

# Lista de Exercícios

Fundamentos de Algoritmos – INF05008

Professor Responsável: Ana L. C. Bazzan

## 2. Expressões Aritméticas

1. [Fácil] Crie programas utilizando o *DrRacket* para resolver as seguintes expressões:

- (a)  $\frac{x+2}{2} - 2x$
- (b)  $144 - x^2 + x$
- (c)  $\sqrt{x} - 2x + 12$
- (d)  $x^3 - \frac{x}{4} + x^{-1}$
- (e)  $2 \cdot \sqrt{x+5} - 26$
- (f)  $x \cdot (x+3) - 4$

Através dos programas criados, determine o valor de cada função para  $x \in \{2, 3, 5, 10\}$ . Experimente verificar a execução de cada expressão utilizando o *Stepper*.

- 2. [Fácil] Faça um programa que, dados os valores para os catetos de um triângulo retângulo, calcule o valor da hipotenusa. A hipotenusa é calculada pelo *Teorema de Pitágoras* que é dado pela expressão  $a^2 = b^2 + c^2$ , onde  $b$  e  $c$  são os catetos e  $a$  a hipotenusa.
- 3. [Fácil] Modifique o programa anterior para que calcule o valor de um dos catetos. Para isso, a função deverá receber os valores do cateto conhecido e da hipotenusa.
- 4. [Fácil] Escreva um programa utilizando o *DrRacket* para a conversão de temperatura ( $^{\circ}F$  para  $^{\circ}C$ ). A expressão que calcula o valor correspondente em  $^{\circ}C$  é  $C = \frac{F-32}{1,8}$ .
- 5. [Fácil] Similar ao exercício anterior, escreva um programa que, dado um valor de temperatura em  $^{\circ}C$ , calcule seu valor correspondente em  $^{\circ}F$ . Determine o ponto de ebulição da água em  $^{\circ}F$  (considere o ponto de ebulição da água como  $100^{\circ}C$ ).
- 6. [Médio] Escreva um programa em *DrRacket* para o cálculo da área de um trapézio. Para isso, o programa deverá utilizar as variáveis base maior ( $B$ ), base menor ( $b$ ) e altura ( $h$ ). A área de um trapézio pode ser calculada pela fórmula  $A_t = \frac{(B+b) \cdot h}{2}$ .
- 7. [Médio] Escreva uma função no *DrRacket* que faça uso de uma variável única, composta pelo raio de uma circunferência. A função deverá retornar a soma do comprimento circunferência com sua área ( $C = 2 \cdot r \cdot \pi$  e  $A = \pi \cdot r^2$ ).
- 8. [Médio] Crie um programa para o cálculo do volume de um cilindro. Execute a função criada com distintos valores de raio  $r \in \{1.5, 2, 5\}$  e altura  $h \in \{12, 20, 32\}$ .

9. [Médio] O salário líquido de uma empresa é calculado descontando do salário bruto uma determinada porcentagem referente ao imposto cobrado dos trabalhadores. O salário bruto, por sua vez, consiste na multiplicação da quantidade de horas trabalhadas pelo valor pago pela hora. Crie um programa em *DrRacket* que compute o valor do salário líquido. Teste o programa criado para uma quantidade total de 110 horas a um valor de \$ 15.50 a hora trabalhada e 11% de imposto.
10. [Difícil] Uma loja de artigos variados possui uma política de cálculo para o valor a ser cobrado dos seus clientes na compra de suas mercadorias. O valor a ser pago pelo cliente é composto pelo valor unitário da mercadoria multiplicado pela quantidade a ser comprada. Usualmente, a loja oferece um desconto que é subtraído do valor a ser pago no ato da venda. O valor unitário de cada mercadoria é obtido somando o custo do bem com um valor de lucro, determinado por uma porcentagem. O desconto também é determinado por uma porcentagem que é aplicada sobre o valor total no ato da venda. Crie um programa para computar o valor final a ser pago pelo cliente em uma compra. Após criado o programa, determine qual a porcentagem máxima (considerar somente múltiplos de 5) de desconto para que o vendedor não tenha prejuízo em uma venda de 12 unidades de um artigo com custo \$ 8.40 com 33% de lucro.
11. [Difícil] Crie funções para conversão de meses em anos e vice-versa (*mes2ano* e *ano2mes*). Crie também funções para conversão de taxas de juros mensais para taxas anuais e vice-versa (*mensal2anual* e *anual2mensal*). Utilize estas funções para resolver os problemas a seguir.
  - (a) Um indivíduo investiu \$ 35.000,00 em uma aplicação durante 1 semestre à taxa de juros simples de 18,68% a.a. Em quanto o capital foi aumentado ao final do período?
  - (b) Em um período total de 2,5 anos, um capital de \$ 12.200,00 foi aplicado à uma taxa de juros de 7,2% a.m. Determine o montante ao final do período.

