

# Cifra de César

O objetivo deste exercício-programa é a construção de um algoritmo de criptografia. A palavra criptografia vem do grego *kryptós* (escondido) e *grápho* (escrita), e é a arte ou ciência de tornar incompreensível uma mensagem originalmente escrita com clareza. Esta é uma ciência muito antiga, e vem sendo usada principalmente para fins militares e diplomáticos.

## 1 Cifra de César

Um esquema de codificação de textos usado há mais de 2000 anos é a chamada Cifra de César. Esse esquema de codificação era utilizado pelo imperador romano para se comunicar com seus generais. O esquema de codificação funciona da seguinte forma.

Numa primeira etapa, substitua as ocorrências de brancos e sinais de pontuação por uma sequência de caracteres seguindo a seguinte tabela:

**Tabela 1**

(branco)	WBRW
,	WVRW
.	WPTW
;	WPVW
:	WDPW
!	WEXW
?	WINW
-	WHFW

Depois de feita essa substituição o texto terá apenas letras do alfabeto. Então, dado um natural  $k$  substitua cada letra  $i$  pela  $(i+k)$ -ésima letra do alfabeto (após a última letra segue-se a primeira novamente). Observe o exemplo a seguir:

Texto original:

ESSE EXERCICIO-PROGRAMA VAI SER MUITO LEGAL.

Após substituir-se os brancos e sinais de pontuação:

ESSEWBRWEXERCICIOWHFWPROGRAMAWBRWVAIWBRWSERWBRWMUITOWBRWLEGALWPTW

Então, usando-se  $k = 2$ , obtemos o seguinte texto codificado:

GUUGYDITYGZGTEKEKQYJHYRTQITCOCYDTYXCKYDTYUGTYDTYOWKVQYDTYNGICNYRVY

## 2 O que você deve fazer

Você irá programar um sistema completo de criptografia, ou seja, você deverá fazer um programa em linguagem C# que:

1. leia um string *s* contendo um texto;
2. leia um natural *k*;
3. leia uma opção: 1 para cifrar ou 2 para decifrar;
4. se a opção for 1, então seu programa deverá cifrar o texto contido no string *s* e imprimí-lo;
5. se a opção for 2, então seu programa deverá decifrar o texto contido no string *s* e imprimí-lo (com brancos e sinais de pontuação).

### 3 Como você deve fazer este exercício

Para fazer seu programa, você deverá implementar as seguintes funções (sub-rotinas):

- a) `void troca_sinal (string frase_sai, string frase_ent, char sinal, string pal)`

Percorre o texto armazenado em *frase\_ent*, trocando toda ocorrência do sinal de pontuação armazenado em *sinal* pela palavra guardada em *pal*. A frase obtida é armazenada em *frase\_sai*.

Por exemplo, a chamada

```
troca_sinal (frase, "AH! AH! AH!", '!', "WEXW");
```

coloca no vetor *frase* o string "AHWEXW AHWEXW AHWEXW".

- b) `void troca_todos_sinais (string frase_sai, string frase_ent)`

Troca todos os sinais da Tabela 1 que ocorrem na frase armazenada em *frase\_ent* pelos respectivos strings e coloca a frase obtida em *frase\_sai*.

Por exemplo, a chamada

```
troca_todos_sinais (frase, "FACIL? - SIMPLES!");
```

coloca no vetor *frase* o string "FACILWINWWBRWWHFWWBRWSIMPLESWEXW".

**OBS.:** Esta função deve **obrigatoriamente** usar a função do **item a**.

- c) `void troca_letras (string frase_sai, string frase_ent, int k);`

Esta função troca cada letra *i* da frase guardada em *frase\_ent* pela (i+k)-ésima letra do alfabeto (após a última letra, segue-se a primeira novamente). A nova frase obtida é armazenada em *frase\_sai*.

Por exemplo, a chamada

```
troca_letras (frase, "ESSEWBRWEXERCICIOWBRWEWBRWLEGALWPTW", 2);
```

coloca no vetor *frase* o string "GUUGYDTYGZGTEKEKQYDTYGYDTYNGICNYGZY".

d) `int procura_pal (string frase, int i, string pal)`

Esta função procura no string *frase*, a partir da posição *i*, a palavra guardada em *pal*. Esta função retorna -1, se a palavra *pal* não está no string *frase* a partir da posição *i*, ou o índice da primeira ocorrência da palavra.

Exemplos:

```
c = procura_pal ("FACILWINWWBRWWHFWWBRWSIMPLESWEXW", 0, "WINW"); c recebe valor 5.
```

```
c = procura_pal ("FACILWINWWBRWWHFWWBRWSIMPLESWEXW", 9, "WINW"); c recebe valor -1.
```

```
c = procura_pal ("FACILWINWWBRWWHFWWBRWSIMPLESWEXW", 13, "WBRW"); c recebe valor 17.
```

e) `void troca_string (string frase_sai, string frase_ent, string pal, char sinal)`

Percorre a frase guardada em *frase\_ent*, trocando toda ocorrência do string guardado em *pal* pelo sinal de pontuação guardado em *sinal*. A frase obtida é armazenada em *frase\_sai*.

Por exemplo, a chamada

```
troca_string (frase, "AH!WBRWAH!WBRWAH!", "WBRW", ' ');
```

coloca no vetor *frase* o string "AH! AH! AH!".

**OBS.:** Esta função deve **obrigatoriamente** usar a função do **item d**.

f) `void troca_todos_strings (string frase_sai, string frase_ent)`

Troca todos os strings da Tabela 1 que ocorrem na frase guardada em *frase\_ent* pelos respectivos sinais e coloca a frase obtida em *frase\_sai*.

Por exemplo, a chamada

```
troca_todos_strings (frase, "FACILWINWWBRWWHFWWBRWSIMPLESWEXW");
```

coloca no vetor *frase* o string "FACIL? - SIMPLES!".

**OBS.:** Esta função deve **obrigatoriamente** usar a função do **item e**.

## 4 O seu programa

O seu programa (`static void main (...)`) deve utilizar **obrigatoriamente** as funções dos itens b, c e f.

## 5 Desafio

Você seria capaz de decifrar o texto abaixo?

VJNSINZJKNSINVZENSINKVOKNSINZENSINUVLKJTYNGKNNSINVJNSINNRVIVNSIN  
QLNSINVZEWRTYNMINNSINNVEENSINUUVINSINKVOKNSINZENSINGFIKLXZVJZJTYN  
SINXVJTYIZVSVENSINNFIUVENSINNRVIVNGKNNSINZTYNSINNVZJJNGKNNSINZTY  
NSINNVZJJNMINNSINVJNSINZJKNSINEZTYKNSINWRZINGKNNSINYVIQCZTYVENS  
NXCLVTBNLVEJTYVENSINQLDNSINXVNZEEVINVONNVONNVON