracteres como seu segundo parâmetro. Sua função deve retornar uma nova string que consiste na string original e no número correto de espaços iniciais para que a string original apareça centralizada dentro da largura fornecida quando for impressa. Não adicione nenhum caractere ao final da string. Inclui um programa principal que demonstre sua função.

7. **Triângulo válido?** Se você possui 3 canudos, possivelmente de tamanhos diferentes, pode ou não ser possível montar um triângulo juntando as pontas dos canudos. Por exemplo, se todos tiverem 15 cm de comprimento, facilmente você pode formar um triângulo equilátero. Porém, se um canudo tem 15 cm e os outros dois tem 5 cm cada, você não consegue formar o triângulo. Generalizando: se o comprimento de um canudo é maior ou igual que a soma dos comprimentos dos outros dois outros canudos, eles não podem formar um triângulo. Caso contrário, podem formar um triângulo. Escreva uma função que determina se 3 comprimentos podem ou não formar um triângulo. A função deve receber 3 parâmetros e retornar um valor lógico. Além disso, escreva um programa que leia 3 valores do usuário e demonstre o comportamento desta função.

8. **Letras maiúsculas**. Muitas pessoas não usam letras maiúsculas corretamente, especialmente ao digitar em smartphones. Neste exercício, você escreverá uma função que coloca em maiúscula os caracteres apropriados em uma string. O primeiro caractere da string deve ser convertido em letra maiúscula, assim como o primeiro caractere (que não seja espaço) após um “.”, “!” ou "?". Por exemplo, se a função for fornecida com a string “que horas eu tenho que estar lá? qual é o endereço?" então ele deve retornar a string “Que horas eu tenho que estar lá? Qual é o endereço?". Inclua um programa principal que leia uma string do usuário, corrige as letras maiúsculas usando sua função e exibe o resultado.

9. **A string representa um inteiro?** Neste exercício, você deve escrever uma função chamada isInteger que determina se os caracteres em uma string representam ou não um inteiro válido, retornando um valor lógico como resultado. Ao determinar se uma string representa um inteiro, você deve ignorar qualquer espaço em branco à esquerda ou à direita. Uma vez que este espaço em branco é ignorado, uma string representa um inteiro se seu comprimento for pelo menos 1 e contiver apenas dígitos, ou se seu primeiro caractere for + ou - e o primeiro caractere é seguido por um ou mais caracteres, todos os quais são dígitos. Escreva um programa principal que leia uma string do usuário e informe se ela representa ou não um número inteiro.

Dica: para concluir este exercício, você pode achar úteis os métodos lstrip, rstrip e/ou strip do tipo de dados string. A documentação para esses métodos está disponível online.

10. **Números primos**. Um número inteiro positivo é primo se e somente se ele for divisível apenas por 1 e por ele mesmo. Escreva uma função que recebe um valor inteiro positivo e retorna True se ele for primo ou False se ele não for. Escreva um programa principal que leia um número inteiro do usuário e exiba uma mensagem indicando se ele é ou não primo.

11. **Senha aleatória**. Escreva uma função que gere uma senha aleatória. A senha deve ter um comprimento aleatório de 7 a 10 caracteres. Cada caractere deve ser selecionado aleatoriamente das posições 33 a 126 na tabela ASCII. Sua função não receberá nenhum parâmetro. Ele retornará a senha gerada aleatoriamente como seu único resultado. Exiba a senha gerada aleatoriamente no programa principal do seu arquivo fonte.

> **Dica**: para concluir este exercício, você provavelmente vai achar útil a função chr do Python. A documentação para essa função está disponível online.

12. **Verificação de senha válida**. Neste exercício, você deve escrever uma função que determina se uma senha é válida ou não. Definiremos uma boa senha como aquela que tem pelo menos 8 caracteres e contém pelo menos uma letra maiúscula, pelo menos uma letra minúscula e pelo menos um número. Sua função deve retornar True se a senha passada a ela como seu único parâmetro for válida. Caso contrário, ele deve retornar False. Inclua um programa principal que lê a senha do usuário e informa se ela é ou não válida.

13. **Dias em um mês.** Escreva uma função que determina quantos dias há em um determinado mês. Sua função deve receber dois parâmetros: o mês como um número inteiro entre 1 e 12 e o ano como um número inteiro de quatro dígitos. Certifique-se de que sua função retorne o número correto de dias em fevereiro para os anos bissextos. Inclua um programa principal que lê um mês e ano do usuário e exibe o número de dias naquele mês. O exercício 12 da lista 3 pode ajudá-lo a resolver esse problema.

