IFCE - Campus Maracanaú **Ciência da Computação**
Linguagens de Programação **Prof. Thiago Alves**

6ª Lista de Exercícios

Aluno(a): _____ Matrícula: _____

1. Uma tripla (x, y, z) de inteiros positivos é chamada pitagórica se $x^2+y^2 = z^2$. Usando compreensões de listas, defina a função `pyths :: Int -> [(Int,Int,Int)]` que mapeia um inteiro n para todas as triplas pitagóricas com componentes em $[1..n]$. Por exemplo:

   ```
   > pyths 5
   [(3,4,5),(4,3,5)]
   ```

2. Um inteiro positivo é perfeito se é igual a soma de todos os seus divisores, excluíndo o próprio número. Defina uma função `perfects :: Int -> [Int]` que retorna a lista de todos os números perfeitos até um dado limite n usando compreensões de listas. Por exemplo:

   ```
   > perfects 500
   [6,28,496]
   ```

3. Considere o problema de codificação de uma string a fim de disfarçar o seu conteúdo de leitores indesejados. Um método de codificação bem conhecido é a cifra de César, em homenagem a sua utilização por Júlio César. Para codificar uma string, César simplesmente substituía cada letra pela letra três lugares depois no alfabeto. Por exemplo, a string "haskell is fun" é codificada como "kdvnhoo lv ixq".

   De modo mais geral, o fator de desvio de três usado por César pode ser substituído por qualquer número inteiro entre um e vinte e cinco, dando assim vinte e cinco maneiras diferentes de codificar uma string. Por exemplo, com um fator de desvio de dez, a string original acima poderia ser codificada como "rkcuovv sc pex". Implemente a cifra de César para strings de caracteres minúsculas seguindo os passos a seguir:

   Usando `import Data.Char` e as funções `ord :: Char -> Int` e `chr :: Int - Char`, defina uma função `let2int :: Char -> Int` que converte uma letra minúscula entre 'a' e 'z' para o inteiro correspondente entre 0 e 25, e também uma função `int2let :: Int -> Char` que realiza a conversão oposta. Por exemplo,

   ```
   > let2int 'a'
   0
   > int2let 0
   'a'
   ```

   Usando essas duas funções, defina uma função `shift :: Int -> Char -> Char` que aplica o fator de deslocamento para uma letra minúscula. Por exemplo,

   ```
   > shift 3 'a'
   'd'
   ```

   Usando `shift`, defina uma função `encode :: Int -> String -> String` que codifica uma string usando um determinado fator de deslocamento e também uma função `decode :: Int -> String -> String` que decodifica a string.