Criptografia - Transposição Linha x Coluna
Professor: Areolino de Almeida Neto
Curso: Ciência da Computação
Aluno: Lucas de Macedo Terças - 2015042898

Objetivo: O objetivo deste trabalho é a implementação do algoritmo de cifragem e de decifragem da cifra de transposição linha por coluna. Nesta cifra, primeiro transforma-se o texto claro em uma matriz, e então a mensagem é construída por colunas usando-se a ordem alfabética da chave.

Exemplo de Cifragem

Texto Claro: lorem ipsum dolor sit amet
Chave: lucasde (7 caracteres)

Primeiro Passo: Transcrever o texto claro para forma de matriz, sendo o número de colunas da matriz o número de caracteres da chave, e se a última linha não tiver caracteres suficientes, usar letras aleatórias. (a primeira coluna é a chave, porém ela não realmente faz parte da matriz).

l	**u**	**c**	**a**	**s**	**d**	**e**
l	o	r	e	m	i	p
s	u	m	d	o	l	o
r	s	i	t	a	m	e
t	x	x	x	x	x	x

Segundo Passo: Transpor a matriz, porém, a ordem das linhas da matriz resultante deve ser a ordem alfabética da chave.

a	e	d	t	x
c	r	m	i	x
d	i	l	m	x
e	p	o	e	x
l	l	s	r	t
s	m	o	a	x
u	o	u	s	x

Terceiro Passo: Construir o texto cifrado, concatenando as linhas da matriz:
edtx rmix ilmx poex lsrt moax ousx

Exemplo de Decifragem

Texto Cifrado: edtx rmix ilmx poex lsrt moax ousx

Chave: lucasde

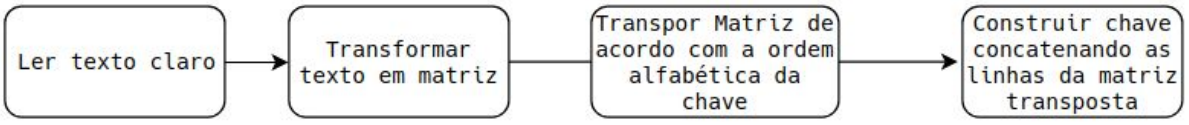
Primeiro Passo: A decifragem também necessita transformar o texto cifrado em matriz, porém, o número de colunas da matriz é o tamanho do texto dividido pelo tamanho da chave.

a	e	d	t	x
c	r	m	i	x
d	i	l	m	x
e	p	o	e	x
l	l	s	r	t
s	m	o	a	x
u	o	u	s	x

Segundo Passo: Construir o texto decifrado, lendo as colunas pela ordem da chave original, por exemplo, a primeira coluna seria: lucasde -> loremip.

Explicação do Programa por Diagrama de Blocos

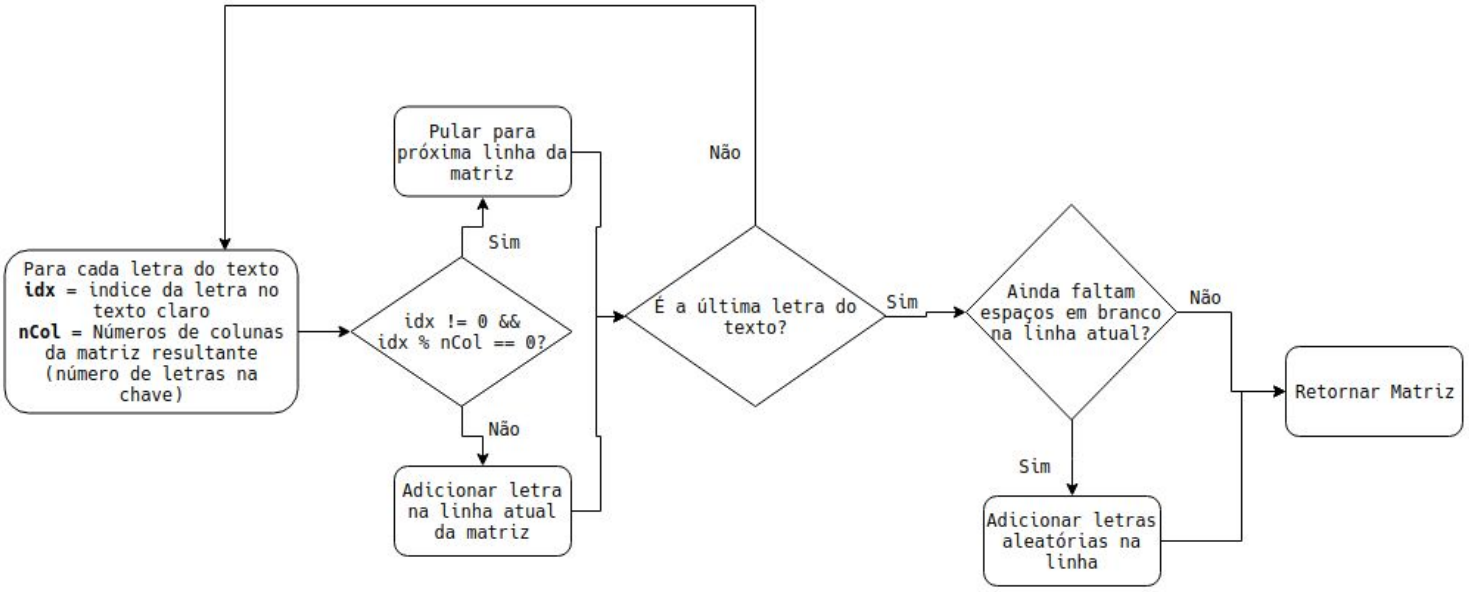
Diagrama ilustrando a visão geral do programa, para uma segurança, o programa repete os passos 2 ao 4 três vezes, construindo a chave a partir da matriz, e realimentando essa chave para o passo 2.



