Aula03 - Funções

Exercícios

- 1. O programa bmi.py serve para calcular o índice de massa corporal, mas está incompleto. O programa inclui três funções. Analise o seu funcionamento.
 - a. Complete a definição da função bodyMassIndex para calcular o índice pela razão $bmi = \frac{weight}{height^2}$. Complete os argumentos na invocação da função, dentro da função principal. Teste o programa.
 - b. Complete a função bmiCategory para <u>devolver</u> uma string com a categoria correspondente ao índice de massa corporal dado. Acrescente uma chamada a esta função na função principal, para obter o nome da categoria. Volte a testar.
- 2. Escreva uma função para calcular o polinómio $p(x) = x^2 + 2x + 3$ e use-a num programa para calcular e mostrar os valores de p(0), p(1), p(2) e p(p(1)). Confira os resultados.
- 3. Defina uma função que devolva o maior dos seus dois argumentos. Por exemplo, max2 (4, 3) deve devolver 4 enquanto max2 (-3, -2) deve devolver -2. Não pode usar a função pré-definida max. Use uma instrução de seleção if ou uma expressão condicional. Teste a função com vários conjuntos de argumentos.
- 4. No mesmo programa, crie uma função max3 que devolva o maior dos seus 3 argumentos. Não pode usar a função max, nem instruções ou expressões condicionais. Recorra apenas à função max2 que definiu atrás. Teste a nova função.
- 5. Escreva uma função, tax (r), que implemente a seguinte função de ramos:

$$tax(r) = \begin{vmatrix} 0.1r & \text{se} & r \le 1000 \\ 0.2r - 100 & \text{se} & 1000 < r \le 2000 \\ 0.3r - 300 & \text{se} & 2000 < r \end{vmatrix}$$

Use uma instrução if-elif-else e evite condições redundantes. Teste a função para diversos valores de r e confirme os resultados. Que valores deve testar?

- 6. Escreva uma função intersects (a1,b1,a2,b2) que devolva True se os intervalos [a1,b1[e [a2,b2[se intersectarem e devolva False, caso contrário. Admita que $a_1 \le b_1 \land a_2 \le b_2$. Sugestão: é mais simples definir quando os intervalos não se intersectam. Experimente responder no CodeCheck.
- 7. Analise e execute o programa dates.py. Faça as correções indicadas abaixo.
 - a. A função isLeapYear devia indicar quando um ano é bissexto, mas está errada. Corrija-a. Um ano é bissexto se for múltiplo de 4, com exceção dos fins de século (múltiplos de 100), que só são bissextos se forem múltiplos de 400. Por exemplo: 1980, 1984, 2004 foram bissextos; 1800 e 1900 foram anos comuns, mas 2000 foi bissexto.
 - b. A função monthDays, para determinar o número de dias de um mês, também está errada. Quando o mês é fevereiro, invoque a função anterior para determinar se o ano é bissexto e devolva 29 dias nesse caso.

- c. Corrija a função nextDay para devolver o dia seguinte corretamente.
- 8. Complete a função hms2sec de forma a devolver o número de segundos correspondente a h horas, m minutos e s segundos. Faça no <u>CodeCheck</u>.
- 9. Complete a função sec2hms de forma a converter um número de segundos sec em h horas, m minutos e s segundos. Faça no <u>CodeCheck</u>.
- 10. ** Escreva uma função countdown (N) que imprima uma contagem decrescente a partir de um número positivo N. Note que pode imprimir N e depois fazer countdown (N-1). Teste a função com diversos valores de N.
- 11. ** O algoritmo de Euclides para determinar o máximo divisor comum de dois números naturais baseia-se na igualdade seguinte:

$$mdc(a,b) = \begin{bmatrix} b & se & r = 0 \\ mdc(b,r) & se & r > 0 \end{bmatrix}$$

onde r é o resto da divisão de a por b. Escreva uma função para calcular o m.d.c. e teste-a com diversos pares de valores.