

Propostas



```
def codificador(info):  
    from random import randint  
    key = randint(1,10)  
    i = key  
    info_codificada = ''  
    for palavra in info.split():  
        for j in range(len(palavra)):  
            char = chr(ord(palavra[j])+i)  
            info_codificada = info_codificada + char  
            i+=1  
        info_codificada = info_codificada + ' '  
    info_codificada = info_codificada + str(key)  
    return info_codificada
```

```
def decodificador(info_cod):  
    lista = info_cod.split()  
    key = int(lista.pop())  
    i = key  
    info_decod = ''  
    for palavra in lista:  
        for j in range(len(palavra)):  
            char = chr(ord(palavra[j])-i)  
            info_decod = info_decod + char  
            i+=1  
        info_decod = info_decod + ' '  
    return info_decod
```

Uma das ideias inicialmente abordadas gerava uma chave aleatória que era alocada na string codificada para que depois pudesse ser extraída e utilizada na decodificação.

```

1 import string, random
2 from random import randint
3
4 texto = 'texto 123 ## TEXTO UM DOIS TRES @@'
5 cripto_lista = []
6 contador_chave = 0
7 chave = ''
8 cripto = ''
9 quebra = 'abcdefghijklmnopqrstuvwxyz#%'
10 sum = 0
11
12 ##CRIPTOGRAFANDO
13 while contador_chave < 10:
14     num = (randint(0,9))
15     chave = chave + chr(random.randint(ord('a'), ord('z')))) + str(num)
16     contador_chave += 1
17 chave += '##'
18
19 print(f'A chave para descriptografar é: {chave}')
20
21 for letra in quebra:
22     chave = chave.replace(letra, '')
23
24
25 for digit in str(chave):