Obs: Para cada problema, teste a conversão tanto da taxa quanto do tempo. A fórmula para cálculo com juros simples é dada por  $M = C \cdot (1 + (i \cdot n))$ .
12. [Difícil] Crie um programa que determine a velocidade média de um veículo. Para isso, o usuário deverá fornecer os valores para as variáveis posição inicial e final (em quilômetros) e tempo inicial e final (em horas). A velocidade média pode ser calculada pela expressão  $V = \frac{\Delta s}{\Delta t}$ .
13. [Médio] Modifique o programa anterior para determinar a posição final de um veículo com base nas demais variáveis.

### 3. Projeto de Algoritmos e 4. Expressões Condicionais;

14. [Fácil] Escreva uma função no *DrRacket* que consome uma variável numérica. Caso o número informado seja maior que 100, ele deve ser dividido por 5, caso contrário deve ser multiplicado por 3.
15. [Fácil] Crie um programa que, dado um número inteiro qualquer, verifique se o mesmo é par ou ímpar. No caso de ser par, o mesmo deve ser dividido por 2. Caso o mesmo seja ímpar, deve ser somado a 1 e posteriormente dividido por 2. Considere utilizar a função (*remainder m n*) para verificar se um número é par ou ímpar.

16. [Fácil] Um proprietário de automóvel gostaria de saber se é mais viável abastecer seu veículo com álcool ou gasolina. Ele sabe que para ser vantajoso abastecer no álcool, o preço do litro do álcool tem que custar até 70% (inclusive) do preço do litro da gasolina. Faça um programa que consuma o preço do litro do álcool e da gasolina e informe qual é combustível mais viável financeiramente.
17. [Fácil] Desenvolva uma calculadora em *DrRacket* para realizar as quatro operações básicas da matemática. Este programa consome uma string e dois números. A *string* representa o tipo da operação, e tem com possíveis valores: "adicao", "subtracao", "multiplicacao" e "divisao". De acordo com a string passada, o programa deve realizar a devida operação matemática entre os dois números. Na operação de divisão, o programa deve validar para que não ocorra divisão por zero.
18. [Fácil] Escreva uma função que determine a fase da vida de um indivíduo. A função deverá consumir uma variável numérica correspondente à idade e informar a faixa na qual o indivíduo se encontra. Até 12 anos a pessoa se encontra no estágio infantil, após isso a pessoa está na juventude até os 28 anos, vida adulta até os 65 anos e terceira idade com mais de 65 anos de idade.
19. [Médio] Escreva um programa para auxílio no fechamento das médias de uma turma de alunos. Para isso, uma função deverá consumir as quatro notas do aluno, calculando sua média aritmética simples. O programa ainda deverá determinar se a situação final do aluno (reprovado se a nota for menor que 5, em exame para nota entre 5 e 7 e aprovado para nota maior ou igual a 7). Para informar a situação do aluno ao usuário utilize o comando (`write 'mensagem'`).
20. [Médio] Modifique o programa anterior para uma média ponderada. A maior nota entre as quatro notas deverá possuir um peso 4/10, enquanto as demais possuirão peso 2/10.
21. [Médio] Crie um programa para cálculo do IMC (Índice de Massa Corporal). Este cálculo é determinado pela expressão  $IMC = \frac{peso}{altura^2}$ . A situação do indivíduo deve ser informada ao usuário de acordo com a tabela abaixo.

Resultado	Situação
Abaixo de 18	Abaixo do peso
Entre 18 e 24.99	Peso normal
Entre 25 e 29.99	Acima do peso
Acima de 30	Obesidade

22. [Médio] Crie um programa que calcule o valor de multa para um sensor de velocidade. O valor base de uma multa é estipulado em \$ 120,00. Caso o infrator ultrapassou a velocidade máxima em até 15km/h, a multa é calculada como 88% do valor base. Se a infração foi de 15km/h a 25km/h superior à velocidade máxima, a multa é igual a 116% do valor base. Para um excedente entre 25km/h e 40km/h, a multa é igual a 180% do valor base. Para excedentes superiores a 40km/h, o valor final é igual a 250% do valor base. Dada a velocidade máxima da via e a velocidade inferida pelo sensor, informe o valor da multa (para velocidades dentro do permitido, o valor da multa é \$ 0,00).
23. [Médio] Em uma competição de salto em altura, a pontuação do atleta é determinada por meio de uma expressão baseada na altura computada e no peso corporal ( $Pont = altura \cdot peso$ ), onde a altura varia entre 1 e 2 metros e o peso entre 50 e 90kg. Escreva

um programa que faça a leitura da altura computada e do peso de dois atletas (A e B) e determine qual deles é o vencedor.

24. [Médio] Uma companhia de crediário mantém um nível de confiança sobre cada um dos seus clientes, a fim de auxiliar na concessão ou negação de crédito. Este nível varia de acordo com o tempo em que o indivíduo é cliente e sua renda mensal. O nível é determinado de acordo com a tabela a seguir.

