

PRIMEIRO TRABALHO PRÁTICO – PROGRAMAÇÃO DE COMPUTADORES

Professor: Renato Dourado Maia

Curso: Engenharia de Alimentos

Turma: Terceiro Período

1 – Objetivos

Este trabalho tem como objetivos o treinamento da programação estruturada utilizando funções, *strings*, funcionalidades para trabalhar com arquivos em modo texto e dicionários, além do estudo dos demais recursos estudados na disciplina.

2 – Detalhamento

2.1 – Um algoritmo de criptografia simples é o *rot13*, que “gira” cada caractere 13 posições no alfabeto. Assim, ‘a’ é substituído por ‘n’ e ‘x’ é trocado por ‘k’, tal como apresentado na Figura 1, sendo que outros caracteres (pontos, caracteres acentuados, etc) não sofrem nenhuma alteração. O *rot13* é um exemplo de criptografia simétrica por chave. Com a criptografia simétrica por chave, tanto o encriptador como o decriptador utilizam a mesma chave.

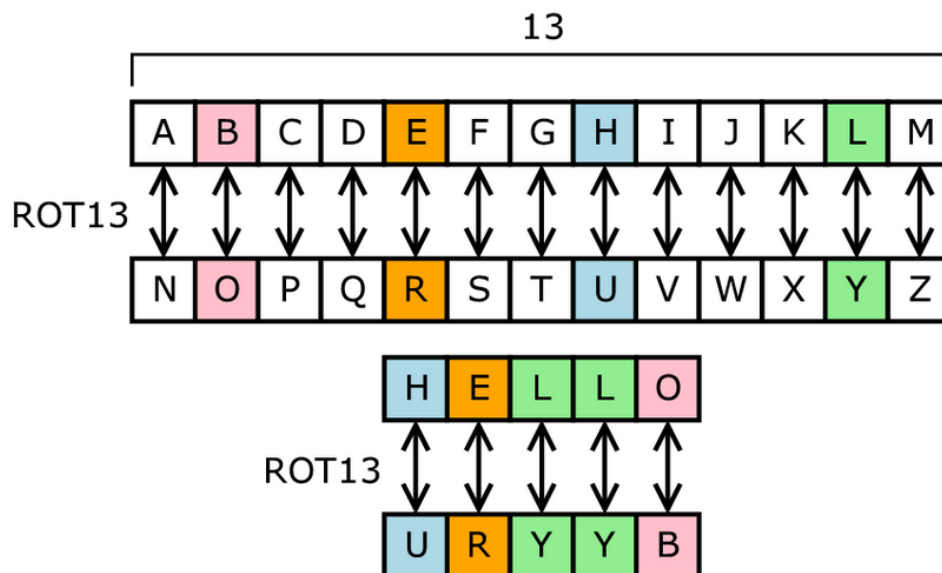


Figura 1. Algoritmo de criptografia *rot13*.

- (a) Escreva uma função que receba como parâmetro o nome de um arquivo, abra-o, leia-o e realize a encriptação/desencriptação utilizando o *rot13*. O arquivo deverá ser gravado em disco, com o mesmo nome do arquivo original, porém com a extensão **.crp**. A função deverá receber um outro parâmetro (lógico) cujo valor *default* deve ser **True**, que determinará se os conteúdos do arquivo lido, bem como o resultado da encriptação, deverão ser apre-

sentados na tela.

- (b) Escreva um programa que teste a função implementada em (a). Esse programa deverá ser implementado em um arquivo diferente daquele em que foi implementada a função da letra (a).

Links Interessantes:

- (a) <http://hex.com.br/rot13/>
(b) <http://www.dantas.com/rot13/>
(c) <http://www.rot47.com/>
(d) <http://www.rot13.com/>

Observações:

A função a ser implementada deverá resolver o problema das seguintes maneiras, sendo que cada uma deve ser implementada em uma função diferente:

- ✓ Utilizando um dicionário.
- ✓ Utilizando uma lista para guardar as letras do alfabeto: cada letra será um elemento da lista.
- ✓ Utilizando uma *string* para guardar as letras do alfabeto.
- ✓ Utilizando duas listas para guardar as letras do alfabeto: cada letra será um elemento da lista.
- ✓ Utilizando duas *strings* para guardar as letras do alfabeto.
- ✓ Utilizando a função *maketrans* do módulo *string* (não utilize essa função nas outras soluções).

Naturalmente, o programa de teste da letra (b) deverá testar todas as funções implementadas.

3 – Algumas Observações

3.1 – O trabalho deve ser feito **em equipes de, no máximo, três acadêmicos**: a discussão dos problemas e das estratégias de solução com os colegas de outras equipes é **permitida e aconselhável**. Todavia, a implementação e a documentação das soluções deve ser feita **apenas pelos membros da equipe**.

3.2 – Caso você utilize quaisquer **fontes externas** para elaborar as suas respostas, elas devem ser **citadas**: artigos ou livros, amigos ou colegas, informações encontradas na Internet, **qualquer coisa encontrada em qualquer lugar!** É melhor tentar solucionar os problemas, pois **solucionar problemas é um componente fundamental** para a nossa área de estudo. Não haverá penalidades no caso de utilização de ajuda externa, desde que devidamente citada e desde que essa ajuda **não seja a cópia** do trabalho de um colega. **Utilizar o trabalho dos outros, como se fosse seu, é plágio, é desonestidade acadêmica.**

3.3 – Os programas devem utilizar **apenas as funcionalidades originais da linguagem Python.**

3.4 – Uma **documentação escrita** deve ser elaborada, explicando os **objetivos** do trabalho e apresentando **resultados** e **discussões**. Naturalmente, a documentação deverá apresentar uma **conclusão**, argumentando sobre as **dificuldades**, sobre o **aprendizado** e sobre a **relevância** do trabalho no contexto do **curso** e da **disciplina**, e apontando eventuais **sugestões** para melhorias em futuras

“edições”. **Todas as decisões de implementação deverão ser explicadas e comentadas na documentação.** A documentação deve apontar **qual versão da linguagem Python** foi utilizada.

3.5 – Os programas devem ser **bem comentados** e os **nomes** de variáveis e funções devem ser **consistentes e informativos**.

3.6 – Quaisquer **suposições** feitas por você(s) (que não estiverem incluídas na descrição original do trabalho) devem também ser bem **documentadas**.

4 – Data de Entrega

Este trabalho **deverá ser entregue até o dia 09/06/2014 por e-mail**: deverá ser enviado **um único arquivo compactado** para o endereço pc@renatomaia.net, com o assunto **PC – TP I**, contendo os códigos e um relatório, que deve ser enviado no formato do editor de textos utilizado por você(s) e também no formato pdf. **O arquivo compactado deve ter o seguinte nome: PC_TPI_Nome1Nome2Nome3 (PC_TPI_RenatoMaiaAndreMendesJoaoSilva, por exemplo, sem espaços e/ou caracteres especiais, utilizando o primeiro nome e o último sobrenome).**

5 – Valor

Este trabalho vale 15 pontos.