26     sum += int(digit)
27
28
29 for letra in texto:
30     cripto_lista.append(str(ord(letra)*sum))
31     cripto_lista.append(chr(random.randint(ord('a'), ord('z'))))
32
33 cripto = "".join(cripto_lista)
34 print(f'A mensagem criptografada é: \n {cripto}')

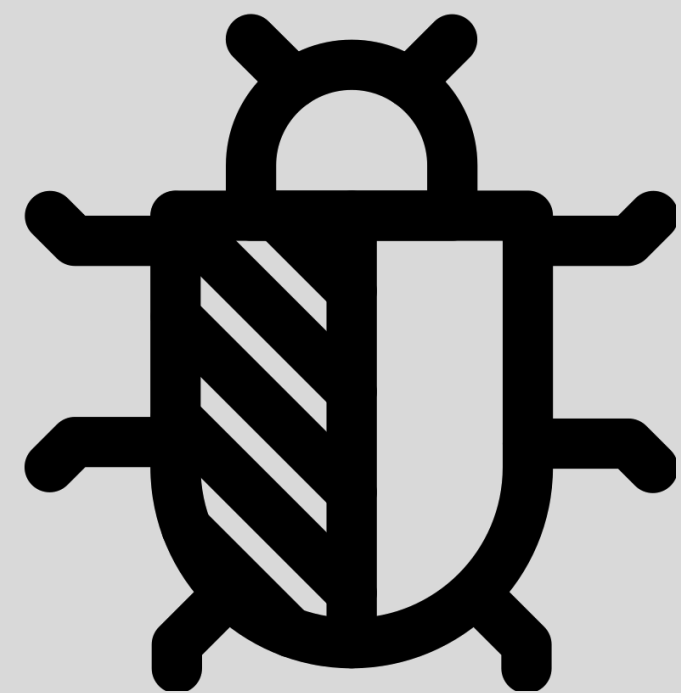
```

```

1 import string, random
2 from random import randint
3
4 texto = 'meu texto é 1'
5 chave_informada = 'b5h5e9q2t3t5b7a4o2w3#%'
6 cripto_informada = '5220i4545q5400u5220b4995t14480q2880r'
7 quebra = 'abcdefghijklmnopqrstuvwxyz#%'
8 decrip_lista = []
9 lista_texto = []
10 sum = 0
11 descripto = ''
12
13 ##DESCRIPTOGRAFAR
14 for letra in quebra:
15     cripto_informada = cripto_informada.replace(letra, '/')
16
17 decrip_lista = cripto_informada.split('/')
18 decrip_lista.pop(-1)
19
20
21 for letra in quebra:
22     chave_informada = chave_informada.replace(letra, '')
23
24
25
26 for digit in str(chave_informada):
27     sum += int(digit)
28
29 for numero in decrip_lista:
30     lista_texto.append(chr(int(int(numero)/int(sum))))
31
32
33 descripto = "".join(lista_texto)
34 print(f'O texto descriptografado é: "{descripto}"')
35

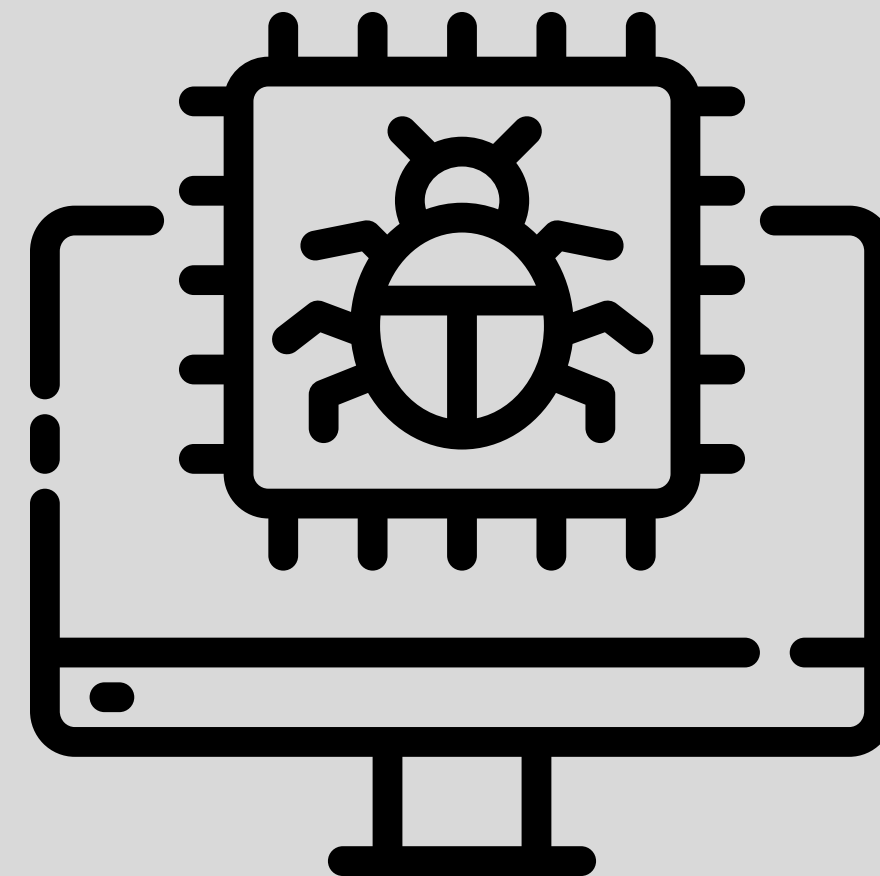
```

```
def cripta (texto, codigo):  
    cript = ''  
    cri = []  
    text = []  
    for i in texto:  
        i = int(ord(i))  
        text.append(i)  
    t = len(text)  
    for i in range(0,t):  
        va1 = text[i]  
        va2 = text[i]  
        cri.append(str(primo[va1]))  
        cri.append(alfa1[va2+1])  
        cri.append(random.choice(codigo))  
  
    cript = ''.join(cri)  
    return cript
```