Tempo	Renda mensal	Nível
Até 1 ano	Menor que \$ 1200,00	1
	Maior ou igual a \$ 1200,00 e menor que \$ 2300,00	2
	Maior ou igual a \$ 2300,00	3
Mais que 1 ano	Menor que \$ 1200,00	2
	Maior ou igual a \$ 1200,00 e menor que \$ 2300,00	3
	Maior ou igual a \$ 2300,00	3

Empréstimos para clientes com nível 1 não são concedidos. Clientes com nível dois são analisados por especialistas. Todos os empréstimos com nível 3 são concedidos. Escreva um programa que, dado o tempo em que o indivíduo é cliente e sua renda, determine a situação da concessão de crédito (*negado, em análise ou aprovado*).

25. [Médio] Crie um programa que calcule o valor do seguro de um veículo. Caso o ano do mesmo seja anterior a 2000, o valor base do seguro é \$ 2000,00. Para veículos mais novos, o valor base cai para \$ 1200,00. Caso o proprietário tenha menos de 25 anos de idade, é acrescido \$ 800,00 no valor do seguro. Caso o veículo seja utilizado para trabalho, é acrescido um valor de \$ 650,00. O programa deverá fazer uso das variáveis necessárias e determinar o valor final do seguro.
26. [Médio] Na disciplina de fundamentos de algoritmos, os alunos são avaliados por suas notas obtidas nas três atividades da disciplina: prova 1 ( $p_1$ ), prova 2 ( $p_2$ ) e listas de exercícios ( $E$ ). A média final ( $M$ ) de cada aluno é calculada através da média aritmética ponderada de suas notas obtidas nessas três atividades, sendo que as atividades valem  $p_1 = 40\%$ ,  $p_2 = 50\%$  e  $E = 10\%$  da média final. Para que o conceito de um aluno seja estimado, o professor precisa ainda avaliar se o aluno possui a frequência ( $F$ ) mínima de participação nas aulas. A conversão da média final e frequência para o conceito é realizada da seguinte forma:

FF:  $F < 75\%$ ;

D:  $0 \leq M < 6.0$ ;

C:  $6.0 \leq M < 7.5$ ;

B:  $7.5 \leq M < 9.0$ ;

A:  $9.0 \leq M$ .

Para auxiliar o professor nesta tarefa, faça um programa em *DrRacket* que consuma as notas de  $p_1$ ,  $p_2$ ,  $E$  e  $F$  e calcule o conceito do aluno. Considere o intervalo  $[0, 10]$  para cada uma das notas e  $[0, 100]$  para a frequência.

27. [Médio] Agora que o professor possui um programa para calcular o conceito dos alunos, ele precisa verificar se os alunos foram aprovados ou reprovados na disciplina. Para que um aluno seja aprovado ele precisa ter conceito A, B ou C, caso contrário o aluno é reprovado. O programa deve consumir a saída obtida pelo programa do exercício anterior

e informar se o aluno está aprovado ou reprovado. No caso de ter sido reprovado, o sistema deve ainda informar o motivo da reprovação (nota ou frequência).

28. [Médio] Uma seguradora de veículos precisa de um programa de computador para facilitar o cálculo do valor da anualidade dos seguros. O valor da anualidade de um seguro é calculado com base no valor do veículo, ano de modelo e idade do condutor. O valor da anuidade do seguro de qualquer veículo se inicia com um valor base de \$ 500,00. Este valor é incrementado de acordo com as características do veículo e condutor. Inicialmente, a seguradora incrementa o valor base em função das características do veículo. Para veículos com modelo inferior a 1995, adiciona-se 20% do valor do veículo ao valor base. Para veículos entre 1995 e 2008, o valor adicionado é de 10% do valor do veículo. Para veículos acima de 2008, o valor a ser incrementado no valor base é de 4% do valor do veículo. A partir deste novo valor obtido a seguradora insere uma taxa de risco, variável em função do perfil do condutor. Como as pesquisas apontam que condutores mais jovens têm maior probabilidade de se envolverem em acidentes, este fator torna o valor do seguro mais atrativo para condutores mais velhos e menos atrativo para os condutores mais jovens. A partir do valor base já modificado, a seguradora incrementa este valor em 20% para os casos onde o condutor possui menos de 25 anos. Para os condutores com idade entre 25 e 50 anos, o valor do seguro não é incrementado. Já os condutores acima de 50 anos recebem um desconto de 10% sobre o valor total do seguro. Com base nesta descrição, desenvolva um programa em *DrRacket* que consuma o valor de um veículo, ano de modelo e idade do condutor e compute o valor da anualidade do seguro.
29. [Difícil] Escreva um programa no *DrRacket* que receba os valores de medida de três segmentos de reta. O programa deverá determinar se estas retas podem formar um triângulo. Para formar um triângulo, nenhuma de suas arestas pode ser maior que a soma das outras duas.
30. [Difícil] Uma empresa cobra de seus funcionários uma taxa mensal referente ao traslado de suas residências até o local de trabalho. A taxa é descontada do salário conforme a distância entre a empresa e a residência do funcionário e proporcional ao valor de seu salário. A tabela a seguir determina o valor a ser descontado na folha de pagamento.

