Computação I - Python

Laboratório 1

Para completar essa primeira aula, você deve agora definir e testar funções para resolver alguns problemas que fornecemos a seguir. Por enquanto, sabemos apenas como definir funções que realizam algumas contas simples e devolvem o resultado dessas contas. É esse tipo de problemas que você vai ter que resolver nessa lista. Dividimos a lista em 3 grupos de problemas:

- Problemas de aplicação direta de algum cálculo algébrico, como foi o caso do problema da média, do dobro e da soma. São problemas simples onde a identificação das informações de entrada e saída da função assim como qual o cálculo que deve ser feito é praticamente imediata.
- Problemas também puramente matemáticos, mas aplicados à geometria. Aqui também, a identificação das informações de entrada e saída da função é praticamente imediata e para a identificação do cálculo que deve ser feito basta lembrar das aulas de geometria.
- Aplicações variadas. Para você resolver os problemas desse grupo você precisa aplicar todas as etapas para o desenvolvimento de uma solução que vimos ao longo da nossa aula, mesmo que com algoritmos ainda muito simples.

Lembre-se de sempre:

- estruturar a fase de estudo do problema ANTES de começar a fazer o algoritmo ou programa; Identificar entradas; identificar a saída esperada; RESOLVER o problema; identificar como validar a solução; codificar; testar.
- documentar suas funções, descrevendo em cada uma, suas entradas e o que ela faz;
- escolher nomes elucidativos para suas funções e parâmetros;
- pensar em valores de teste relevantes para testar sua função. Ela tem alguma resposta esperada para valores negativos? Valores fracionários? Que tal testar também com valores no extremo do conjunto de dados de interesse da função (maiores valores esperados, menores valores esperados)?

Cálculos Geométricos

1. Defina uma função que calcule a área de um retângulo dados seus dois lados. Teste pelo menos para os seguintes pares de entrada:
   - 5 e 7; resposta esperada é 35
   - 15 e 2; resposta esperada é 30
   - 500 e 700; resposta esperada é 350000

- 5 e 0; resposta esperada é 0

Solution:

```
def areaRetangulo(lado1, lado2):
    ''' Retorna a área de um retângulo dados seus dois lados 'lado1' e 'lado2'. '''
    ''' Explicação: Sendo área base * altura, considerando
    base como sendo 'lado1' e altura como sendo 'lado2',
    temos 'lado1' * 'lado2' '''
    return lado1 * lado2
```

2. Defina uma função que calcule a área da superfície de um cubo que tem c por aresta.

Solution:

```
def areaCubo(aresta):
    ''' Retorna a área da superfície de um cubo que tem por
    aresta 'aresta'. '''
    ''' Explicação: Um cubo tem 6 faces, e cada face é um
    quadrado de lado 'aresta'. Como a área de cada quadrado é
    base * altura, isto é, 'aresta' * 'aresta', a área da
    superfície do cubo é 6 * 'aresta' * 'aresta'. '''
    return 6 * aresta**2
```

3. Defina uma função que calcule a área da coroa circular (anel) formada por dois círculos de raios $r1$ e $r2$ ($r1 > r2$ e $Pi = 3.14$). Teste pelo menos para os seguintes pares de entrada:

- 2 e 1; resposta esperada é 9.42
- 15 e 5; resposta esperada é 628
- 100 e 0; resposta esperada é 31400

Solution:

```
def areaCoroaCircular(r1, r2):
    ''' Retorna a área da coroa circular formadas pelos
    círculos de raio 'r1' e 'r2', sendo 'r1' > 'r2'. '''
    ''' Explicaçao: O melhor exemplo de coroa circular é uma
    moeda de 1 real. A parte dourada dela é o que chamamos
    de coroa circular. Para calculá-la, tiramos da área do
    círculo maior (total) a área do círculo menor (parte
    prateada), sobrando apenas a parte dourada. Ou seja,
    'PI' * 'r1' ** 2 - 'PI' * 'r2' ** 2. '''
    return 3.14 * (r1**2 - r2**2)
```

Cálculos Algébricos

4. Termine o desenvolvimento da função que calcula a média de dois números. Teste pelo menos para os seguintes pares de entrada:

   - - 5 e 7;
   - 2 e -2;
   - 5 e 5;
   - 3 e 4;
   - 3.0 e 4.0;

Solution:

```
def mediaAritmetica(x, y):
    ''' Retorna a média aritmética entre dois números 'x' e
    'y'. '''
    ''' Explicação: Média aritmética entre dois números nada
    mais é do que o "meio termo" entre eles. Para calculá-la,
    basta somá-los e dividir pela quantidade, isto é, 2.
    Ou seja, ('x' + 'y')/2 '''
    return (x + y)/2
```

5. Defina uma função que calcule a ordenada de uma função de segundo grau dados os parâmetros a, b, c e a abscissa.
   Observação: Não é procurar as raízes de uma equação de segundo grau! É só calcular o valor do resultado de uma função representada por um polinômio do segundo grau para um dado x (a abscissa).

Solution:

```
def ordenada(a, b, c, x):
    ''' Retorna a ordenada de uma função de segundo grau,
    dados seus parâmetros 'a', 'b', 'c' e uma abscissa 'x'.'''
    ''' Explicação: A função do segundo grau é da forma
    y = a*x**2 + b*x + c. Logo, temos nossa fórmula prontinha! '''
    return a * x**2 + b * x + c
```

6. Defina uma função que calcule a média ponderada de dois números com os respectivos pesos.
   Observação: Não estamos especificando no enunciado a ordem em que as informações de entrada vão ser fornecidas. Você é livre para decidir, mas não deixe de explicar qual é essa ordem na descrição da função.

