

Disciplina: Paradigmas de Programação
Professor: Maicon Rafael Zatelli
Entrega: Moodle

Atividade III - LISP

Atenção: Faça um ZIP com todos os arquivos de solução. Use o nome do arquivo de maneira a entender qual problema você está resolvendo. Por exemplo, problema1.lisp, problema2.lisp e assim por diante.

Resolva os seguintes problemas na linguagem LISP:

1. Crie uma estrutura de dados para pontos 2D, a qual deve possuir os campos *x* e *y*.
 - A:** Crie uma função `distancia (a b)`, a qual recebe dois pontos como parâmetro e deve retornar a distância entre eles.
 - B:** Crie uma função `colineares (a b c)`, a qual recebe três pontos como parâmetro e deve retornar se eles são colineares ou não. DICA: os três pontos serão colineares se o determinante de suas coordenadas for igual a 0.
 - C:** Crie uma função `formaTriangulo (a b c)`, a qual recebe três pontos como parâmetro e deve retornar se eles podem ser usados para formar um triângulo.
2. Modifique o arquivo `arvore.lisp` (disponível no Moodle) de forma a adicionar novas operações a nossa árvore. Assuma que nossa árvore não é uma árvore binária de busca.
 - A:** Crie uma função com a seguinte assinatura: `ocorrenciasElemento (arv x)`, a qual recebe um número e deve retornar a quantidade de ocorrências dele na árvore.
 - B:** Crie uma função com a seguinte assinatura: `maioresQueElemento (arv x)`, a qual recebe um número e deve retornar a quantidade de números maiores que ele na árvore.
 - C:** Crie uma função com a seguinte assinatura: `mediaElementos (arv)`, a qual deve retornar a média dos números na árvore.
 - D:** Crie uma função com a seguinte assinatura: `quantidade (arv)`, a qual deve retornar a quantidade de elementos na árvore.
 - E:** Crie uma função com a seguinte assinatura: `elementos (arv)`, a qual deve retornar uma lista com todos os elementos na árvore.
 - F:** Crie uma função com a seguinte assinatura: `substituir (arv x y)`, a qual recebe dois números como parâmetro e deve buscar e alterar todas as ocorrências de *x* pelo valor *y*.
 - G:** Crie uma função com a seguinte assinatura: `posordem (arv)`, a qual deve retornar a lista de números visitados em percorrimento pós-ordem.
 - H:** Crie uma função com a seguinte assinatura: `preordem (arv)`, a qual deve retornar a lista de números visitados em percorrimento pré-ordem.
 - I:** Crie uma função com a seguinte assinatura: `emordem (arv)`, a qual deve retornar a lista de números visitados em percorrimento em-ordem.
 - J:** Crie uma função com a seguinte assinatura: `subtraiParesImpares (arv)`, a qual deve retornar a soma de todos os números pares - (menos) a soma de todos os números ímpares.
 - K:** Crie uma função com a seguinte assinatura: `novoNo (x)`, a qual recebe um número como parâmetro e deve retornar uma nova árvore com apenas o nó contendo o valor *x*.
 - L:** Crie uma função com a seguinte assinatura: `inserir (arv x)`, a qual recebe um número como parâmetro e deve inseri-lo na árvore respeitando as seguintes condições:
 - Tente adicioná-lo primeiramente em um nó interno com **ou** a árvore direita **ou** a árvore esquerda nula (mas não ambas nulas).
 - Se não for possível, transforme uma folha em um nó interno adicionando *x* **ou** no lado esquerdo **ou** no lado direito do nó.

DICA: crie algumas funções auxiliares para facilitar.