Valor do salário	Distância	Valor do desconto
Até \$ 2000,00	Menor que 5km	\$ 80,00
	Maior ou igual a 5km e menor que 10km	\$ 120,00
	Maior ou igual a 10km	\$ 200,00
Mais que \$ 2000,00	Menor que 5km	\$ 110,00
	Maior ou igual a 5km e menor que 10km	\$ 160,00
	Maior ou igual a 10km	\$ 240,00

Com base nessas informações, crie um programa que compute o salário final do funcionário com base no valor do salário e na distância de sua residência em quilômetros.

## 5. Informações Simbólicas

31. [Fácil] Apresente o contrato da função `symbol=?`. Apresente também dois exemplos de entradas que retornam verdadeiro e dois exemplos de entradas que retornam falso.

32. [Fácil] Apresente o contrato da função `symbol?` (Note que esta é diferente de `symbol=?`). Apresente dois exemplos de entradas que retornam verdadeiro e dois exemplos de entradas que retornam falso.
33. [Médio] Para todos os casos a seguir, explique o que acontece quando as funções são executadas no `DrRacket`. No caso de erro, apresente uma solução para o que o comando apresente uma saída válida.
- (a) `(symbol=? 'Hello "Hello")`
  - (b) `(symbol? 'Hello 'Hello)`
  - (c) `(symbol=? '50 50)`
  - (d) `(symbol=? 'cinquenta "50")`
  - (e)
34. [Médio] Desenvolva um programa para retornar a classe social que uma pessoa pertence. Esse programa recebe como entrada uma renda e retorna um dos seguintes símbolos: `'baixa`, `'média` e `'alta`. É considerado classe baixa aquele que possui renda inferior a \$ 1280,00; classe média entre \$ 1280,00 e \$ 4800,00; já a classe alta é quem ganha acima de \$ 4800,00.
35. [Fácil] Faça um programa que consuma uma variável de um tipo qualquer (número, string, booleano ou estrutura) e imprima na tela o tipo da variável.
36. [Médio] Desenvolva a função `(pinta-forma tipo dimensão cor modo)` para pintar objetos, de acordo com as entradas consumidas. O atributo `tipo` é um símbolo que representa a forma do objeto a ser pintado (`'triângulo-equilátero`, `'círculo` ou `'quadrado`). A `dimensão`, por sua vez, representa a dimensão física do objeto, ou seja: lado para o triângulo equilátero e quadrado; e raio para o círculo. A `cor` representa a coloração na qual o objeto será pintado. O atributo `modo` define se a forma de preenchimento do objeto será sólida (`'solid`) ou apenas contorno (`'outline`). Utilize o *teachpack image* - através do comando: `(require 2htdp/image)` - para pintar objetos em Racket. As expressões a seguir, pintam os respectivos objetos.

- **triângulo equilátero:** `(triangle dimensão modo cor);`
- **círculo:** `(circle dimensão modo cor);`
- **quadrado:** `(square dimensão modo cor).`

Após desenvolvido, teste o programa com as seguintes entradas:

- `(pinta-forma 'triângulo-equilátero 20 "magenta"'outline);`
- `(pinta-forma 'triângulo-equilátero 15 "black"'solid);`
- `(pinta-forma 'triângulo-equilátero 25 "red"'outline);`
- `(pinta-forma 'círculo 14 "black"'outline);`
- `(pinta-forma 'círculo 23 "red"'solid);`
- `(pinta-forma 'círculo 20 "magenta"'outline);`
- `(pinta-forma 'quadrado 21 "magenta"'solid);`
- `(pinta-forma 'quadrado 18 "black"'outline);`
- `(pinta-forma 'quadrado 14 "red"'outline);`

37. [Difícil] Considere um motorista que precisa entregar mercadorias em três cidades distintas. A ordem das cidades é determinada antecipadamente. Antes de iniciar sua viagem, o motorista precisa se certificar de que possui combustível suficiente para realizar o percurso. Escreva um programa que, dadas as distâncias a serem percorridas para chegar a cada cidade, o consumo do veículo (em quilômetros por litro de combustível) e a quantidade de combustível disponível, informe quais cidades podem ser visitadas. A saída deverá ser informada ao usuário mediante os símbolos 'Cidade1, 'Cidades1e2 e 'Todas-Cidades.
38. [Fácil] Escreva uma função que determine o vencedor de um jogo de par ou ímpar. Para isso, o programa deverá receber os nomes dos dois jogadores, o valor informado por cada jogador e a escolha ('par ou 'ímpar) do primeiro jogador. O programa deverá retornar o nome do jogador vencedor.
39. [Médio] Deseja-se construir um programa para auxílio da determinação do resultado de uma determinada eleição. A eleição possui três candidatos ao cargo pretendido. Escreva uma função que receba os nomes dos candidatos e a quantidade de votos de cada um deles. A função deverá informar o nome do candidato vencedor, se houver. No caso de nenhum candidato possuir mais de 50% dos votos, deverá ser informado segundo turno e o nome dos dois respectivos candidatos.
40. [Médio] Escreva uma função que receba os nomes de duas pessoas. Além disso, a função deve consumir o ano, mês e dia de nascimento de ambos. A função deverá retornar o nome da pessoa mais velha.
41. [Médio] O plano de cargos e salários de uma determinada empresa classifica seus colaboradores de acordo com níveis (A, B, C e D). Funcionários com tempo de serviço menor ou igual a 2 anos ou com salário inferior a R\$ 2000,00 são classificados como nível D. Funcionários com menos de 2 anos de serviço e salário superior a R\$ 2000,00 possuem nível C. Além destes, funcionários com menos de 4 anos de serviço e salário inferior a R\$ 3500,00 também estão enquadrados no nível C. Para funcionários com menos de 4 anos de serviço e salário maior que R\$ 3500,00 é atribuído o nível B. O nível B também é atribuído a funcionários com mais de 4 anos de serviço e salário inferior a R\$ 5000,00. Demais funcionários possuem nível A. Com base no exposto, escreva uma função que receba o tempo de serviço e o salário de um funcionário e determine o nível no qual o mesmo se encontra ('A, 'B, 'C ou 'D).

