

Desafio 2

Desafio: Interceptamos uma mensagem porém ela está criptografada. Entretanto, sabemos que a mensagem foi encriptada da seguinte forma: "cada letra torna-se sua correspondente na ordem do alfabeto eg: a=1, b=2 e então cada letra passa pela seguinte função $f(x) = x^2 + 2x$ ". Você deve decriptar utilizando um código que não utilize força bruta pois necessitamos de velocidade na hora de decriptar as próximas mensagens. Você consegue fazer um decoder?

Resolução do Desafio:

- 1) Foi recebido uma mensagem criptografada com diversos números contidas no arquivo, como pode-se observar na imagem abaixo.



| Open | | encrypted-text | | Save | | | |
|------|-----|----------------|-----|------|-----|-----|-----|
| | | ~/Downloads | | | | | |
| 1 | 3 | 360 | 3 | 728 | 255 | 80 | 483 |
| 195 | 3 | 224 | 3 | 224 | 483 | 195 | 24 |
| 35 | 440 | 35 | 360 | 195 | 99 | 224 | 3 |
| 24 | 255 | 24 | 255 | 195 | 224 | 99 | 255 |
| 24 | 255 | 399 | 399 | 35 | 483 | | |
| 2 | 15 | 255 | 224 | 80 | 35 | 15 | 99 |
| 195 | 35 | 528 | 35 | 360 | 3 | 440 | 255 |
| 360 | 195 | 35 | 224 | 440 | 3 | 24 | 3 |
| 3 | 288 | 255 | 360 | 323 | 483 | 35 | 399 |
| 440 | 35 | 399 | 440 | 35 | 399 | 323 | 483 |
| 224 | 255 | 288 | 255 | 24 | 35 | 35 | 528 |
| 99 | 440 | 3 | 360 | 288 | 255 | 99 | 399 |
| 168 | 80 | 35 | 399 | 255 | 99 | | |
| 4 | 224 | 3 | 440 | 483 | 360 | 35 | 728 |
| 3 | 195 | 3 | 399 | 399 | 323 | 483 | 3 |
| 99 | 399 | 440 | 3 | 195 | 8 | 195 | 224 |
| 255 | 288 | 255 | 24 | 35 | 24 | 3 | 360 |
| 360 | 35 | 399 | 288 | 255 | 399 | 99 | 8 |
| 5 | 483 | 168 | 440 | 360 | 3 | 288 | 3 |
| 399 | 399 | 3 | 360 | 35 | 195 | 15 | 255 |
| 195 | 288 | 168 | 35 | 440 | 3 | 195 | 35 |
| 224 | 440 | 35 | 3 | 399 | 399 | 483 | 3 |
| 288 | 255 | 399 | 399 | 99 | 8 | | |

- 2) Sabendo que para descriptografar é necessário entender os seguintes dizeres:
"cada letra torna-se sua correspondente na ordem do alfabeto eg: a=1, b=2 e então cada letra passa pela seguinte função $f(x) = x^2 + 2x$ ".
- 3) Para melhor entender este problema, irei dar um exemplo de uma resolução para esses dizeres fornecido acima.
Sabendo que precisamos encontrar o resultado de **x** da função **$f(x) = x^2 + 2x$** , e esse **x** nunca será negativo pois foi informado que cada letra torna-se sua correspondente na ordem do alfabeto, ou seja, seguindo o alfabeto de **a até z**, o número de **x** será de **1 até 26**, e utilizando um dos números presente no arquivo '**encrypted-text**' podemos chegar a seguinte equação:
 $f(x) = x^2 + 2x = 360$
- 4) Para resolver esta equação descobrindo o valor de **x**, podemos utilizar a fórmula de Bhaskara.

$$\Delta = b^2 - 4.a.c$$

$$\Delta = (-2)^2 - 4.1.-360$$

$$\Delta = 4 + 1440$$

$$\Delta = 1444$$

$$(-b + \sqrt{\Delta})/2.a \Rightarrow (-2 + \sqrt{1444})/2.1 \Rightarrow (-2 + 38)/2 \Rightarrow x = 36/2 \Rightarrow x=18$$

- 5) Depois de entender o conceito, foi realizado um programa em python que lê o conteúdo criptografado do arquivo '**encrypted-text**' utilizando a lógica explicada e retorna o conteúdo descriptografado em um outro arquivo '**decrypted-text**'

Explicação do Código:

O código foi comentado e separado em funções para que pudesse ser melhor entendido.