Solution:

```
def mediaPonderada(x, p1, y, p2):
    ''' Retorna a média ponderada de dois números 'x' e 'y'
    com seus respectivos pesos 'p1' e 'p2'. '''
    ''' Explicação: Média ponderada entre dois números se
    calcula multiplicando cada valor pelo seu respectivo peso,
    somando as quantias para cada valor e dividindo o
    resultado pela soma dos pesos. Ou seja, ('x'*'p1' +
    'y'*'p2')/('p1'+'p2') '''
    return (x * p1 + y * p2)/(p1 + p2)
```

7. Chamamos de *erro* de uma aproximação, a diferença entre essa aproximação e o valor que tentamos aproximar. No caso de uma PG infinita com razão $0 \leq q < 1$, podemos somar alguns termos da PG e assumir que já que esses termos estão tendendo a zero, podemos parar em algum momento e considerar que o resultado obtido é uma aproximação da soma de todos os elementos da PG. Quanto mais termos somarmos, melhor a aproximação, ou seja, menor o erro. Defina uma função que calcule o erro entre o valor da soma de uma PG infinita a partir de 1.0 e a soma dos n primeiros termos dessa PG. **A soma dos termos de uma PG é** $1/(1-q)$**, onde** q **é a razão e** $0 \leq q < 1$. Deixamos para você a tarefa de lembrar a fórmula da soma do n primeiros termos.

Solution:

```
def erroPG(q, n):
    ''' Retorna o erro da aproximação da soma da PG infinita
    de razão 'q' (0 <= q < 1) e a soma dos 'n' primeiros
    termos dessa mesma PG. '''
    ''' Explicação: A soma da PG infinita 1/(1 - q) nada mais
    é do que a fórmula da soma da PG comum, porém com 'n'
    tendendo ao infinito, e, por consequência, q**'n' tende a
    0, visto que 0 <= q < 1. Assim, basta calcular a diferença
    das fórmulas. '''
    return 1/(1 - q) - (q**n - 1)/(q - 1)
```

Cálculos Aplicados

8. Defina uma função que, dado o valor da conta de um restaurante, calcule a gorjeta do garçom, considerando que a gorjeta deve ser 15% do valor da conta.

Solution:

```
def gorjeta(conta):
    ''' Retorna o valor da gorjeta do garçom dado o valor da 'conta' '''
    ''' Explicação: Dado o valor da 'conta', basta calcular 10% dele. '''
    return 0.15 * conta
```

9. Defina uma nova função que, dado o valor da conta de um restaurante e a porcentagem exigida pela legislação para a gorjeta, calcule o valor dessa gorjeta.

Lembre-se de incluir na descrição da função como essa porcentagem deve ser informada pelo usúario. Por exemplo: 10% é 10 ou 0.1?

Solution:

```
def gorjetaTaxa(conta, taxa):
    ''' Retorna o valor da gorjeta do garçom dado o valor da
    'conta' e a 'taxa', representando 'taxa'%, isto é, 10
    representa 10%. '''
    ''' Explicação: Dado o valor da 'conta', e a 'taxa' em sua
    forma de 0 a 100, basta calcular 'conta' * 'taxa'/100,
    devido à representação escolhida para a taxa. '''
    return conta * taxa/100
```

10. Defina uma função que calcule o saldo final de uma conta, dado o saldo inicial, o número de meses e a taxa de juros mensal (juros simples).
Obs. Aqui temos o mesmo problema que tivemos na questão da gorgeta. O usuário precisa saber como deve fornecer a informação.

Saldo Final = Saldo Inicial (1 + juros.meses)

Solution:

```
def jurosSimples(saldoInicial, meses, taxa):
    ''' Retorna o valor do saldo final de uma conta, aplicado
    uma 'taxa' por 'meses' à um 'saldoInicial'. '''
    ''' Explicação: Basta replicar a fórmula saldoFinal =
    saldoInicial * (1 + juros * meses). '''
    return saldoInicial * (1 + taxa*meses)
```

11. Defina uma função que calcule a distância que a correnteza arrasta um barco que atravessa um rio. São conhecidas: a velocidade da correnteza, a largura do rio e a velocidade do barco perpendicular à correnteza. Isso! Você vai ter que pensar na solução do problema na Física antes de pensar em começar a programar...

Solution:

```
def deslocamento(correnteza, largura, barco):
    ''' Retorna o valor do deslocamento do barco dada a
    velocidade da 'correnteza', a 'largura' do rio e a
    velocidade do 'barco'. '''
    ''' Explicação: Para descobrir por quanto tempo a
    correnteza deslocou o barco, temos que descobrir quanto
    tempo levou para ele se deslocar até o outro lado do rio.
    Para isso, fazemos 'largura'/'barco'. Agora que sabemos o
    tempo, basta multiplicar esse tempo pela velocidade do
```

```
    barco (perpendicular à correnteza) e descobrimos seu
    deslocamento! '''
    return (largura/barco) * correnteza
```

o tempo para chegar à outra margem depende apenas da velocidade do barco e da largura (largura/vbarco). Dado o tempo, o quanto o barco vai derivar, depende apenas da correnteza (tempo * corrente).