

Relatório do trabalho de programação funcional
Aluno: Marcos Vinicius Batista Sampaio
Códigos realizados junto com: ISDAEL RODRIGUES OLIVEIRA

Resumo:

Este relatório trata da descrição detalhada do funcionamento dos códigos implementados para responder as questões passadas como trabalho avaliativo que conta com cinco questões.

Introdução:

Como relatado no título, o principal paradigma usado é o funcional que é visto na disciplina de programação funcional. Com questões sendo desenvolvidas na linguagem Haskell, também muito utilizada em universidades para introdução do paradigma, o relatório tem a seguir: Seções específicas onde discutirei mais sobre como as questões foram resolvidas, apresentando o funcionamento, Resultados que mostra as saídas das questões de acordo com o explicado e pedido no trabalho.

Seções Específicas:

Questão 1:

Para iniciar a questão 1 contem 3 funções, uma main que executa o código uma auxiliar para imprimir as listas e a principal de funcionamento que é a fatorar. Na função main é feita a inserção dos 3 números que devemos fatorar e em seguida passamos esses números como inteiros realizando a transformação com o read para a função fatorar. A função fatorar recebe 5 parâmetros, os 3 números digitados pelo usuário, um fatorial padrão e um numero de contador, por padrão começa passando sempre o inicio da fatoração 2 e o contador 0, `fatorar(num1, num2, num3, fat, cont)`, logo feito isso a função contem 9 opções, onde atendido as condições pode ser chamado o próprio fatorial como recursão onde acontece a divisão do numero pelo fat e a passagem do fat e do contador de divisões. Há uma opção caso todos sejam indivisiveis pelo numero fatorial é realizado uma chamada do com o fatorial aumentando em 1 o fat e depois a lista e a primeira opção para sair do loop de recursão que se todos forem 1, o resultado é chamar a lista com as divisões e suas quantidades.

Questão 2:

Temos aqui a necessidade de três funções, sendo uma main responsável por solicitar todos os custos em relação a apresentação e fazer as chamadas iniciais das outras funções, uma função que calcula a renda por seção e uma que faz o calculo de quantas seções são necessárias para obter lucro. A main é composta por quatro `getLine`, respectivamente para custo geral da peça, custo por seção, valor do ingresso e a quantidade de assentos por seção, além disso na main é feita a chamada da função de seções que retorna a quantidade total de ingressos e depois a mesma função para retornar o numero de seções, todos os parâmetros são passados como float utilizando o `read`. A função `renda` tem três parametros float e retorna como resultado um float, é passado o valor do ingresso, quantidade de cadeiras e o valor geral da seção, após isso é feito a multiplicação do valor do ingresso pelo numeros de cadeiras e diminui o resultado pelo valor da seção, isso é feito para ter o resultado do lucro por seção dado o valor e a quantidade de cadeiras. A função `seção` calcula quantas seções serão necessárias para obter lucro, tem 5 parâmetros sendo ingresso, assentos, seção, investimento e o retorno, o retorno é passado com base na função `renda`, que passa seu retorno de função. Tem duas opções a função, uma calculando o retorno financeiro de lucro, se o retorno

passado pela renda for menos ou igual o investimento, é feito a soma feito através da chamada recursiva da própria função caso não, é retornado apenas a renda.

Questão 3:

Aqui temos o uso de uma função random para realizar o sorteio dos números aleatórios, essa função será usado para gerar dois números aleatórios, o da ana e o da beatriz, seguindo a carolina fica com a cadeira que sobrar, os números são gerados dentro da main de um a 100 em formato inteiro, após isso temos a função cadeira que define as cadeiras utilizando um mod de número três que funciona se o resto de mod três for zero, a cadeira selecionada é a dois, se o resto for um, a cadeira selecionada é a um e por fim se o resto de mod três for igual a 2 a cadeira selecionada é a zero. Para definir a segunda cadeira selecionada que pertence a beatriz é feita outra função que passa o retorno de duas chamadas da função cadeira com x e y, e dentro da função cadeirabeatriz é feito uma comparação, no primeiro caso se a for diferente de x a beatriz pode se sentar na cadeira sorteada, porque aqui indica que a cadeira sorteada é diferente da de ana, caso não, é feito a comparação, se a cadeira de ana for igual a um, beatriz senta na cadeira zero retornado pela função, se for igual a zero beatriz senta na cadeira dois que é retornado pela função, caso nenhuma dessas três opções seja a selecionada, entra no otherwise retornando a cadeira um para beatriz se sentar. Por fim existe a terceira função para decidir onde Carolina deve se sentar, é feito comparação de forma de união das duas variáveis utilizando o retorno da função cadeira e dependente do resultado, pode ser dois ou zero e no otherwise retornando um para ser decidido a última cadeira.