- 1) Biblioteca math do Python utilizada para ajudar a calcular a fórmula de bhaskara.

```
import math # Biblioteca para calcular bhaskara
```

- 2) Foram utilizadas três declarações de lista, respectivamente, um para guardar os números contidos no arquivo '**encrypted-text**', o outro para colocar o valor de **x** resultado da função $f(x) = x^2 + 2x$, e por último, para colocar a mensagem descriptografada utilizando a lógica do alfabeto de **a até z**, onde o número de **x** será de **1 até 26**.

```
mensagem_criptografada = []  
resultado_funcao = []  
mensagem_descriptografada = []
```

- 3) Função para colocar os números contidos no arquivo '**encrypted-text**' em uma lista.

```
def mensagem_cripto_lista(arquivo):  
    for linha in arquivo: # Colocar os numeros da mensagem criptografada em uma lista  
        for n in linha.split(' '):  
            if n != '':  
                mensagem_criptografada.append(n)  
  
    calcular_bhaskara(mensagem_criptografada)
```

- 4) Função para calcular o valor de x utilizando a fórmula de Bhaskara

```
def calcular_bhaskara(mensagem_criptografada):  
    for numero_criptografado in mensagem_criptografada: # Colocar os numeros da mensagem criptografada em uma lista  
        if numero_criptografado != '\n': # Equação de bhaskara  
            numero_criptografado = int(numero_criptografado)  
            numero_criptografado = numero_criptografado * -1  
            calculo_bhaskara = (2**2) - (4*1*numero_criptografado)  
            calculo_bhaskara = math.sqrt(calculo_bhaskara)  
            calculo_bhaskara = (-2+calculo_bhaskara)/(2*1)  
            resultado_funcao.append(calculo_bhaskara)  
        elif numero_criptografado == '\n':  
            resultado_funcao.append('\n')  
  
    converter_numero_em_letra(resultado_funcao)
```

- 5) Função para retorno da lógica do alfabeto de **a até z**, onde o número de **x** será de **1 até 26**.

```
def converter_numero_em_letra(resultado_funcao):  
    for numero in resultado_funcao: # Conversão de números para letras  
        if numero == 1:  
            mensagem_descriptografada.append('a')  
        elif numero == 2:  
            mensagem_descriptografada.append('b')  
        elif numero == 3:  
            mensagem_descriptografada.append('c')  
        elif numero == 4:  
            mensagem_descriptografada.append('d')  
        elif numero == 5:  
            mensagem_descriptografada.append('e')  
        elif numero == 6:  
            mensagem_descriptografada.append('f')  
        elif numero == 7:  
            mensagem_descriptografada.append('g')  
        elif numero == 8:  
            mensagem_descriptografada.append('h')  
        elif numero == 9:  
            mensagem_descriptografada.append('i')  
        elif numero == 10:  
            mensagem_descriptografada.append('j')  
        elif numero == 11:  
            mensagem_descriptografada.append('k')  
        elif numero == 12:  
            mensagem_descriptografada.append('l')  
        elif numero == 13:  
            mensagem_descriptografada.append('m')  
        elif numero == 14:  
            mensagem_descriptografada.append('n')  
        elif numero == 15:  
            mensagem_descriptografada.append('o')  
        elif numero == 16:  
            mensagem_descriptografada.append('p')  
        elif numero == 17:  
            mensagem_descriptografada.append('q')  
        elif numero == 18:  
            mensagem_descriptografada.append('r')  
        elif numero == 19:  
            mensagem_descriptografada.append('s')  
        elif numero == 20:  
            mensagem_descriptografada.append('t')  
        elif numero == 21:  
            mensagem_descriptografada.append('u')  
        elif numero == 22:  
            mensagem_descriptografada.append('v')  
        elif numero == 23:  
            mensagem_descriptografada.append('w')  
        elif numero == 24:  
            mensagem_descriptografada.append('x')  
        elif numero == 25:  
            mensagem_descriptografada.append('y')  
        elif numero == 26:  
            mensagem_descriptografada.append('z')  
        elif numero == '\n':  
            mensagem_descriptografada.append('\n')  
  
    salvar_mensagem_descriptografa_em_arquivo(mensagem_descriptografada)
```

- 6) Função que salva a mensagem descriptografada no arquivo **decrypted-text**.

```
def salvar_mensagem_descriptografa_em_arquivo(mensagem_descriptografada):  
    arquivo = open('decrypted-text.txt', 'w')  
    for letra in mensagem_descriptografada:  
        arquivo.write(letra)  
  
    arquivo.close()
```

- 7) Função **main** que abre o arquivo criptografado e chama a primeira função apresentada no **tópico 3**.

```
if __name__ == '__main__':  
    arquivo = open('encrypted-text.txt', 'r') # Abertura do arquivo que contém a mensagem criptografada  
    mensagem_cripto_lista(arquivo) # Chamada da primeira função
```