

**MAC0121 – Algoritmos e Estruturas de Dados I****Segundo semestre de 2020****Criptografando textos – Entrega: 25 de setembro de 2020**

O objetivo deste exercício-programa (EP) é familiarizar os alunos com o uso da linguagem de programação C. Vocês irão utilizar caracteres, vetores e leitura de arquivos. O programa poderá ser usado para criptografar um texto. O esquema criptográfico utilizado está descrito abaixo.

**1 Descrição do EP**

Neste EP vamos implementar um esquema de criptografia conhecido desde os tempos romanos, chamada de “Cifra de Cesar”. Conta a lenda que era o método usado por Julio Cesar, imperador romano de 49 a.C. a 44 a.C., para se comunicar com seus generais.

A ideia da cifra é muito simples. O imperador combinava com seus generais um número e então substituíam cada letra pela letra deslocada do número combinado. Assim, se o número combinado era 3, a letra 'A' seria substituída por 'D', 'B' por 'E' e assim por diante, até que 'X' é substituída por 'A', 'Y' por 'B' e 'Z' por 'C'. Para tornar as coisas um pouco mais divertidas :) vamos complicar a cifra :))

É dado um texto em um arquivo composto por letras maiúsculas, minúsculas (sem acentos), espaços em branco, ', ', '.', '!', '?', e '\n', terminado pelo caractere de fim de arquivo 'EOF'. O primeiro passo é transformar os caracteres especiais acima em letras da seguinte forma:

- brancos: são removidos
- , (vírgula): “vr”
- . (ponto): “pt”
- : (dois pontos): “dp”
- ! (exclamação): “ex”
- ? (interrogação): “in”
- nova linha: “nl”

Assim, se a frase de entrada foi

Para um bom entendedor, meia palavra basta.

após esta primeira fase a frase ficaria:

Paraumbomentendedorvrmeiapalavrabastapt

Depois deste primeiro passo, é aplicada a rotação no alfabeto, só que usamos dois números: um para deslocar as letras maiúsculas e outro para as letras minúsculas. Assim, se combinamos  $d_1 = 3$  para as maiúsculas e  $d_2 = 5$  para as minúsculas, a frase acima seria criptografada como:

Sfwfzrgtrjsyjsijitwawrjnfufqfawfgfxyfuy

Para o texto<sup>1</sup>,

Em horas inda louras, lindas  
Clorindas e Belindas, brandas  
Brincam nos tempos das Berlindas  
As vindas vendo das varandas.

Usando  $d_1 = 10$  e  $d_2 = 7$ , (imprimindo em 80 colunas)

OtovyhzpukhsvbyhzcypukhzusMsvypukhzlLlspukhzczyihukhzusLypujhtuvzaltwvzkhzLlysp  
ukhzusKzcpukhzcukvkzhzchyhukhzwa

## 2 Como ler o texto de um arquivo?

Seu EP deverá ler da entrada os inteiros  $d_1$  e  $d_2$  correspondentes aos deslocamentos desejados e o nome de um arquivo de texto onde estará o texto a ser criptografado.

```
char nome[80]; /* recebe o nome externo do arquivo a ser lido */
```

```
printf ("Digite o nome do arquivo que será criptografado: ");  
scanf ("%s", nome);
```

Com isso, você poderá abrir o arquivo para leitura:

```
FILE * arq; /* variável que recebe um ponteiro para o arquivo que será lido */
```

```
arq = fopen (nome, "r"); /* abre o arquivo com o nome lido para leitura "r" */  
if (arq == NULL) {  
    printf ("Não encontrei o arquivo %s\n", nome);  
}
```

Feito isso você poderá ler normalmente (se o arquivo foi aberto corretamente), agora do arquivo e não mais da entrada padrão:

```
tam = 0;  
while (!feof (arq)) {  
    fscanf (arq, "%c", &texto[tam]);  
    tam++;  
}
```

A função `feof` verifica se o arquivo acabou (*end of file*). A função `fscanf` é semelhante ao `scanf`, mas lê de um arquivo, e não da entrada padrão (teclado).

**Importante:** Não é permitido neste EP o uso das funções de biblioteca de strings. Os textos devem ser considerados como vetores de caracteres.

---

<sup>1</sup>Fernando Pessoa, em “Cancioneiro”

### 3 Desafio

Este esquema de criptografia é muito simples, e especialistas não recomendam seu uso para aplicações reais. Abaixo seguem três textos criptografados usando o esquema (não vou contar os valores usados para os deslocamentos ;). Como verão, não é difícil descobrir o texto original.

