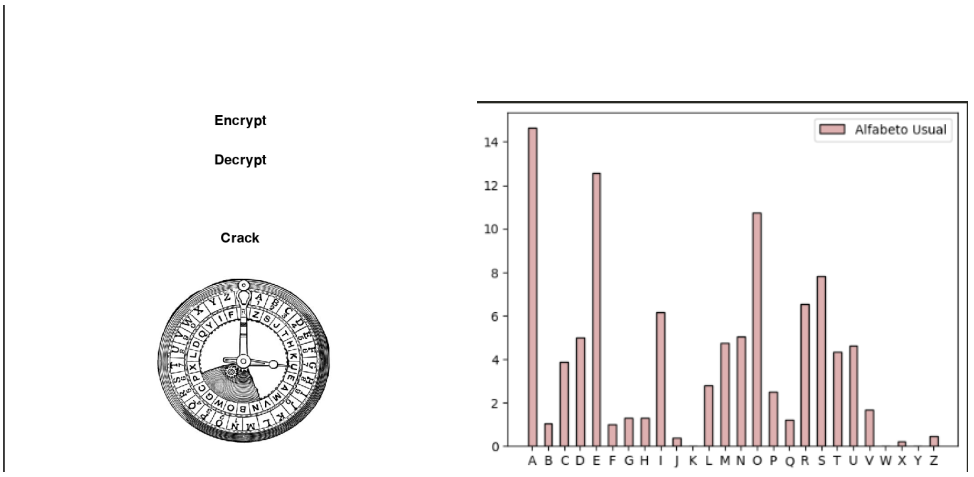
Relatório

Vinícius Lima Passos
Gustavo Rodrigues Gualberto

==== Geral, execução

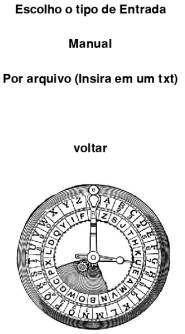
Todo o código encontra-se em main.py, basta executá-lo na linha de comando que o programa será iniciado. Na pasta, há a presença de arquivos de texto de saída e entrada, para os quais é possível utilizar para testes, denotando-se “QuebraEngEx.txt” e “QuebraPortEx.txt” para textos cifrados disponíveis para quebra.
A navegação pelo programa ocorre majoritariamente pelo mouse, com entradas no teclado em caso de input.
Para plotagem de frequências e interfaces gráficas foram utilizadas as bibliotecas pyplot e o pygame.

Para a entrada de texto, há o processo de normalização que transforma todas as letras em maiúsculas, retira pontuação e acentuação e transforma letras como o ç de volta em c.
A justificativa dessas medidas decorre de que o método de quebra não utiliza a frequência de pontuação, além disso não há a presença da frequência do ‘ç’ ou de letras individuais acentuadas.

