

Projeto 1

ATE meia noite de 26/8

Submeta o arquivo proj1.hs em texto com o programa via classroom.

1

Implemente a função **trocатodos** que recebe o valor `velho` e o valor `novo` e uma `lista` e retorna a lista com **todas** as instancias de `velho` na lista trocada por `novo`.

```
trocатodos 4 10 [1,2,3,4,5,4,6,7,4]
==> [1, 2, 3, 10, 5, 10, 6, 7, 10]
```

```
trocатodos 4 10 [1,2,3,5,6,7]
==> [1,2,3,5,6,7]
```

2

implemente a função **cumsum** que dado uma `lista` de números retorna a lista com a soma cumulativa desses números. (na lista retornarda, a posição *i* contem a soma dos elementos da lista original até a posição *i*)

```
cumsum [4]
==> [4]
```

```
cumsum [5,10,2,3]
==> [5,15,17,20]
```

Detalhes

As duas funções devem estar num mesmo arquivo submetido **tarefa1.hs**

Eu nao vou rodar o arquivo em batch assim não se preocupe com as mensagens de erro sobre a função `main`

Para as 2 funções, voce pode definir funções auxiliares (fora do corpo) ou funções locais mas as funções `trocатodos` recebe 2 valores e uma lista apenas, e a `cumsum` 1 lista apenas, respectivamente.

- As implementações não podem usar funções pre-definidas do Haskell, com a excessão dos operadores matemáticos e as funções `head` e `tail`. Se vc precisa de alguma função, precisa implementa-la
- haverá uma pequena perda na nota se as funções não usam o mecanismo de regras e `pattern matching`. Nao é necessario usar `guards`, mas pense em usa-los se for o caso.

