

Princípios de Programação

Trabalho para casa 1

Universidade de Lisboa
Faculdade de Ciências
Departamento de Informática
Licenciatura em Engenharia Informática

2018/2019

A. A cifra de César com uma chave numérica K converte cada caracter de uma string num outro caracter, saltando K posições no alfabeto. Neste exercício vamos usar a representação ASCII do alfabeto; por exemplo, a letra 'a' (ASCII 97) quando encriptada com $K = 3$ fica 'd' (ASCII 100). Para descodificar algo que esteja na cifra de César, basta andar K posições para trás.

Para converter um caracter no seu código ASCII pode utilizar a função `ord` constante no módulo `Data.Char`. Para a conversão inversa utilize a função `chr` presente no mesmo módulo. Exemplo:

```
> cifrarCesar 3 "Vamos atacar os gauleses!"  
"Ydprv#dwdfdu#rv#jdxohvhv$"  
> decifrarCesar 3 "Ydprv#dwdfdu#rv#jdxohvhv$"  
"Vamos atacar os gauleses!"
```

B. A cifra de César é muito simples. A de Obelix é ligeiramente mais sofisticada. Dada uma chave K , a primeira letra na *string* salta K posições, a segunda $K + 1$, a terceira $K + 2$ e por aí adiante. Para descriptar deverá efetuar o processo inverso. Exemplo:

```
> cifrarObelix 3 "Vai acordar o Panoramix!"  
"Yen&hkx|om\DEL.~0as\129\131\135w\132\129\145;"  
> decifrarObelix 3 "Yen&hkx|om\DEL.~0as\129\131\135w\132\129\145;"  
"Vai acordar o Panoramix!"
```

C. Foi recentemente provado que qualquer número positivo pode ser decomposto na soma de 3 números palíndromos. Um palíndromo é um número que fica igual quando escrito de trás para a frente, como por exemplo

1423241. Dado um número positivo, a função `tresPalindromos` devolve uma combinação de 3 palindromos cuja soma seja o número dado. Exemplo:

```
> tresPalindromos 8763  
(0, 535, 8228)
```

Notas

1. Os trabalhos serão avaliados automaticamente. Respeite os nomes e os tipos das *cinco* funções que tem de escrever.
2. Não se esqueça de juntar uma assinatura para cada função que escrever.
3. Para resolver estes problemas deve utilizar *apenas* a matéria dos dois primeiros capítulos do livro (“Starting Out” e “Types and Typeclasses”). Pode usar qualquer função constante no **Prelude** e no módulo `Data.Char`.
4. Lembre-se que as boas práticas de programação Haskell apontam para a utilização de várias funções simples em lugar de uma função única mas complicada.

Entrega. Este é um trabalho de resolução individual. Os trabalhos devem ser entregues no Moodle até às 23:55 do dia 10 de outubro de 2018.

Ética Os trabalhos de todos os alunos serão comparados por uma aplicação computacional. Lembre-se: “Alunos detetados em situação de fraude ou plágio, plagiadores e plagiados, ficam reprovados à disciplina (sem prejuízo de ser acionado processo disciplinar concomitante)”.