



**TRABALHO**  
**Linguagens de Programação**  
**Prof. Lucas Ismaily**

**SOBRE ORIENTAÇÃO A OBJETOS [10 pontos]**

1. Yoda tem uma pequena livraria e gostaria de criar um pequeno sistema que torne automático o processo de vendas e compras dos livros. Na livraria do Yoda tem três tipos de livros, a saber: aventura, drama e comédia. Todos os livros possuem nome, valor, quantidade em estoque e um identificador único. Os livros de aventura possuem ilustrações; os livros de drama possuem capa dura; e os de comédia possuem capa tipo brochura. Inicialmente, o Yoda não quer armazenar os seus clientes, porém gostaria de efetuar e guardar as vendas, consultar qual o seu estoque de livros, quanto dinheiro ele tem em caixa, além de efetuar compras de novos livros.

Sua tarefa é implementar o sistema da livraria do Yoda. **Não precisa utilizar banco de dados. Também não precisa criar uma interface gráfica. Contudo, é necessário uma navegabilidade no sistema mesmo que via teclado.**

**SOBRE LINGUAGEM FUNCIONAL [10 pontos]**

Você pode utilizar Python (de modo funcional) ou qualquer outra linguagem funcional para resolver as seguintes questões. Importante: é **expressamente proibido** o uso de instruções de repetição, atribuições de variáveis para usos futuros (variáveis auxiliares). Em resumo, o seu código deve ser o mais funcional possível. Ademais, o uso da instrução *if/else* é liberado.

2. Faça um programa que recebe um inteiro  $n$  e imprima os  $n$  primeiros termos da sequência de Fibonacci.
3. Faça um programa que, dado uma lista numérica, imprima uma 2-tupla, tal que contenha a média aritmética dos elementos esteja na primeira posição e o desvio padrão na segunda posição.
4. Faça um programa que recebe duas listas  $A$  e  $B$  e imprime a diferença simétrica de  $A$  e  $B$ . A diferença simétrica de  $A$  e  $B$  é dada por  $(A-B) \cup (B-A)$ .
5. Faça um programa que recebe uma lista de inteiros  $L$  e dois números  $p$  e  $q$  inteiros e imprima a lista  $L$  sendo que toda ocorrência de  $p$  deve ser trocada por  $p+q$ .



6. Faça um programa que recebe um inteiro  $n$  e uma *string* como entrada e imprime uma nova *string* com cada vogal (letras 'a', 'e', 'i', 'o', 'u' minúsculas ou maiúsculas) repetida  $n$  vezes, os outros caracteres devem permanecer inalterados.
7. Faça um programa que dada duas listas numéricas ordenadas  $L_1$  e  $L_2$ , imprime um *merge* destas, ou seja, imprime uma nova lista ordenada  $L_3$  resultante da junção de  $L_1$  e  $L_2$ .
8. Faça um programa que recebe um inteiro positivo  $n$  e uma *string*  $W$  e determina se  $W$  contém uma *substring* de tamanho maior ou igual a  $n$  que é palíndroma. A saída deve ser apenas as palavras *sim* ou *nao*.
9. Faça um programa que recebe uma lista numérica  $L$  e imprime  $L$  ordenada. Não é permitido utilizar funções nativas de ordenação.
10. Faça um programa que recebe uma lista de inteiros e imprime somente os números que são primos.
11. Faça um programa que recebe como entrada coordenadas de um retângulo no plano e um ponto  $p$  no plano, e imprima 'sim' se o ponto  $p$  está dentro do retângulo e 'nao' caso contrário. Suponha que o retângulo seja representado por uma tupla ( int, int, int, int), contendo respectivamente as coordenadas  $x$  e  $y$  do ponto no seu canto inferior esquerdo, seguidas das suas medidas de largura e altura. Sabendo que o eixo  $y$  cresce de baixo para cima e o eixo  $x$  da esquerda para direita.

