Programação Estruturada

Folha de exercícios

Cifra "*Vigenére*"

Os métodos elementares de codificação de mensagens baseiam-se em substituir cada letra no texto original por outra para obter o texto cifrado. Uma função de substituição muito simples é a *rotação do alfabeto* de k posições. Por exemplo, a rotação de duas posições é a seguinte transformação:

$$\begin{pmatrix} A & B & C & \dots & Y & Z \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ C & D & E & \dots & A & B \end{pmatrix}$$

Uma rotação é completamente identificada pelo seu índice (o número k de posições que rodamos) ou pela letra em que enviamos a letra 'A'. Assim, a rotação acima é identificada por $k = 2$ ou pela letra 'C'. Note que há apenas 26 rotações diferentes e que a rotação de índice 0 é a identidade.

Com apenas 26 hipóteses diferentes não é muito difícil descodificar uma mensagem cifrada por uma rotação: bastaria testar os 26 casos exaustivamente até obter texto comprêensivel.

Podemos sofisticar a cifra usando uma *palavra-chave* para as rotações, obtendo a *cifra Vigenére*. Suponhamos que queremos cifrar o texto

"NAVIO CHEGA HOJE".

Começamos por combinar antecipadamente uma palavra-chave com o receptor da mensagem, por exemplo: LUAR. Eliminamos todos os espaços e pontuações do texto ficando apenas com letras de A a Z. Em seguida sobrepomos o texto a cifrar com a palavra-chave, repetida o número de vezes necessário; para cada letra do texto efectuamos a rotação definida pela letra da chave correspondente:

Texto:	N	A	V	I	O	C	H	E	G	A	H	O	J	E
Chave:	L	U	A	R	L	U	A	R	L	U	A	R	L	U
Cifrado:	Y	U	V	A	A	W	H	V	R	U	H	G	U	Y

O texto cifrado fica então

"YUVAAWHVRUHGUY".

Para descodificar o texto cifrado basta efectuar as *rotações com a chave inversa*, isto é a chave cujos índices são os "simétricos módulo 26":

Chave:	L	U	A	R
Índices:	11	20	0	17
Simétricos:	$26 - 11 = 15$	$26 - 20 = 6$	$26 - 0 = 26$	$26 - 17 = 9$
Chave inversa:	O	F	Z	I

1. Escreva uma função em C de codificação usando esta cifra; o protópo da função deverá ser:

```
void vigenere(char *key, char *plain, char *cypher);
```

onde

- `char *chave` é um apontador para a chave;
- `char *plain` é um apontador para o texto original;
- `char *cypher` é um apontador para local onde deve ficar o text cifrado (com o mesmo espaço que o comprimento do texto original).

Suponha que a chave e o texto original apenas contêm letras maiúsculas (carácteres de `A` a `Z`).

2. Escreva um programa em C que leia uma chave e um texto e escreva o texto codificado.

   Todos os caracteres de pontuação, espaços no texto original deverão ser suprimidos. As letras minúsculas deverão ser convertidas em maúsculas antes de efectuar a codificação.

   Exemplos: Se a entrada for

```
LUAR
Navio chega Hoje!
```

   então o resultado será:

```
YUVAAWHVRUHGUY
```

3. Escreva uma função para "inverter" uma chave módulo 26 letra-a-letra; o protótipo deverá ser

```
void invkey(char *key);
```

   A execução da função deve alterar a chave, transformando-a na sua inversa. Por exemplo se `char *key="LUAR"` então após a inversão deve ser `char *key="OFZI"`.

   Usando esta função escreva um programa de descodificação da cifra anterior. A entrada deve ser uma chave e um texto cifrado e o resultado deve ser o texto descodificado. Exemplo: se a entrada for

```
LUAR
YUVAAWHVRUHGUY
```

   então a saída deverá ser:

```
NAVIOCHEGAHOJE
```