1. Desenvolva algumas funções para codificação e decodificação de strings. Escreva a função codificaRot13, que recebe uma string como argumento e retorna uma string codificada com o algoritmo *rot13*, que substitui cada caracter da string pelo valor do caractere mais treze, subtraindo vinte e seis caso o resultado seja maior que a última letra, de forma que "abCde" seja substituída por "noPqr", "kLmnoPq" seja substituída por "xYzabCd", e "UVWxyz" seja substituida por "HIJklm". Somente os caracteres alfabéticos não acentuados devem ser modificados. Por exemplo, se a string "Revolução de 1930" for passada como argumento para essa função, ele retornará "Eribyhçãb qr 1930". Uma característica interessante do algoritmo *rot13* é que, se uma string codificada por ele for passada de novo pelo próprio algoritmo, a string original será retornada. Escreva também uma função decodificaRot13 que seja somente uma chamada para a função codificaRot13.

2. O *algoritmo de César* de criptografia de strings é uma versão melhorada do algoritmo *rot13*: o seu funcionamento é o mesmo, só que, em vez de substituir cada caractere por um caracter treze posições depois, o algoritmo de César recebe um valor chamado *chave*, e usa esse valor como o número de posições que devem ser puladas para a criptografia. Por exemplo, se a chave for 1, o algoritmo pulará uma posição ao codificar as letras, então se a string passada for "Java", o resultado será "Kbwb". O algoritmo de decodificação deve receber a mesma chave, só que deve substituir os caracteres da string por valores em posições anteriores. Escreva uma função codificaCésar que implementa o algoritmo de César, recebendo como argumentos uma string e uma chave (valor numérico) e retornando a string criptografada. Essa função deve considerar que **somente** as letras não-acentuadas devem ser criptografadas, as letras acentuadas, números, espaços e outros símbolos devem continuar como estão. Escreva também uma função decodificaCésar.
   Dica: Para simplificar o algoritmo, considere que o valor da chave só pode estar entre 1 e 25. Existem ao menos duas maneiras de implementar esse algoritmo.

3. O *algoritmo das fatias* de criptografia usa como entrada uma string e um valor numérico, sendo que esse valor numérico (o número de fatias) deve ser bem menor que o tamanho da string. O algoritmo de codificação fatia a string de entrada, tomando caracteres de *N* em *N* posições, onde *N* é o número de fatias, formando *N* novas strings cada uma com o comprimento *M* onde *M* é o comprimento da string original dividido por *N*. As strings criadas assim são concatenadas, sendo o resultado da codificação da string original. Para decodificar uma string criptografada com esse algoritmo, é necessário ter o valor numérico. Por exemplo, para criptografar a string "Programação em Java" (19 caracteres) usando 4 como número de fatias, o primeiro passo seria completar a string de forma que tenha um número de caracteres múltiplo de 4, para "Programação em Java " (20 caracteres), para simplificar o algoritmo. Fatiando essa string, pegando de quatro em quatro caracteres, obtemos quatro novas strings:

   "Prçea"
   "raãmv"
   "omo a"
   "ga J "

   A string criptografada seria o resultado da concatenação dessas strings ou "Prçearaãmvomo aga J ". Para decodificar essa string, basta repetir o processo de codificação, mas usando *M* como o número de fatias, obtendo as strings:

   "Prog"
   "rama"
   "ção "
   "em J"
   "ava "

   cuja concatenação resulta em "Programação em Java ". Escreva a função codificaFatias que recebe uma string e um valor numérico, codificando a string usando o valor e retornando a string criptografada. Escreva também a função decodificaFatias que deve fazer o processo reverso.
   Dica: A função decodificaFatias pode ser uma chamada para a função codificaFatias com o valor numérico adequado.

4. O *algoritmo das pontas* de criptografia recebe uma string como argumento e produz uma outra string como resultado, e pode ser descrito da seguinte forma: enquanto a string de entrada contiver caracteres, remova o primeiro e o último caracteres da string de entrada e os coloque na string de saída. Dessa forma, se a string "Programação em Java" for entrada no algoritmo, este mostrará como saída a string "ParvoagJr ammea çoã". A decodificação de uma string pode ser feita da seguinte forma: crie duas strings temporárias, e para cada par de caracteres extraídos da string codificada de entrada adicione o primeiro no fim da primeira string e o segundo no início da segunda string. A concatenação das duas strings é o resultado da decodificação. Escreva as funções codificaPontas e decodificaPontas para codificar e decodificar uma string usando esse algoritmo.

Fonte: Exercícios extraídos da lista de exercícios do livro *'Introdução à Programação Orientada a Objetos*, ISBN 85-352-1206-X, Editora Campus.