DeBug

CRIPTOGRAFIA



Mensagem

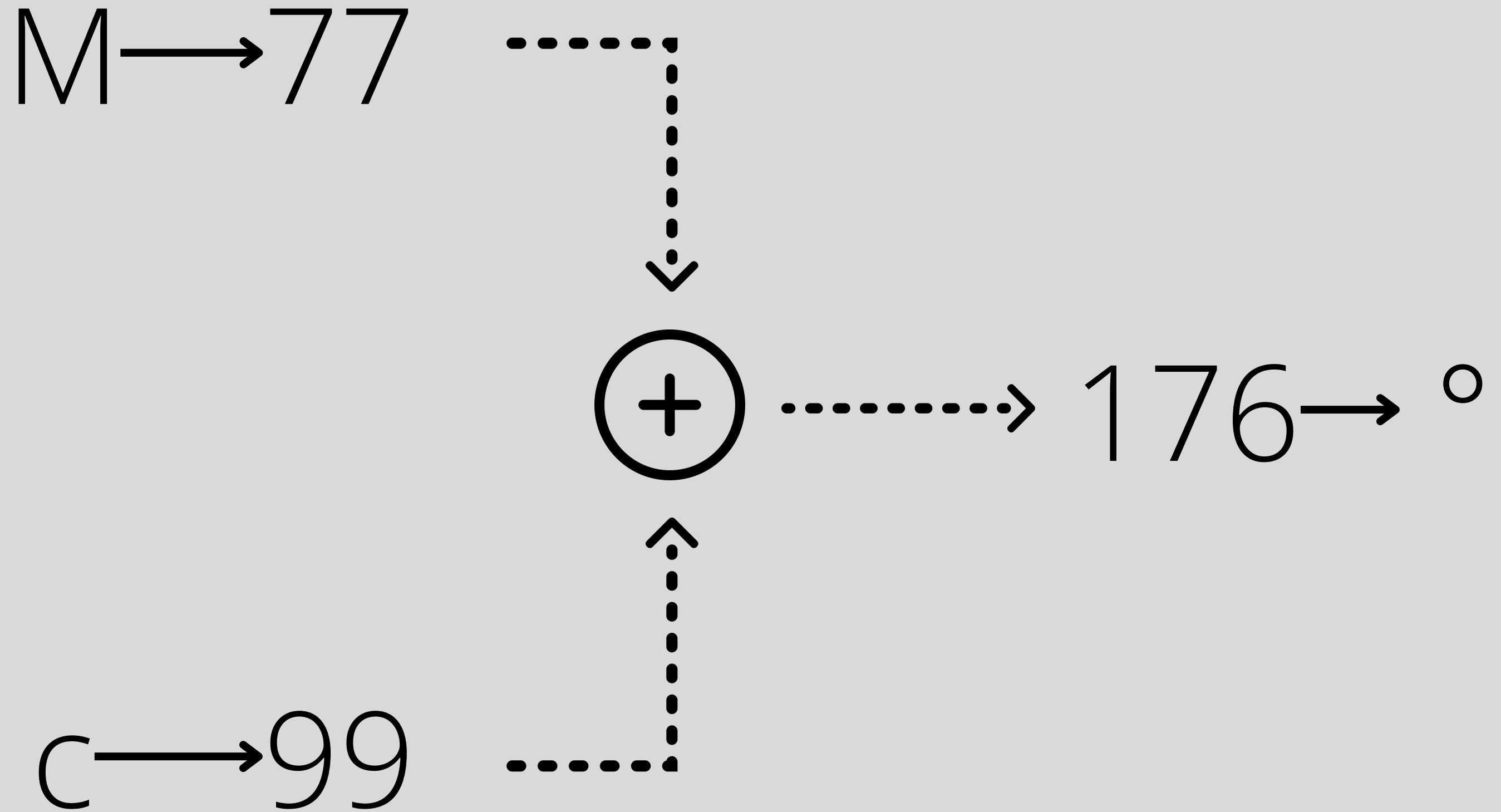
3 v 4 H c

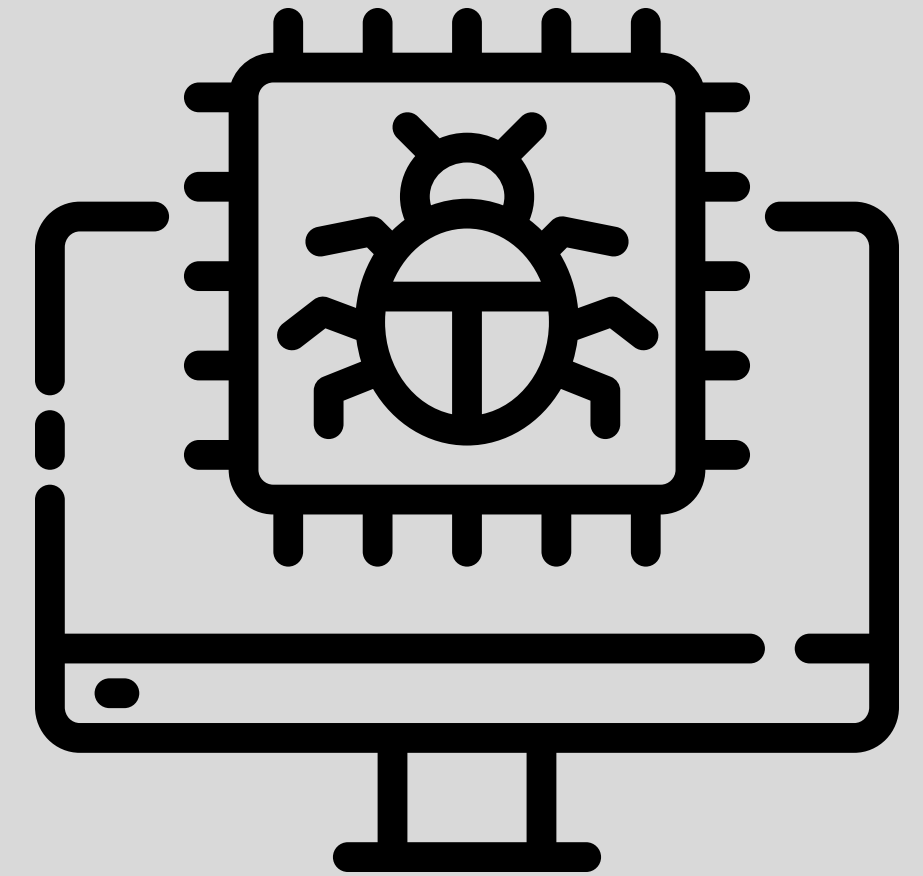
