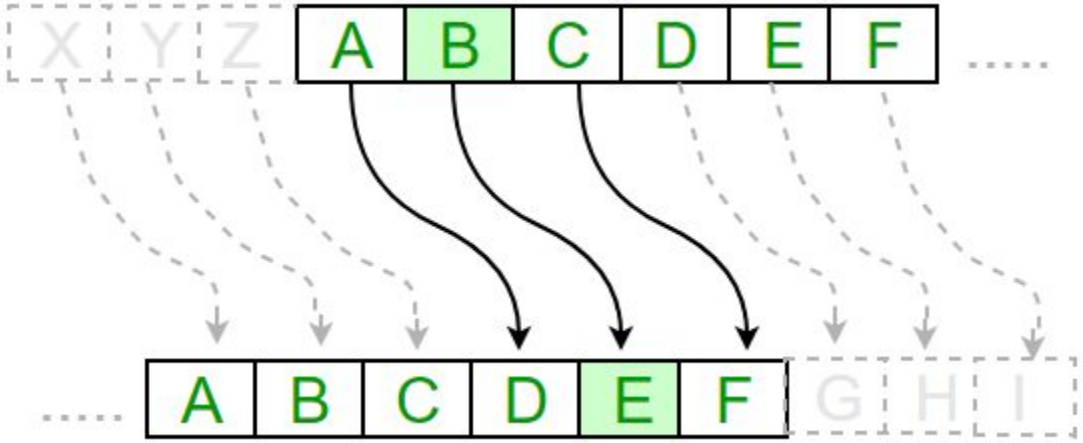
# Criptografia Clássica

# Cifra de César

- ➔ Atribuída a Júlio César

- ➔ Criptografia de substituição

# Como funciona



# Como funciona

$$E_n(x) = (x + n) \mod 26.$$

$$D_n(x) = (x - n) \mod 26.$$

# Encriptando

```
def encrypt(s, key):
    enc = ""

    for i in range(0, len(s)):
        if (s[i].isupper()):
            enc += chr(((ord(s[i]) - ord("A") + (key % 26 + 26)) % 26) + ord("A"))
        elif (s[i].islower()):
            enc += chr(((ord(s[i]) - ord("a") + (key % 26 + 26)) % 26) + ord("a"))
        else:
            enc += s[i]

    return enc
```

# Decriptando

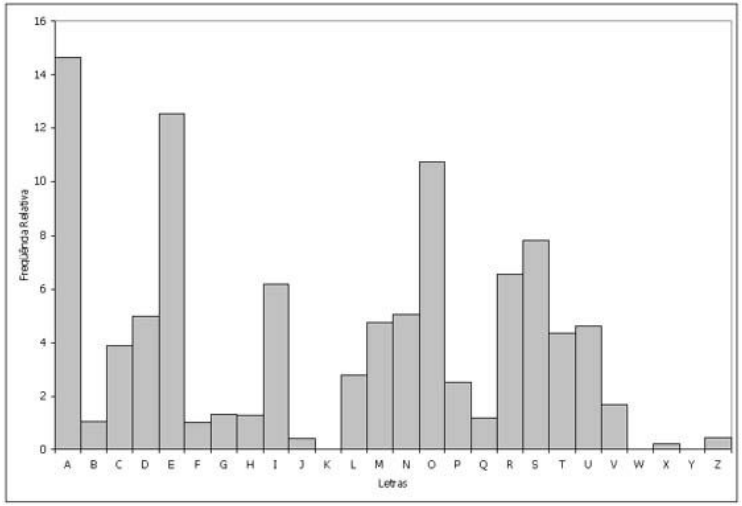
```
def decrypt(s, key):
    dec = ""

    for i in range(0, len(s)):
        if (s[i].isupper()):
            dec += chr(((ord(s[i]) - ord("A") - (key % 26 + 26)) % 26) + ord("A"))
        elif (s[i].islower()):
            dec += chr(((ord(s[i]) - ord("a") - (key % 26 + 26)) % 26) + ord("a"))
        else:
            dec += s[i]

    return dec
```