## 6. Estruturas

42. [Fácil] O que é uma estrutura em DrRacket? Apresente um exemplo prático.
43. [Fácil] O que faz a função `struct`?
44. [Fácil] Considere a estrutura (`define-struct aluno (nota1 nota2 nota3 nota4)`) e faça o que é exigido a seguir.
- (a) Crie uma estrutura `aluno` com as seguintes notas 8, 9, 7 e 6.
  - (b) Para o aluno criado, apresente o código seletor que extrai a `nota2`.
  - (c) Desenvolva um programa que consuma um `aluno` e retorne a média aritmética simples de suas notas. Teste o programa no aluno criado anteriormente.

45. [Médio] Apresente as definições necessárias para representar a estrutura de um aluno de pós graduação. O aluno possui quatro informações principais: número de matrícula, nome do curso que está cursando, instituição de ensino, e curso de graduação cursado. Uma instituição de ensino é composta pelo seu nome e ano de fundação. Já o curso de graduação, por sua vez, possui um nome do curso e instituição de ensino.
46. [Médio] Considere as estruturas do exercício anterior e apresente as expressões necessárias para representar os alunos de pós graduação: Marcelo, Fernando e Gabriel. O Marcelo (matrícula 00237755) cursa Doutorado em Computação na UFRGS (fundada em 1934), e é formado em Ciência da Computação na UFRGS. O Fernando (matrícula 00236578), cursa Mestrado em Computação na UFRGS, e é formado em Sistemas de Informação pela UFSM (fundada em 1960). Já o Gabriel (matrícula 512010987), cursa Mestrado em Informática na UFSM, e possui graduação Engenharia de Software pela UDESC (fundada em 1965). Considere a reutilização das estruturas geradas as instituições de ensino.
47. [Médio] Desenvolva uma função que consuma um elemento do tipo `alunoPos`, e retorne verdadeiro caso o aluno esteja cursando a pós graduação na mesma instituição de ensino em que cursou a graduação. Teste o programa nas estruturas do exercício anterior.

## 7. Tipos Mistos

48. [Fácil] Avalie as expressões a seguir e explique as mensagens de erro:
- (a) `(number? '10);`
  - (b) `(symbol? 'um-simbolo 'outro-simbolo);`
  - (c) `(posn? um-posn);`
  - (d) `(struct=? "(make-posn 1 4)");`
  - (e) `(boolean? 'true).`
49. [Fácil] Altere as expressões do exercício anterior para que elas retornem verdadeiro.
50. [Médio] Uma fábrica de copos quer que você desenvolva um programa para calcular o volume que seus copos possuem. Os copos desenvolvidos pela empresa possuem duas formas geométricas distintas: cilíndricos e paralelepípedais. Com base nessa descrição, faça o que é pedido:
- (a) Defina a estrutura `copo-cilíndrico`, que possui os atributos raio e altura.
  - (b) Defina a estrutura `copo-paralelepipedal`, que possui comprimento, altura e largura.
  - (c) Desenvolva a função `calcula-volume`, que consome um `copo-cilíndrico` ou `copo-paralelepipedal` e retorna o seu volume comportado em  $\text{cm}^3$ . Para calcular o volume de um `copo-paralelepipedal`, utilize  $V = c \times a \times l$ ; e para o volume de um `copo-cilíndrico`,  $V = \pi \times r^2 \times h$ .
51. [Médio] Uma locadora de veículos precisa de um programa para calcular o valor da diária de locação de seus veículos. A empresa dispõe de dois tipos de veículos: carros e motos. Ambos os tipos de veículos possuem os atributos em comum: ano, modelo (descrição) e valor de mercado. Os carros ainda podem possuir alguns atributos adicionais, como:



ar condicionado, direção hidráulica e vidros elétricos. O valor base cobrado pela diária é de 0.25% do valor de mercado do veículo. Os carros ainda possuem um acréscimo de R\$30,00 por opcional. Já as motos, possuem apenas uma taxa fixa de seguro, no valor de R\$70,00, que deve ser acrescida no valor base da diária. Com base nessa descrição, faça o que é pedido:

- (a) Defina a estrutura `carro`.
- (b) Defina a estrutura `moto`.
- (c) Crie pelo menos 3 exemplos para cada uma das estruturas definidas.
- (d) Crie a função `calcula-diária` para efetuar o cálculo da diária de locação.
- (e) Teste a função `calcula-diária` nos exemplos anteriormente desenvolvidos.