14. **Datas mágicas.** Uma “data mágica” é uma data na qual a multiplicação do dia pelo mês é igual aos dois últimos dígitos do ano. Por exemplo, 10 de junho de 1960 é uma data mágica porque 10 vezes 6 é igual a 60, que são os dois últimos dígitos do ano. Escreva uma função que determina se uma data é ou não mágica. A função deve receber 3 parâmetros inteiros (dia, mês e ano), e retornar um valor lógico. Escreva um programa main que utilize sua função para determinar e imprimir todas as datas mágicas do século XX. O exercício anterior pode ser útil para resolver este problema.

15. **Dígitos hexadecimais e decimais**. Escreva duas funções chamadas hex2int e int2hex, que devem fazer a conversão entre dígitos hexadecimais (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E e F) e números inteiros de base 10 . A função hex2int é responsável por converter uma string contendo **um único dígito** hexadecimal em um inteiro de base 10, enquanto a função int2hex é responsável por converter um inteiro entre 0 e 15 em um único dígito hexadecimal. Cada função pegará o valor a ser convertido como seu único parâmetro e retornará o valor convertido como o único resultado da função. Certifique-se de que a função hex2int funcione corretamente para letras maiúsculas e minúsculas. Observação: se você não sabe como converter números entre bases diferentes, veja o quadro explicativo ao final da lista de exercícios.

16. **Conversão arbitrária de base numérica.** Escreva um programa que permita ao usuário converter um número de uma base para base 10 e vice-versa. Seu programa deve suportar bases entre 2 (binário) e 16 (hexadecimal) para o número de entrada e o número de resultado. Divida seu programa em várias funções, incluindo uma função que converte de uma base arbitrária em uma base 10, uma função que converte de uma base 10 em uma base arbitrária. A primeira função deve receber como parâmetros uma string com o número a ser convertido para base 10, e o valor da base deste número (portanto, de 2 a 16) e deve retornar como resultado o número convertido na base 10. A segunda função deve receber como parâmetros um numero na base 10 e a base para qual queremos converter o número. Como resultado, a função deve retornar uma string com o número convertido. Além disso, faça um programa principal que lê as bases e o número de entrada do usuário. Você pode encontrar parte da solução deste problema no exercício anterior e nos exercícios 14 e 15 da lista 4.

Conversão de base numérica:

para converter um número na base b para decimal, dado um número natural b > 1 (base) e o conjunto de símbolos $\{\pm 0,1,2,\ldots,b-1\}$ (algarismos deste sistema numérico), a sequência de símbolos $(d_n d_{n-1}\ldots d_1 d_0, d_{-1}d_{-2}\ldots)_b$ na base b pode ser convertida para base 10 com a seguinte fórmula (acrescentando "-" no início se a sequencia de símbolos for negativa):

$$numero_{10} = d_n \cdot b^n + d_{n-1} \cdot b^{n-1} + \ldots + d_0 \cdot b^0 + d_{-1} \cdot b^{-1} + d_{-2} \cdot b^{-2} + \ldots$$

Exemplo 1: para converter o número binário $(1001,101)_2$ em decimal:

$$numero_{10} = (1001,101)_2$$

$$numero_{10} = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$numero_{10} = 8 + 0 + 0 + 1 + 0{,}5 + 0 + 0{,}125 = 9{,}625$$

Ou seja, o número $(1001,101)_2$ é igual a 9,625 no sistema decimal.

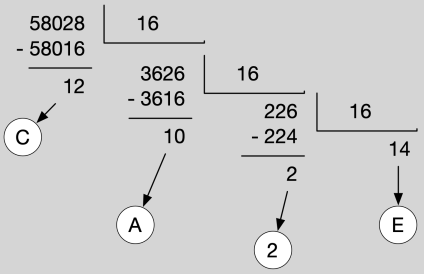
Exemplo 2: para converter o número $(E2AC)_{16}$ para base 10, temos:

$$numero_{10} = (E2AC)_{16}$$ Obs.: note que E corresponde a 14, A a 10 e C a 12.

$$numero_{10} = 14 \cdot 16^3 + 2 \cdot 16^2 + 10 \cdot 16^1 + 12 \cdot 16^0$$

$$numero_{10} = 57344 + 512 + 160 + 12 = 58028$$

Para converter da base 10 para base b fazemos o inverso, com divisões inteiras sucessivas e resto dessas divisões. Por exemplo, para converter $(58028)_{10}$ para base 16 (hexadecimal), fazemos o seguinte:



Montando o número com os caracteres encontrados na ordem inversa, temos E2AC, portanto o número 58028 na base 10 corresponde a E2AC na base 16.

Fonte:
https://www.ufrgs.br/reamat/CalculoNumerico/livro-oct/rdneadm-sistema_de_numeracao_e_mudanca_de_base.html