

TRABALHO DE PROGRAMAÇÃO FUNCIONAL PARA A SEGUNDA AVALIAÇÃO

Obs.1: Vocês devem utilizar nestes trabalhos tanto listas compreensivas quanto funções de alta ordem

Obs.2: Para as todas as funções deve-se determinar os tipos das entradas e da saída

- 1) Faça um programa em Haskell que leia uma lista de strings e então faça uma função para cada item que:
 - (a) devolva uma lista de tuplas, onde cada tupla deve ser composta por uma string da lista dada e número de vogais da string;
 - (b) devolva uma lista string contendo as strings que seu tamanho seja maior do que 5 e que inicie com vogais.
 - (c) devolva a lista de string substituindo as vogais em cada string por @.
- 2) Faça um programa em Haskell que leia duas listas de strings e então faça pelo menos uma função para cada item que:
 - (a) devolva uma lista ordenada contendo strings das duas listas onde as mesmas deve ter tamanho ≥ 4 .
 - (b) devolva a quantidade de strings das duas listas que iniciam com vogais.
 - (c) devolva uma lista contendo strings das duas listas que iniciam e terminam com vogal.
- 3) Faça um programa em Haskell que leia duas listas ordenadas de inteiros e então faça pelo menos uma função para cada item:
 - (a) devolva duas listas, a primeira contendo os números das posições pares maiores do que 50 e a segunda os elementos ímpares menores que 200.
 - (b) devolva o produto dos elementos das duas listas dos múltiplos de 3 > 50 e dos múltiplos de 7 menos do que 200.
 - (c) Devolva uma lista ordenada contendo elementos das duas listas que sejam maiores do que 50 e que sejam ímpares múltiplos de 3.
- 4) Faça um programa em Haskell que leia uma lista de times de futebol contendo nome do clube, estado e país a qual pertence e ano de fundação do clube. Faça uma função que ordene a lista pelo campo nome do clube usando o Quicksort, e depois possibilite ao usuário ver toda a lista e permita também o usuário buscar informações de um clube pelo nome do clube. (1,5 pontos)
Obs.: cuide para não comparar minúsculas com maiúsculas.

- 5) Faça um programa em Haskell, que cadastre as seguintes lista: alunos(Matricula, Nome, Curso, Período), curso(Código, Nome, Quantidade de Períodos), Disciplinas(Código da Disciplina, Código do Curso, Nome Disciplina, Período) e Notas(Matricula, Código Disciplina, Nota1, Nota2). Lembre-se que o aluno só pode ser cadastrado em um curso já cadastrado, que uma disciplina só pode ser cadastrada em um curso já cadastrado e que notas só podem ser cadastradas para aluno e disciplinas já cadastrados. Caso o usuário queira cadastrar aluno ou disciplina ou notas que não tenham todos os itens cadastrados favor emitir uma mensagem para usuário solicitando o cadastro do mesmo. Não é necessário cadastrar as duas notas para uma disciplina, caso somente uma nota seja cadastrada deverá ser a nota1, além disso se uma nota não for cadastrada o valor deve se -1. Depois permita o usuário verificar os cursos cadastrados, os alunos de um determinado curso, e de um determinado período, as disciplinas de um determinado curso e também de um determinado período do curso. Além disso, permita visualizar as notas de um determinado aluno.
- 6) Refaça o exercício anterior agora colocando os dados em um arquivo. (Ponto extra)

Equipe: os programas podem ser feitos em dupla, mas os relatórios são individuais. Se os programas forem feitos em dupla, a dupla deve ser identificada no envio do código.

Data de Entrega: data agendada na plataforma

Entregar: Código Fonte, Relatório(Conforme Modelo em PDF)

Forma de Entrega: pelo SIGAA, caso tenha algum problema enviar por e-mail(julianaoc@ufpi.edu.br).

Entrevista Individual: agendar horário com a Professora.

Processo da Entrevista: para cada aluno será sorteado 2 programas, dentre os entregues pelo aluno, para que o mesmo explique os 2 programas.