Exercícios sobre a *Wolfram Language* (do *Mathematica*)

(1) A *Cifra de César*, método clássico de criptografia, consiste em trocar as letras de uma mensagem por letras do alfabeto que se encontram a certa distância da primeira.
Por exemplo, uma possível Cifra de César seria trocar A por D, B por E, C por F e assim sucessivamente.

(a) Consulte a documentação do Mathematica a respeito das função RotateLeft;
(b) Crie uma função *GerarCifra* que recebe como entrada um inteiro **n** de posições e devolve uma lista de regras de substituição que associa cada letra à letra que está **n** posições à frente no alfabeto. Por exemplo, *GerarCifra*[3] deve devolver regras de substituição referentes ao exemplo acima.
(c) Crie uma função *Encriptar* que recebe um string **s** e um inteiro **n** e codifica o string **s** de acordo com a cifra dada por *GeraCifra*[**n**].
(d) Crie uma função *Decriptar* que recebe um string **s** e um inteiro **n** e remove a codificação de **s**, supondo que esta foi codificada por *Encriptar*[**s**,**n**].
(e) Com **n**=5, use *Decriptar*, para decifrar a mensagem: "ujiwt ufzqt gfqgn ij tqnajnwf".

(2) A *sequência de Fibonacci* apresenta seus dois primeiros termos iguais a 1 e cada termo seguinte é igual à soma dos dois anteriores. Portanto, os 6 primeiros termos da Sequência de Fibonacci são: 1, 1, 2, 3, 5, 8, ...

(a) Crie uma função *SeqFibonacci*[*n*], **n** maior que 0, e devolve o **n**-ésimo termo da sequência de Fibonacci, gerado recursivamente.
(b) Adapte a função anterior, para que ela admita a forma *SeqFibonacci*[{*n*}], de maneira a gerar os **n** primeiros elementos da sequência de Fibonacci.

(3) A função *Length* no *Mathematica* é especialmente útil para sabermos o tamanho de uma lista. Assim, por exemplo, *Length*[{1,5,7}] gera a saída 3, enquanto *Length*[{}] gera a saída 0.

(a) Crie uma função *Contar* que recebe com entrada um inteiro **n**, e uma lista **L** de inteiros entre 1 e **n,** e devolve uma lista de tamanho **n**, onde o primeiro elemento é o número de vezes que o dígito 1 aparece em **L**, o segundo é o número de vezes que o dígito 2 aparece em **L**, e assim sucessivamente.
Por exemplo, *Contar*[5,{1,1,3,4,3,5}] deve gerar {2,0,2,1,1}.
(b) Teste a função *Contar* com a variável

```
listadeinteiros=RandomInteger[{1,50},100];
```

que será inicializada com uma lista de 100 inteiros entre 1 e 50.

(4) (a) Crie uma função *CompletarLista* que recebe uma lista **L** e um inteiro **n** maior que o tamanho da lista, e "completa" **L** até ela ficar de tamanho **n** da seguinte maneira: com 0s, se a soma dos elementos de **L** for par, e com 1s, se a soma dos elementos de **L** for ímpar.
Assim, por exemplo, *CompletarLista*[{1,2,3},5] gera a lista {1,2,3,0,0}.
(b) Crie uma função *PadronizarListas* que recebe uma lista de listas e faz com que todas fiquem com o tamanho da maior lista recebida usando a função *CompletarLista*.
(c) Crie uma função *SomarListas* que recebe uma lista de listas, completa-as pela função *PadronizarListas*, já que elas não necessariamente têm de ser do mesmo tamanho, e retorna uma lista com as somas dos números presentes nas posições homólogas de cada lista de entrada.