Questão 4:

A questão 4 é composta por alternativas a, b e c. Contem nove funções, verificaExistenciaNaLista tem como parametro um caractere e uma lista, enquanto a lista não for vazia é feita a chamada, contem dois pipes, usa tolower para comparar a com c que é a cabeça da lista e se for igual retorna false, caso não é feita a chamada da própria função com o resto da lista até a lista ser vazia. ContandoCaracteries conta com um parametro apenas que é uma lista até a lista ser vazia e encerrar a chamada, funciona com dois pipes sendo um dos pipes a chamada da função verificaExistenciaNaLista que falei antes, dependendo do retorno da lista é somado mais um para a contagem de caracteries com a chamada da função ou feita a chamada da função com o resto da lista. ContarLista tem como parametro uma lista e uma lista de lista podendo está vazia ou não dependendo do momento da chamada que influencia em qual parte do ContarLista pode cair, sendo que existem quatro variações, sendo [c:r]: [[]], [c:r]: v, [] e []: [], os dois primeiros casos contam os caracteries e imprime fazendo uso da função contandoCaracteries e as ultimas duas com listas vazias apenas imprime que não há mais palavras. Evogal é uma função que apenas compara um caractere passado com as letras vogais do alfabeto, para uso de saber se é ou não uma vogal no início da string ou se possui na string, caso exista retorna true ou false se não tiver. VerificaTipo faz uso de evogal e de funções já pronta do data.char como isDigit para saber se é dígito, islower para saber se a letra é minúscula, isupper para saber se é maiúscula e o último caso para caracteries especiais se usa o otherwise. DevolveTipos tem como parametro uma lista de lista, que faz o retorno dos tipos, para saber quais são a função faz uso da explicada anterior verificaTipo criando uma lista com os tipos. ContaVogais entra com uma lista e usa a evogal para saber quais são vogais, se sim soma mais um e chama o próximo item, caso não só é chamado o próximo item. Por último a devolveMaiorString1 que tem como entrada a lista de strings, usa a contagem de vogais para saber qual a maior, é feita a chamada duas vezes da conta vogais e compara qual é a maior pelos pipes, se o primeiro caso atende é retornado se não, só é feita a chamada da próxima string

Questão 5:

O problema conta com cinco funções incluindo a main, verificaIgualdade na qual você entra com um número e uma lista tem como propósito encontrar os números iguais para que

seja possível fazer a retirada dos numeros iguais, tem duas opções, se c for igual a v, retorna false caso otherwise, ele retorna a própria lista com recursão fazendo a chamada para o próximo item. A função unindoAeB entra com duas listas e faz a uniao em chamada recursiva já que a lista não tem mais os itens repetidos, até a lista ser vazia, a chamada é realizada. MaiorQueCubo tem três parametros, sendo um argumento e duas listas, contem duas opções a primeira sendo se c elevado a 2 for maior que a cabeça da lista ao quadrado recebe e chama a função em recursão até que os itens da lista se acabem e ela fique vazia.

Resultados da Execução do Programa:

Q1:

Entrada = números a fatorar	Saída = repetiu
15 33 46	23 → 1 11 → 1 5 → 1 3 → 1 2 → 1
9 15 23	23 → 1 5 → 1 3 → 2
20 15 16	5 → 1 3 → 1 2 → 4
49 42 45	7 → 2 5 → 1 3 → 2 2 → 1

Q2:

Entrada = custos	Saída
Custo da peça = 50000 Custo por seção = 10000 Valor ingresso = 50 Total de assentos = 300	3300 ingressos vendidos Total de 11 seções
Custo da peça = 100000 Custo por seção = 30000 Valor ingresso = 70 Total de assentos = 500	10500 ingressos vendidos Total de 21 seções
Custo da peça = 11000 Custo por seção = 5000 Valor ingresso = 40 Total de assentos = 200	800 ingressos vendidos Total de 4 seções

Q3:

O número que Ana sorteou foi	O número que Beatriz sorteou foi	A cadeira que Ana ira sentar é a cadeira	A cadeira que Beatriz ira sentar é a cadeira	A cadeira restante para Carolina foi a cadeira
------------------------------	----------------------------------	--	--	--

13	54	1	2	0
40	68	1	0	2
100	53	1	0	2
75	55	2	1	0
45	76	2	1	0

Q4:

Entrada	Saída A	Saída B	Saída C
"Isdael", "Josias", "Flavio", "vinicios"	A quantidade de caracteres presentes em Isdael é: 6 A quantidade de caracteres presentes em Josias é: 5 A quantidade de caracteres presentes em Flavio é: 6 A quantidade de caracteres presentes em vinicios é: 6	"Consoante maiúscula", "Consoante maiúscula", "Consoante maiúscula", "Consoante minúscula"	"vinicios"
[]	[]	""	Sem palavras aqui
"#\$%", "!%@", "Vinicios"	A quantidade de caracteres presentes em #\$% é: 3 A quantidade de caracteres presentes em !%@ é: 3 A quantidade de caracteres presentes em Vinicios é: 6	"Caractere especial", "Caractere especial", "Consoante maiúscula"	"Vinicios"

Q5:

Entrada	Saída
([2,6,7],[4,5,8])	(A-B) (B-A) == [2,6,7,4,5,8] O cubo dos primeiros elementos é: 72 Soma entre os quadrados dos elementos: 113