3 v 4 H c



c H 4 v 3

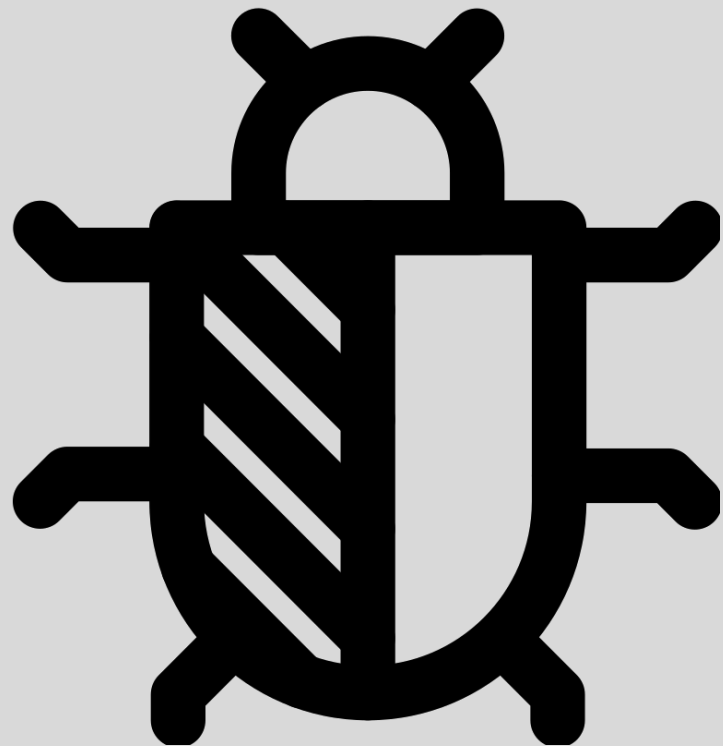
M	e	n	s	a	g	e	m
C	H	4	v	3	C	H	4