## 9. Listas

- 52. [Fácil] Considere as seguintes linguagens de programação: C, C++, C#, Java, Javascript, PHP, Ruby, Perl, R, Matlab, DrRacket e Python. Construa uma lista de símbolos para armazenar todas as linguagens. Considere a criação da lista utilizando `cons` e `list`.
- 53. [Fácil] Construa uma função que consuma a lista de símbolos construída no exercício anterior, e retorne a quantidade de elementos desta lista.
- 54. [Difícil] Desenvolva um programa que consuma uma lista de salários e verifique se todos os salários são superiores ao salário mínimo (também consumido pelo programa).
- 55. [Difícil] Considere a lista `(cons 1 (cons 'dois (cons "três" (cons 4 (cons 'cinco (cons 6 empty))))))`, e faça o que é pedido a seguir:
  - (a) Desenvolva uma função para contar a quantidade de elementos na lista.
  - (b) Desenvolva uma função para computar a quantidade de elementos do tipo símbolo na lista.
  - (c) Desenvolva uma função para computar o somatório dos elementos do tipo número da lista.
  - (d) Desenvolva uma função que retorne uma nova lista sem elementos do tipo número.
- 56. [Fácil] Desenvolva a função `inteiros-zero-até-n` : `Número -> Lista-de-Números` que consome um número inteiro positivo  $n$  e retorna uma lista com todos os números inteiros  $n$  até zero.
- 57. [Médio] Desenvolva a função `inteiros-até-n` : `Número -> Lista-de-Números` que consome um número inteiro positivo  $n$  e retorna uma lista com todos os números inteiros de 0 até  $n$ .
- 58. [Fácil] Uma lista de chamadas é composta pelos nomes dos alunos de determinada turma. Escreva uma função que, dada uma lista de chamadas e o nome de um indivíduo, verifique se o mesmo está matriculado na turma em questão.
- 59. [Médio] Desenvolva uma função que, dada uma lista de idades, verifique se todos os indivíduos são maiores de idade. Após isso, modifique a função anterior para que elimine da lista os indivíduos idosos (com idade maior ou igual que 60 anos).

60. [Médio] Um veículo é composto por um modelo, marca e ano de fabricação. Uma lista de veículos é composta por elementos do tipo veículo. Escreva uma função que receba uma lista de veículos e elimine veículos fabricados antes do ano de 1990.
61. [Difícil] Desenvolva um programa que consuma uma lista de veículos (exercício anterior) e retorne o veículo mais antigo.
62. [Médio] Escreva uma função que consuma uma lista de pessoas (altura e peso) e retorne uma lista de valores correspondentes ao índice de massa corpórea - IMC ( $IMC = peso/altura^2$ ).
63. [Médio] Uma estrutura televisor possui um tamanho, dado em polegadas e um valor. Escreva uma função que, dada uma lista de televisores, aumente em 7% o valor dos elementos com tamanho maior ou igual a 29 polegadas.
64. [Difícil] Escreva uma função que, dada uma lista de televisores (exercício anterior), ordene os elementos ascendentemente de acordo com o tamanho.

## 14 e 15. Definições de Dados Autorreferenciáveis

65. [Médio] Escreva a função `árvore-espelhada` que cria uma imagem de espelho de uma árvore binária (AB). Dado uma determinada AB, esta função deverá gerar uma nova AB, de modo que os nós esquerda e direita de um mesmo nível da árvore sejam permutados. A Figura 1 apresenta um exemplo de árvore consumida (esquerda) e de sua respectiva saída espelhada (direita). Para resolver este problema, considere a seguinte definição de dados:

```
(define-struct nó (id nível esq dir))

; Uma AB (árvore binária) é:
; 1. empty
; 2. (make-nó id info esq dir), onde
; id : String (representa o id do nó)
; nível : Número (Nível do nó)
; esq : AB (representa a subárvore esquerda do nó)
; dir : AB (representa a subárvore direita do nó)
```

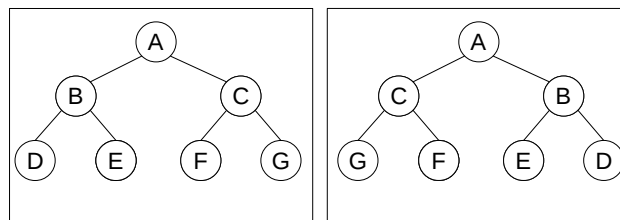


Figura 1: Árvore binária original (esquerda) e sua versão espelhada (direita).

