Departamento de Ciência da Computação – IME-USP

MAC 115 – Introdução à Computação para Ciências Exatas e Tecnologia

BACHARELADO EM FÍSICA – PRIMEIRO SEMESTRE DE 1997

Terceiro Exercício-Programa — Data de entrega: até 10 de novembro de 1997.

UM POUQUINHO DE CRIPTOGRAFIA

O objetivo deste exercício-programa é decifrar um texto que foi cifrado por um dos métodos utilizados em criptografia.

Criptografia é a área da Ciência da Computação que estuda o ciframento e deciframento de informações com o objetivo de impedir o serviço de espionagem.

Um tipo de ciframento, chamado de **substituição monoalfabética**, consiste em simplesmente trocar uma letra do alfabeto por outra (de acordo com uma tabela dada), mantendo os demais símbolos (espaço em branco, sinais de pontuação, etc) intactos. As letras maiúsculas e minúsculas não são diferenciadas.

Exemplo:

letras do texto:

$$A\ B\ C\ D\ E\ F\ G\ H\ I\ J\ K\ L\ M\ N\ O\ P\ Q\ R\ S\ T\ U\ V\ W\ X\ Y\ Z$$

letras correspondentes do texto cifrado:

$$I\ S\ N\ L\ B\ Q\ C\ R\ D\ T\ P\ E\ Z\ A\ G\ U\ Y\ K\ O\ X\ J\ W\ H\ M\ V\ F$$

texto1: Que programa legal!
texto cifrado1: YJB UKGCKIZI EBCIE!

texto cifrado2: WGJ CGOXIK LB QIFBK BOXB BMBKNDNDG-UKGCKIZI.
texto2: ?

Note que existem 26! (aproximadamente 4×10^{26}) ciframentos possíveis.

Faça um programa em C que tenha como entrada um texto cifrado e ajude o usuário no processo de deciframento, assumindo que a cifra utilizada foi a substituição monoalfabética.

A **técnica de deciframento** que você deve utilizar é baseada no fato de que as letras do alfabeto não ocorrem todas com a mesma freqüência. Nos textos em português, a letra 'a' é a mais comum, 'e' é a segunda mais comum, e assim por diante, até 'k', 'w' e 'y' que quase não ocorrem. Então, como um primeiro passo para decifrar o texto, você deve contar a ocorrência de cada letra no texto cifrado e tentar relacionar essas freqüências com a freqüência padrão conhecida das letras em português. Por exemplo, se 'q' foi a letra que mais ocorreu no texto cifrado, é razoável desconfiar que 'q' está substituindo no texto cifrado a letra 'a' do texto original. Logo, você trocará todas as ocorrências da letra 'q' no texto cifrado pela letra 'a' e as ocorrências da letra 'a' pela letra 'q'. Analogamente, você pode repetir esse procedimento para todas as 26 letras do alfabeto.

A seguir vamos descrever os passos mais importantes deste programa.

1. A ordem de freqüência padrão de ocorrência das letras em português é a seguinte:

   $$A\ E\ O\ S\ I\ R\ N\ D\ T\ C\ M\ U\ P\ L\ V\ G\ B\ F\ Q\ H\ J\ X\ Z\ K\ W\ Y$$

   No seu programa, você pode armazenar essa informação em um vetor de caracteres, digamos o vetor `LetrasOrdemFreqPortug[26]`.

2. O seu programa deve ler o texto cifrado de um arquivo, armazená-lo em um vetor de caracteres, digamos `Texto[MAX]` (onde `MAX` é uma constante, por exemplo, 300) e imprimi-lo.

3. Como o exercício-programa anterior, o seu programa deve contar o número de vezes que cada letra do alfabeto aparece no texto cifrado e ordenar as letras do alfabeto de acordo com o número de vezes que estas ocorrem no texto cifrado (em ordem não-crescente). Digamos que o resultado de tal ordenação fique guardado no vetor `LetrasOrdemFreqCifrado[26]`.

4. Como primeira tentativa de deciframento, o seu programa deve trocar no texto cifrado a ocorrência da letra mais frequente pela letra mais freqüente em português, a segunda letra mais frequente pela segunda letra mais freqüente em português, e assim por diante. Ou seja, a letra `LetrasOrdemFreqCifrado[i]` deve ser trocada pela letra `LetrasOrdemFreqPortug[i]`, para cada i, $0 \leq i < 26$. Tente fazer essa troca de um jeito eficiente, percorrendo o texto apenas uma vez, e não uma vez para cada uma das letras.

5. Em geral o texto obtido (armazenado, digamos, no vetor de caracteres `TextoDecifrado[MAX]`) depois desta primeira tentativa de deciframento não é a mensagem correta decifrada. Algumas letras não terão sido trocadas pelas letras corretas. O programa entra então em uma segunda fase, onde se inicia uma interação com o usuário. Repetidamente, o texto corrente armazenado no vetor `TextoDecifrado[MAX]` é impresso na tela e o programa pergunta ao usuário se este gostaria de trocar algumas letras. Em caso afirmativo, o programa pede ao usuário que digite um par de letras L1 e L2 e troca no texto corrente L1 por L2 e vice-versa. O texto corrente depois da troca é mostrado e o programa volta a perguntar se o usuário quer trocar mais letras.

6. Quando finalmente o usuário responde que não quer trocar mais letras, o texto corrente é armazenado em um arquivo de saída (usando a receita dada abaixo) e o programa termina a execução.

<u>Exemplo</u>:

texto cifrado: BOXI B JZI ZBAOICBZ NDQKILI.

```
FreqLetrasCifrado = [1 4 1 1 0 0 0 0 5 1 1 1 0 1 2 0 1 0 0 0 0 0 0 1 0 3]
LetrasOrdemFreqCifrado = [I B Z O D X A K Q N C J L E F G H M P R S T U V W Y]
```

primeira tentativa
de deciframento: ESRA E UOA OENSAMEO CITDAPA.

Como esperávamos, a mensagem não foi decifrada corretamente na primeira tentativa. Suponha que o usuário entre com o par O M:

tentativa
de deciframento: ESRA E UMA MENSAOEM CITDAPA.

Em seguida, com o par R T:

tentativa
de deciframento: ESTA E UMA MENSAOEM CIRDAPA.

Assumindo que o usuário entre com os pares O G, depois D R, seguido por D F, e finalmente P D, temos:

tentativa
de deciframento: ESTA E UMA MENSAGEM CIFRADA.

Saída final: **texto cifrado:** BOXI B JZI ZBAOICBZ NDQKILI.

texto decifrado: ESTA E UMA MENSAGEM CIFRADA.

Gravando a saída em um arquivo

```
#include <stdio.h>

main()
{
   FILE *Entrada,   /* Apontador para o arquivo de entrada. */
        *Saida;     /* Apontador para o arquivo de saída.   */
   char Arquivo_de_Entrada[40],
        Arquivo_de_Saida[40];

   printf("Entre com o nome do arquivo contendo o texto cifrado: ");
   scanf("%s", Arquivo_de_Entrada);

   if ((Entrada = fopen(Arquivo_de_Entrada, "r")) == NULL)
        printf("Erro: não pode abrir o arquivo de entrada.\n");

   ...

   printf("Entre com o nome do arquivo de saida: ");
   scanf("%s", Arquivo_de_Saida);

   Saida = fopen(Arquivo_de_Saida, "w");

   ...
        fprintf(Saida, "%c", TextoDecifrado[i]);
   ...

   fclose(Entrada);
   fclose(Saida);
}
```