DeBug

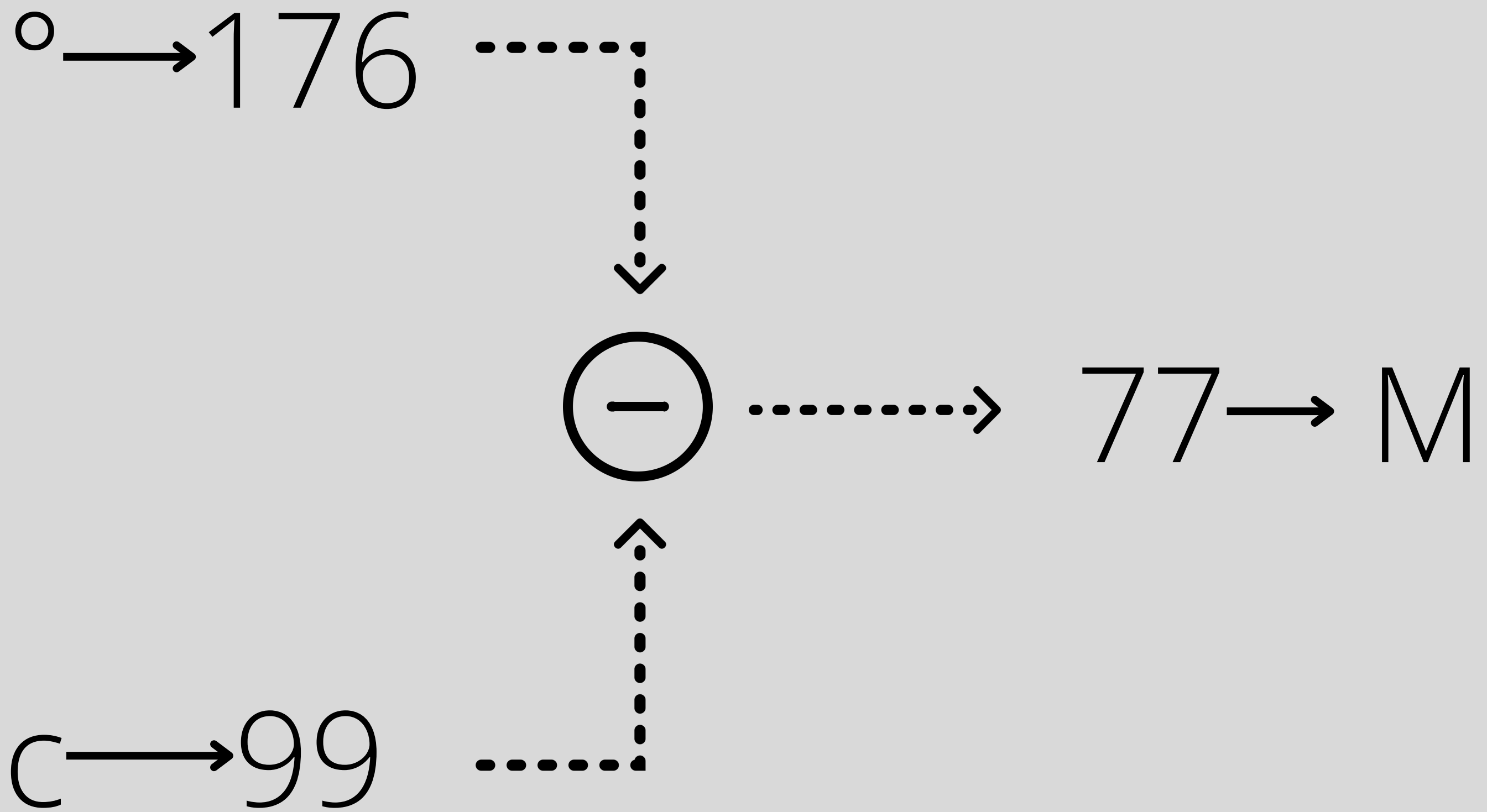
DECRYPTOGRAFIA

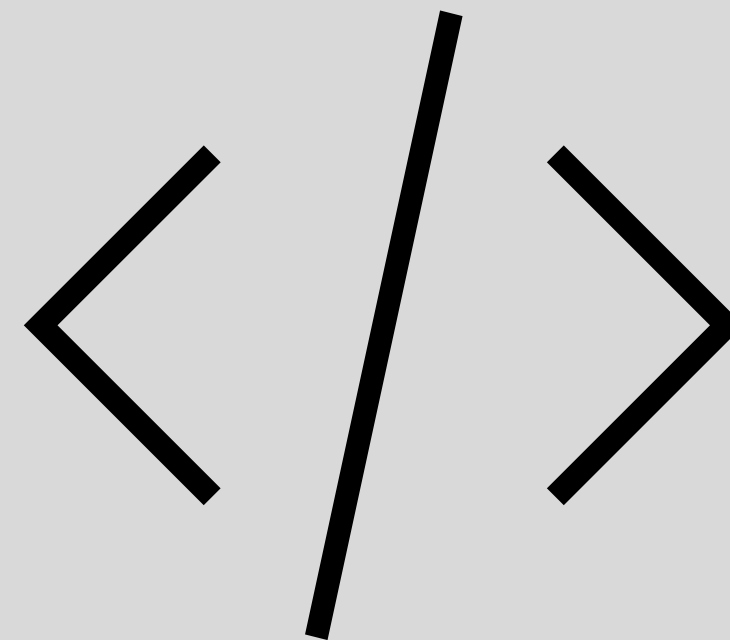