66. [Médio] Considere a estrutura de nó utilizada no exercício anterior para construir a árvore binária (AB) apresentada na Figura 2. O nível de um nó representa a sua altura em relação ao nó raiz da AB (nó raiz possui nível zero). A partir desta descrição, faça o que é pedido:
- (a) Desenvolva uma rotina para computar a altura desta árvore.
  - (b) Desenvolva uma rotina para computar o nível em que se encontra o nó de menor nível que possui ambos os filhos vazios.
  - (c) Modifique o programa do item (b) para que ele encontre o nó de maior ou menor nível que não possui filhos. Utilize função de alta ordem para que a rotina se torne genérica.
  - (d) Modifique o atributo “nível” de todos os nós da árvore de modo que seus valores não correspondam com os seus reais níveis. Em seguida, crie um programa que percorra todos os nós da AB e atualize o nível em que cada nó se encontra.

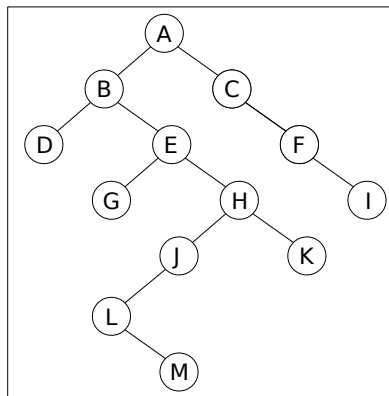


Figura 2: Árvore binária. Nó de maior nível, "M". Nó de menor nível sem filhos, "D".

67. [Médio] Considere uma árvore genealógica onde cada elemento (pessoa) armazena o nome, idade, sexo, seu pai e sua mãe. Crie um programa que percorra a árvore e informe a idade média dos indivíduos.
68. [Fácil] Considere a mesma árvore genealógica do exercício anterior. Escreva uma função que, dada uma pessoa, verifique se algum dos seus ancestrais consiste em uma mulher com mais de 40 anos de idade.
69. [Médio] Crie uma função que receba uma AB cujo id dos elementos seja um número inteiro. A função deve apresentar a soma dos ids pares da árvore.
70. [Difícil] Crie uma função que recebe uma ABP e os dados de um novo nó. A função deverá inserir o novo nó na respectiva posição da árvore.
71. [Médio] Desenvolva um programa que receba uma ABP e um número. A função deverá retornar a soma de todos os ids maiores que o número passado.
72. [Fácil] Considere uma árvore genealógica e escreva uma função que, dada uma pessoa, verifique se algum dos seus descendentes consiste em uma mulher com mais de 40 anos de idade.

73. [Médio] Crie uma função que receba uma árvore genealógica e conte quantos descendentes possuem mais de 30 anos de idade.
74. [Médio] Crie uma função que, dada uma árvore genealógica, aumente a idade em 5 anos da pessoa cujo nome seja passado como parâmetro.

## 19 e 20. Funções de Alta Ordem

75. [Fácil] Pesquise na documentação do *DrRacket* (<http://mirror.racket-lang.org/releases/6.2/doc/>) ou no livro texto da disciplina (<http://htdp.org/>) as funções de alta ordem apresentadas abaixo. Para cada uma delas, defina o seu contrato e de exemplos.

- (a) `andmap`;
- (b) `ormap`;
- (c) `apply`;

**Nota:** Como algumas destas funções só estão disponíveis na linguagem **Intermediate Student**, é necessário modificar a linguagem no *DrRacket*.

76. [Médio] A expressão `lambda`, disponível na linguagem **Intermediate Student with lambda**, é uma forma de definir funções anônimas, isto é, uma função sem nome. Isto posto, defina em que situação é necessário fazer uso de tal expressão e apresente pelo menos dois exemplos práticos de utilização.

**Dica:** Ver o capítulo 24 do livro texto da disciplina (<http://htdp.org/>) para mais informações sobre expressões `lambda`.

77. [Difícil]

O *DrRacket* possui diversas funções abstratas para processar listas. Uma lista com as principais funções pode ser encontrada na Seção 21.2 do livro texto da disciplina (<http://htdp.org/>). O objetivo destas listas é simplificar muitas tarefas de programação e facilitar a leitura de programas. Isto posto, utilizando as funções de alta ordem presentes no livro (`map`, `filter`, `foldr` e as demais que julgar necessário) desenvolva algoritmos que consomem uma lista de números e realizam os seguintes cálculos:

- média aritmética  $\left(\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i\right)$ ;
- variância amostral  $\left(\sigma^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2\right)$ ;
- desvio padrão amostral quando não se sabe a média  $\left(\sigma = \sqrt{\frac{\sum x_i^2 - \frac{1}{n} (\sum x_i)^2}{n-1}}\right)$ .

