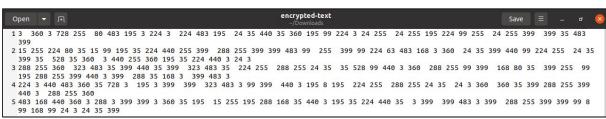
## **Desafio 2**

**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.





- 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:

$$f(x) = x^2 + 2x = 360$$

 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.

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
    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)
        calculo_bhaskara=(-2+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 convertor namero en letra(résultado funcio):

for numero in resultado funcio: * siminitad de numero para intras

if numero = 1:

    enmagen descriptografada append('a')

ili numero = 2:
    enmagen descriptografada append('c')

ili numero = 4:
    enmagen descriptografada append('c')

ili numero = 5:
    enmagen descriptografada append('c')

ili numero = 5:
    enmagen descriptografada append('c')

ili numero = 5:
    enmagen descriptografada append('c')

ili numero = 7:
    enmagen descriptografada append('c')

ili numero = 2:
    enmagen descriptografada append('c')

ili numero = 1:
    enmagen descriptografada append('c')

ili numero = 10:
    enmagen descriptografada append('c')

ili numero = 12:
    enmagen descriptografada append('c')

ili numero = 13:
    enmagen descriptografada append('c')

ili numero = 14:
    enmagen descriptografada append('c')

ili numero = 15:
    enmagen descriptografada append('c')

ili numero = 10:
    enmagen descriptografada append('c')

ili numero = 10:
    enmagen descriptografada append('c')

ili numero = 20:
    enmagen descriptografada append('c')

ili numero = 21:
    enmagen descriptografada appe
```

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.text', 'r') # Abertura do arquivo que contem a mensagem criptografada
    mensagem_cripto_lista(arquivo) #Chamada da primeira função
```