Esse diagrama mostra o processo de transformar o texto claro em matriz:

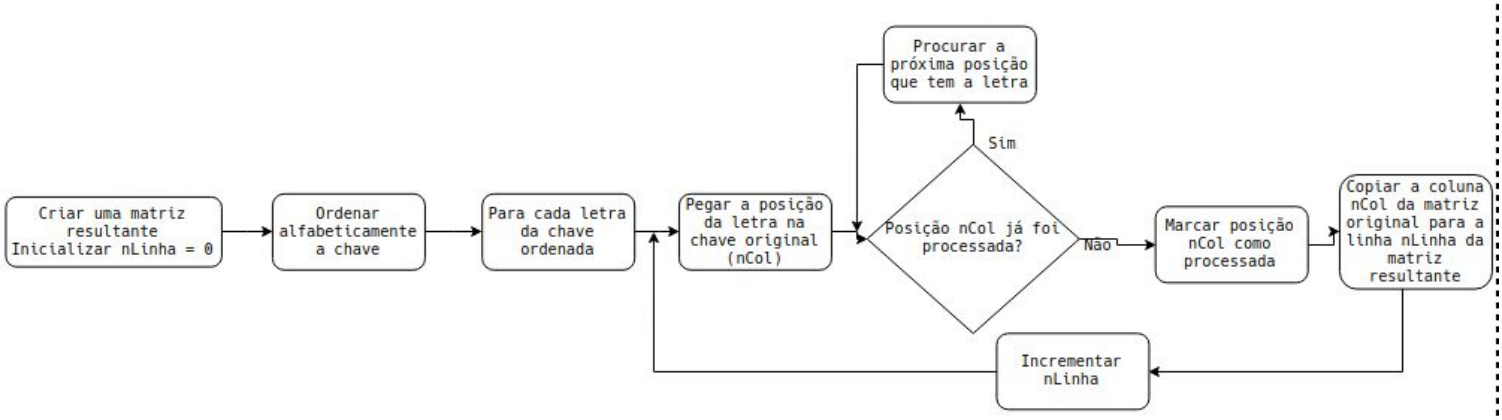
Para cada letra do texto claro, adicionar a letra na linha atual da matriz, e se o índice da letra módulo o número de letras na chave for 0, pular para a próxima linha da matriz, assim se assegurando que cada linha tenha o mesmo número de caracteres que a chave.

Se for a última letra do texto, e se ainda tiver espaços faltando na linha atual, adicionar letras até na linha, e então retornar a matriz.



Esse diagrama mostra o processo de transpor a matriz do passo anterior, seguindo a ordem alfabética da chave:

Primeiro se ordena alfabeticamente a chave, depois para cada letra dessa chave ordenada, salva onde essa letra se encontra na chave normal, e copia a coluna com essa posição para a matriz resultante, registrando em algum lugar (no código, usei um vetor de booleanos) se essa posição já foi processada. Isso é necessário por que a chave pode ter letras repetidas.



Funcionamento do Programa

Para compilar o programa, navegue até a pasta onde se encontra ele por terminal, e execute o comando **make**, isso irá criar um executável chamado transposicao.

Para executar, execute o arquivo executável criado (./transposicao). O programa irá lhe pedir a chave, e o local do texto claro, e então irá botar o texto cifrado em textos/texto_cifrado.txt e o decifrado em textos/texto_decifrado.txt

```
# lucastercas @ hyperion in ~/workspace/ufma/criptografia/transposicao on git:master x [21:43:23]
$ make
g++    -c -o main.o main.cpp
g++    -c -o util.o util.cpp
g++    -c -o decifrar.o decifrar.cpp
g++    -c -o cifrar.o cifrar.cpp
g++ -o transposicao main.o util.o decifrar.o cifrar.o

# lucastercas @ hyperion in ~/workspace/ufma/criptografia/transposicao on git:master x [21:43:33]
$ ./transposicao
=== Cifra de Transposição - Linha x Coluna ===
Digite a chave: lucasde
Digite o local do texto claro: textos/texto_claro.txt

=> Stage 1: edtxrmixilmxpoexlsrtmoaxousx

=> Stage 2: xmsotllxmptsiomxexeorxrudixa

=> Stage 3: otxdspeulioxloraxxmxtseimmxr
Texto cifrado guardado em textos/texto_cifrado.txt

=> Stage 1: xmsotllxmptsiomxexeorxrudixa

=> Stage 2: edtxrmixilmxpoexlsrtmoaxousx

=> Stage 3: loremipsumdolorsitametxxxxxx
Texto decifrado guardado em textos/texto_decifrado.txt
```