- Texto 1:

AcivlwviakqdzcuivrwbzbwvtlmaamaycmdqdmuviawujzivtlqaamlxFiqdzMiztwamfamzoickpmv  
idqlixbvtvtKakiaiaamaxqiuwapwumvavtycmkwzzmuibzialmuctpmzmaxbvtKbizlmbitdmhnwaami  
hctdzvtviwpwcdmaambivbwalmamrwaxbvtvtYjwvlmxiaaikpmqwlmxmzvialxvtxmzviajzivkiaxz  
mbiaiuizmtiaxbvtZiziycmbivbixmzvidzumNmcadzvtxmzocvbiumckwzikiwxbvtZwzmuumcawtp  
wvvtviwxmzocvbiuvilixbvtvtYpwumuibzialwjowlmvtmamzqwdzaquxtmamnwzmbxbvtAciamviw  
kwvdmzaixbvtDmuxwckwadzzizwaiuqowavtwpwumuibzialwawktwamlwjowlmxbvtvtWmcNmcadz  
xwzycmumijivlwviabmvtamaijqaiaycmmcviumziNmcavtamaijqaiaycmmcmzinzikwxbvtvtWcvlwuc  
vlwdiabwucvlwdzvtammcumkpiuiaamBiqucvlwvtamzqicuzquidzviwamzqicuiawtckiwxbvtWcv  
lwucvlwdiabwucvlwdztuiqadiabwmumckwzikiwxbvtvtOcvlwlmdqibmlqhmzvtuiamaaitcivtui  
amaamkwvpiycmvtjwbiuiomvbmkuwdqlkwuwllqijwxbvtvt

- Texto 2:

RmtixngePhnixsexlmhftmxlixexxlxgjn tkmbxklokgeRmtixwxnqgeexltnuxkzbglxmexlvhnkz  
xmmxlxgkhgwxexlimgRmtixmkhblgeRfbgvsexlihbokhglxgetfxexlgeRmtixjntmkxgexmexh  
bzghgxgkhnxexlimgRmtixvbgjgePatnyyxswxnqvnbeexkxltlnixwxanbexwtglnxihxexgeRm  
tixlbqgexmytbmxlryhgwxexlhbzghglxmexlihbokhglimgRmtixlximgYhkljnxbellhgmmxgwk  
xloktchnmxsexlmhftmxlokextbeatvaxokexmarfxmexetnkbxkimgeRmtixanbmgeFtexsokihbokx  
sxmetbllxsfbcmxkwhnvxfxgmtvhnokmwntgmjntkxgmvgjfbgnmxlimgRmtixgxnygeCxgwtgm  
vxmxfilokikxixkxsexltnuxkzbglxmexlvhnkzmmxlmStbmxlexlvnbkxlxitkxffxgmhnghtg  
lexanbexwxheboxixgwtgmjnbgsxfbgnmxlimgRmtixwbqgeIxxbybxsetvnbllhgwlexznfxlihnk  
jnxbelgxlhbmgmienlyxkfxlimNchnmxsexltehkltnfxetgzxwxmhftmxlxmikhehgzxsetvnbllhgl  
nkmhnmixmbmyxnxgwtgmwbqfbgimgRmtixhgsxgeFtexsxmihbokxslbuxlhbimg

- Texto 3:

EnjEzijsSjifpyntsawinjzwxuwzjslqnhmnsnmwjwUjcyxfrqzslinjXtjwyjwHwzsixyzjhpaxjwp  
jmwxljsjmrnlzslxezxyfjsinlpjnyxzjgwywflzslxajwtwiszslzsiSnsikqjnxhmjynpjyynjwzs  
lxzjgjbfbhmzslxfzklfgjszjgwywflzslxljxjyefqxqfjslxyjxzsiebjnyqfjslxyjxXtwynijsy  
nknenjwymfyawsjssyfsiwnyyjwTyjqjifxXtwyWjwpjmwxnkskwfxywzpyzwnksfsenjwzslxljxjq  
xhmkyawfgjwifinjCjkzlsnxxjinjxjwHjxjqxhmfkynrxnjgjsCzhmxyfgjsqfjsljwjsWjwpjmw  
nkskwfxywzpyzwnksfsenjwzslxljxjqxhmfkylxjyjelwjljqyxnsiawl jmtjwyifxXtwygjxyjsk  
fqqxfsinjanjwyjTyjqjuyUfyxfjhmqnmbnwifgjwfzhmifxWjwpjmwxnkskwfxywzpyzwnksfsenjw  
zslxljxjqxhmfkylxjyjesthmizwhminjWjwrtjljsxeztwiszslxezxyfjsinlpjnyxzjgwywflz  
slxajwtwiszslxljxhmqljsuyNnyxjhmzsikzjskenlCzhmxyfgjsnxyinjxjxXtwynrNtrjsyijwfxz  
xnhmyxwjnhmxyjLfsinifykzjwifxqfjslxyjsnhytgxtqjyXtwyijxEjzyxhmjsawifxfzxxhmqnj  
xxqnhmryCzhmxyfgjsljxhmwnjgjsbnwi(ifezlqjnhmrjmw)uysq