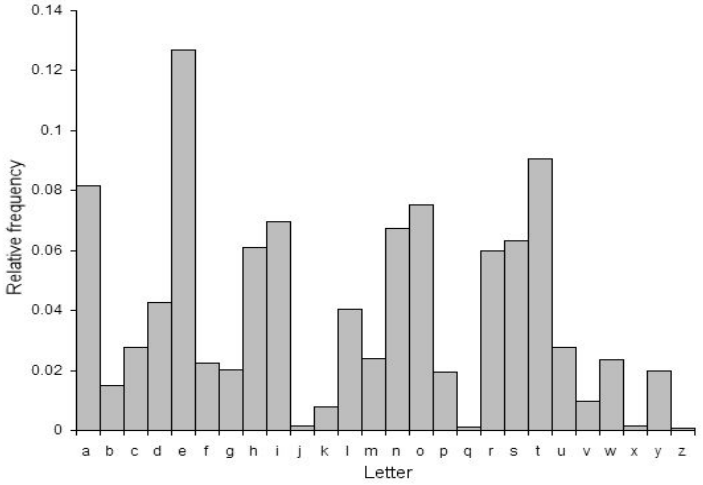
# Atacando por Brute Force

```
def decrypt(s, key):
    dec = ""

    for i in range(0, len(s)):
        if (s[i].isupper()):
            dec += chr(((ord(s[i]) - ord("A") - (key % 26 + 26)) % 26) + ord("A"))
        elif (s[i].islower()):
            dec += chr(((ord(s[i]) - ord("a") - (key % 26 + 26)) % 26) + ord("a"))
        else:
            dec += s[i]

    return dec
```

```
for i in range(0, 26):
    print("key: ", i, " - ", decrypt(s, i))
```

# Ataque por análise da frequência das letras

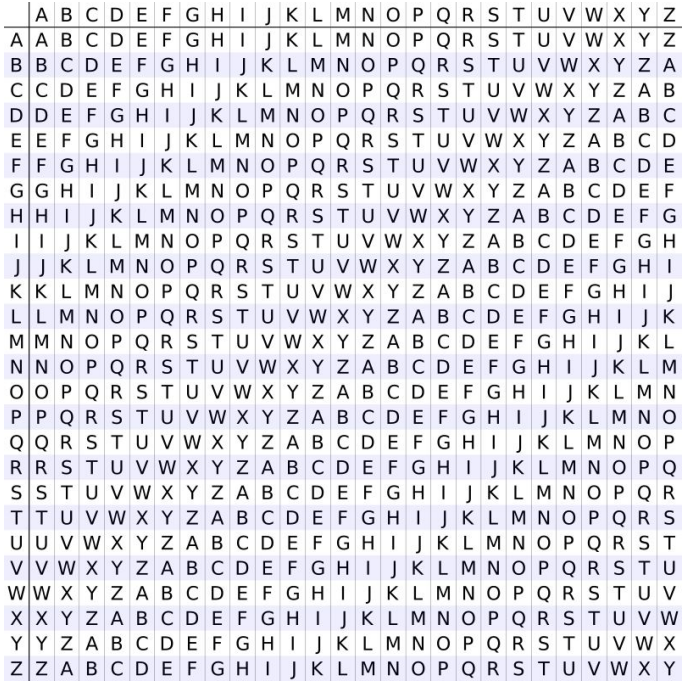


# Cifra de Vigenère

- Atribuída a Blaise de Vigenère
- Descrita por Giovan Battista Bellaso em 1553
- Criptografia de substituição polialfabética
- Série de cifras de César

# Como funciona



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

Exemplo:

Mensagem: “ATACARBASESUL”

Chave: “LIMAO”

Texto cifrado: “LBMCOCJMSSDCX”

# Como funciona

$$C_i = E_K(M_i) = (M_i + K_i) \mod 26$$

$$M_i = D_K(C_i) = (C_i - K_i) \mod 26$$

# Encriptando

```
def encrypt(s, key):
    enc = ""
    for i in range(len(s)):
        if (s[i].isupper()):
            value = (ord(s[i]) + ord(key[i % len(key)])) % 26
            enc += chr(value + ord("A"))
        else:
            enc += s[i]
    return enc
```

# Decriptando

```
def decrypt(s, key):
    dec = ""
    for i in range(len(s)):
        if (s[i].isupper()):
            value = (ord(s[i]) - ord(key[i % len(key)])) % 26
            dec += chr(value + ord("A"))
        else:
            dec += s[i]
    return dec
```

# Ataques

- ➔ Brute force com um dicionário

- ➔ Ataque de Charles Babbage (Friedrich Kasiski)

https://inventwithpython.com/hacking/chapter21.html

# Passos do Ataque de Babbage

1. Encontrar sequências repetidas
2. Pegar divisores do tamanho dos espaços
3. Separar a String de acordo com o tamanho da chave
4. Analisar a frequência das letras
5. Brute force com as possíveis chaves

Obrigada!