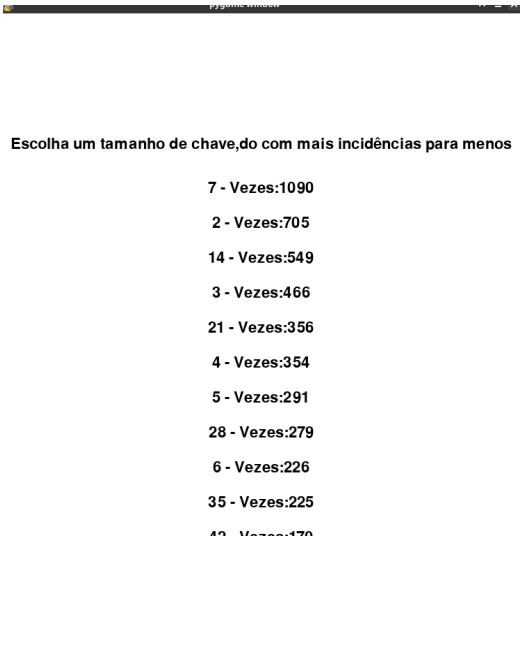


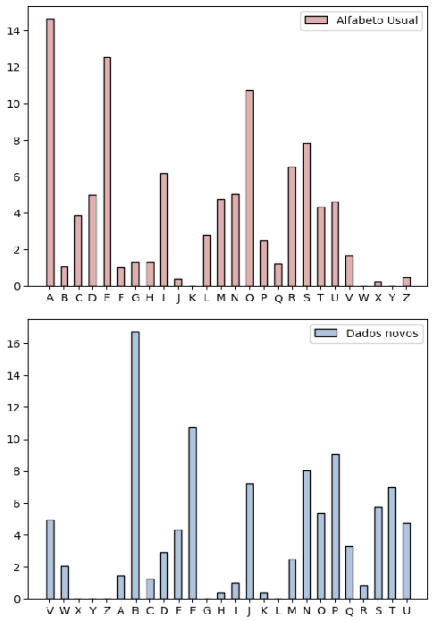
Ao iniciar o programa é possível escolher dentre 3 opções, cifrar ou decifrar com base em uma chave ou quebrar um texto cifrado.

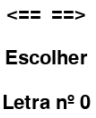
Após a seleção, se escolhe se a entrada do texto/chave será manual ou por arquivos.
No caso manual, digita-se a o texto/chave.
Em caso de arquivos, digita-se o nome dos arquivos de texto a serem extraídos pelo programa.
A saída em todos os casos é escrita em arquivos informados pelo programa.



Nos casos de cifrar e decifrar com chave, o processo é trivial, são informadas as entradas e o programa gera a saída em arquivos, como mencionado.







Para a quebra de cifra, após a entrada do texto, o usuário escolhe a língua e em seguida o possível tamanho da chave, ordenado de acordo com as devidas incidências. Após isso, o usuário pode transladar o gráfico para as letras da cifra, comparando e por fim as escolhendo, produzindo uma chave que também é guardada em um arquivo.

=====Descrição das funções criptográficas/relacionadas e como funcionam

Parte 1 - cifragem
Com poucas funções a parte de cifragem apresenta uma função de normalização de texto *normalize*, a qual deixa o texto de acordo com as especificações definidas na introdução. Em seguida há a função *keygen* responsável por alongar a chave até o tamanho do texto caso esse seja maior.
Por fim, há as funções de encriptação e decriptação, as quais passam letra por letra, somando com a chave e retirando os devidos offsets
==========
Parte 2 - quebra

O processo de quebra, por ser mais envolvido, apresenta mais funções:

A função encontrarEspacosTriosRepetidos(texto_cifra) recebe o texto cifrado e retorna todos os trios de caracteres repetidos e seus respectivos coeficientes de espaço. Para isso, utiliza-se laços aninhados, para melhor organização da lista, que funciona como uma lista de “pseudotuplas”, as quais registram o trio seguido de seu coeficiente de espaço entre dois trios. Cada laço aninhado percorre singularmente por cada lista por meio de seu comprimento.

A função numerosFatorados(num) tem como objetivo retornar todos os divisores de cada coeficiente de espaço entre dois trios. Para isso, faz-se um laço que acrescenta cada divisor que não retorna resto em uma lista e remove-se o 1 por que uma chave não pode ter comprimento 1.

A função triosMaisRepetidos(seqFactors) verifica quais trios são mais repetidos no texto cifrado. Para isso, cria-se uma lista de frequência de divisores a fim de criar uma lista ordenada decrescentemente das frequências obtidas.

A função encontraTamanhosProvaveis(ciphertext) utiliza as funções acima a fim de retornar quais tamanhos da chave são mais prováveis. Para isso, a função chama as funções acima e as associam para assim, retornar uma lista de quais tamanhos são mais prováveis em ordem crescente de prioridade.

Por fim, a função achar_frequencias recebe um tamanho de chave e o texto cifrado para calcular as frequências relativas das substrings geradas pela chave desse determinado comprimento, realizando uma contagem e retornando uma lista de listas de frequências para cada letra da chave.

Esses valroes e dados são todos repassados para o pyplot e para o pygame, os quais executam o processo de apresentar esses dados, as funções que compreendem a classe Apresentação são apenas para o propósito de apresentar o programar, logo, dispensam análise, denota-se apenas a estrutura de *exec* , que mostra o fluxo geral do programa, o qual é possível seguir e analisar as funções.