

- (1) Implementar uma função que verifica se uma string contém um palindromo

```
palindromo :: String -> Bool
```

- (2) A soma do comprimento de qualquer dois lados de um triângulo é sempre maior do que o comprimento do terceiro lado. Fazer uma função que recebe o comprimento dos três lados de um triângulo e verifica essa condição

```
> verificaTriangulo 2 2 2
True
>verificaTriangulo 1 4 1
False
```

- (3) Fazer uma função que recebe a base e o expoente e calcula a potência:

```
> potencia 2 4
16
```

- (4) Implementar uma função que recebe como entrada um número `n` representando uma posição na sequência fibonnacci e calcula o `n`-ésimo elemento da sequência. Ex:

```
fib 0 = 0
fib 1 = 1
fib 2 = 1
fib 3 = 2
fib 4 = 3
...
```

A partir da terceira posição um elemento da sequência é sempre a soma dos dois anteriores

- (5) Implementar uma função que verifica se um número é primo

```
isPrime :: Int -> Bool
```

- (6) Faça uma função que recebe 3 números inteiros e devolve como resposta o maior deles

- (7) Faça uma função que recebe 3 números inteiros e devolve uma lista com esses números em ordem