**OBS.:** A entrada de todas as questões é pelo teclado (entrada padrão), e é digitado apenas a entrada, sem nenhum caractere especial. Por exemplo, se a entrada for uma lista numérica, deverá ser digitado 1 2 3 5 4 e não [1, 2, 3, 5, 4], ou seja, apenas os números separados por espaço são digitados. Outrossim, o único valor impresso na tela deve ser o resultado. Qualquer impressão diferente do resultado, como por exemplo, "digite a entrada" ou "a saída é" será considerado **resposta errada**. Novamente, imprima apenas o resultado. Por exemplo, na Questão 9, se a entrada for 1 2 11 6, a única impressão na tela será 1 2 6 11. Caso a resposta seja vazia, você deve imprimir apenas um espaço em branco. Por exemplo, se na Questão 10 não houverem números primos, sua impressão será apenas um espaço em branco. Ademais, nas respostas que são listas, a única coisa impressa são os elementos de forma horizontal e separados por espaços, por exemplo, 1 2 3 4 em vez de [1, 2, 3, 4]. Novamente e repetidamente, qualquer impressão ou entrada de dados que não seguir o padrão explicado aqui, será considerado **resposta errada**, se está em dúvida quanto a saída e/ou entrada dos dados, pergunte.



### SOBRE PROGRAMAÇÃO LÓGICA [10 pontos]

12. Utilizando Prolog implemente um predicado *primos*(  $L$  ) que determina se todos os números na lista  $L$  são primos.
13. Utilizando Prolog implemente o predicado *n\_esimo*(  $N$  ,  $L$  ,  $X$  ) , onde  $X$  é o  $N$ -ésimo elemento da lista  $L$ .
14. Utilizando Prolog implemente o predicado *duplica\_X*(  $X$  ,  $L_1$  ,  $L_2$  ), onde  $L_2$  é a lista  $L_1$  com todas as ocorrências do elemento  $X$  duplicados.
15. Utilizando Prolog implemente o predicado *tamanho*(  $L$  ,  $X$  ), onde  $X$  é comprimento da lista  $L$ .
16. Utilizando Prolog implemente o predicado *media\_desvio\_padrao*(  $L$  ,  $M$  ,  $DP$  ), onde  $M$  é média e  $DP$  é o desvio padrão dos elementos da lista  $L$ .
17. Utilizando Prolog implemente o predicado *n\_ocorrencias\_X*(  $L$  ,  $X$  ,  $N$  ), onde  $N$  é o número de ocorrências de  $X$  na lista  $L$ .
18. Utilizando Prolog implemente o predicado *moda*(  $L$  ,  $X$  ), onde  $X$  é a moda da lista  $L$ . A *moda* de uma lista é o elemento com o maior número de ocorrências.
19. Implemente o predicado *elimina\_repetidos*(  $L_1$  ,  $L_2$  ) que recebe em  $L_2$  o resultado de  $L_1$  sem elementos duplicados. Por exemplo, se  $L_1 = [ 1 , 1 , 1 , 5 , 5 ]$ , então  $L_2 = [ 1 , 5 ]$ .
20. Implemente um programa em Prolog sobre a seguinte família:  
  
Construa um banco de fatos de uma família. De modo que seja possível representar os seguintes predicados: sexo (masculino ou feminino), irmã, irmão, descendente, mãe, pai, avô, avó, tio, tia, primo e prima.
21. Utilizando Prolog implemente o predicado *ultimo\_primeiro\_maior\_menor*(  $L$  ,  $U$  ,  $P$  ,  $M$  ,  $N$  ) sendo  $U$  o último,  $P$  o primeiro,  $M$  o maior e  $N$  o menor elemento da lista  $L$ .



## INFORMAÇÕES IMPORTANTES

A nota do código do trabalho será a média aritmética das notas nos três paradigmas: Orientação a Objetos, Funcional e Lógico. A nota será validada de acordo com o resultado obtido na apresentação. A apresentação será simples, farei apenas algumas perguntas. Não será apresentado para toda a turma. Caso não apresente, receberá **nota zero**.

A entrega será **somente** pelo SIPPA, numa pasta zipada com todas as questões (no caso da questão de Orientação a Objetos, você pode zipar o projeto). Nota: cada questão de funcional e lógico deve estar em um arquivo separado. O prazo máximo da entrega do trabalho é para o dia **04/12/2017**. As apresentações serão agendadas por e-mail.

**Trabalho individual.** Sejam honestos com vocês e comigo, por favor. Se for detectado qualquer tipo de fraude, os envolvidos receberão nota **zero**. **Note: os envolvidos receberão zero, não importa se você foi a origem ou o destino, ambos receberão nota zero.**