Neste exercício você deve implementar algoritmos iterativos para realizar o cálculo de cada função. Logo, não há necessidade de utilizar recursão em nenhum dos casos. A equação  $\sqrt{\sigma^2}$  pode ser utilizada nos exemplos para validar a implementação do algoritmo, mas não deve ser utilizado na função responsável por calcular o desvio padrão. Assuma em todos os casos que a lista de números possui pelo menos dois números.

78. [Médio] Utilizando os conceitos de funções de alta ordem e abstração funcional, crie o programa *calculadora-estatística* : (*Lista-de-Números* -> *Número*)  
*Lista-de-Números* -> *Número* que dado uma lista de números calcula a média aritmética, variância amostral ou desvio padrão. Utilize as implementações das funções utilizadas no exercício anterior.
79. [Fácil] Utilizando definições locais, crie uma função que, dada uma lista de nomes de pessoas e um nome, verifique se o nome se encontra na lista em questão.
80. [Médio] Utilizando definições locais, crie um programa que some todos os números de uma lista menores que um valor, o qual deve ser passado como parâmetro juntamente com a lista.
81. [Médio] Crie uma função de alta ordem que receba uma função e uma lista quaisquer. A função deverá retornar uma lista do mesmo tipo da lista passada. Por exemplo, para uma lista de números e uma função soma-dois, o resultado deverá ser a lista original com seus valores somados com dois.
82. [Difícil] Considere uma estrutura para armazenar dados de uma pessoa (nome, idade, peso e sexo). Crie uma função que consuma uma função de seleção e uma lista de pessoas, retornando uma lista com as pessoas selecionadas. Por exemplo, a função de seleção pode considerar apenas pessoas do sexo feminino, ou apenas maiores de idade.
83. [Fácil] Considerando que o peso é determinado pelo produto da massa e da gravidade, crie as funções *peso-na-terra*, *peso-na-lua* e *peso-em-jupiter*. As funções deverão calcular o respectivo peso, dadas as gravidades de 9,8 para a Terra, 1,67 para a lua e 22,9 para Júpiter. Crie uma função de alta ordem que, dada a massa de uma pessoa e a função desejada, calcule seu peso.
84. [Médio] Crie as funções *media-aritmetica*, *media-harmonica* e *media-geometrica*. Após isso, crie uma função *calcula-media*, que recebe uma função de cálculo de média e uma lista de números, retornando o respectivo valor de média.

## 25. Recursão Generativa

85. [Médio] Diferencie recursão generativa de recursão estrutural. Para cada uma, apresente pelo menos um exemplo.
86. [Médio] O Restaurante Universitário da UFRGS está tomando algumas medidas para diminuir o tempo que seus usuários permanecem na fila. Uma análise constatou que o crescimento do tempo de espera nas filas está relacionado com o tempo gasto pelos funcionários dos caixas para fazer o troco. Uma das medidas propostas para minimizar este problema é a instalação de caixas eletrônicos na parte externa dos restaurantes, possibilitando aos usuários sacar dinheiro em moedas. Sua tarefa é desenvolver o software que realiza o saque. O programa deverá consumir uma lista com os valores de moedas disponíveis na máquina (ex. um centavo, cinco centavos, dez centavos e etc.) e o montante a ser sacado (em centavos). No momento do saque, o programa deverá utilizar sempre a menor quantidade possível de moedas. Considere que a máquina possui uma quantidade ilimitada de moedas de cada tipo. Utilize apenas moedas brasileiras (1, 5, 10, 25, 50 e 100 centavos).

O exemplo a seguir apresenta uma execução do programa para realizar o saque do montante “137 centavos”.

```
(make-change (list 50 25 10 5 1) 137))
```

esta chamada deverá produzir: `(list 50 50 25 10 1 1)`.

87. [Difícil] Crie uma função que receba uma lista de números e retorne uma lista com os números ordenados.
88. [Difícil] Crie uma função que, dada uma lista de números, retorne a soma dos dois menores números desta lista. Para isso, ordene a lista com o algoritmo *quick-sort*.
89. [Difícil] Crie uma função que consuma duas listas de números. A função deverá retornar os elementos das duas listas em uma única lista em ordem crescente.
90. [Difícil] Escreva um programa que consuma duas listas de números. O programa deverá somar os dois maiores elementos da primeira lista com os dois menores elementos da segunda lista, retornando assim um número.

## 28. Algoritmos de Backtracking

91. [Fácil] Dado um grafo, escreva uma função que devolva uma lista com os nodos de grau superior a 2.
92. [Difícil] Escreva uma função que receba como parâmetros um grafo não-cíclico, um nodo de origem e um nodo de destino. O algoritmo deverá retornar um caminho possível do nodo de origem ao nodo de destino. Caso não exista, a função deve retornar *false*.
93. [Difícil] Modifique o programa da questão anterior para que calcule caminhos mínimos em grafos cíclicos. Teste a função com um par de nodos origem-destino com formação de ciclos.
94. [Difícil] Crie uma função que, dado um grafo e um nodo de origem, liste todos os nodos alcançáveis a partir do nodo de origem.