°çé Êî

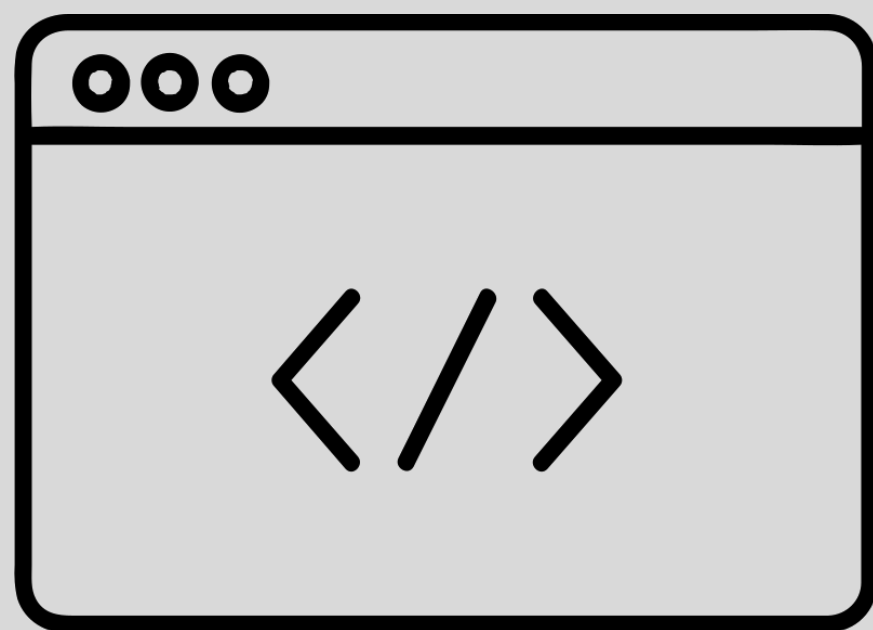
°	ç	é		Ê	i	
c	H	4	v	3	c	

Chave: 3 v 4 H c





Código

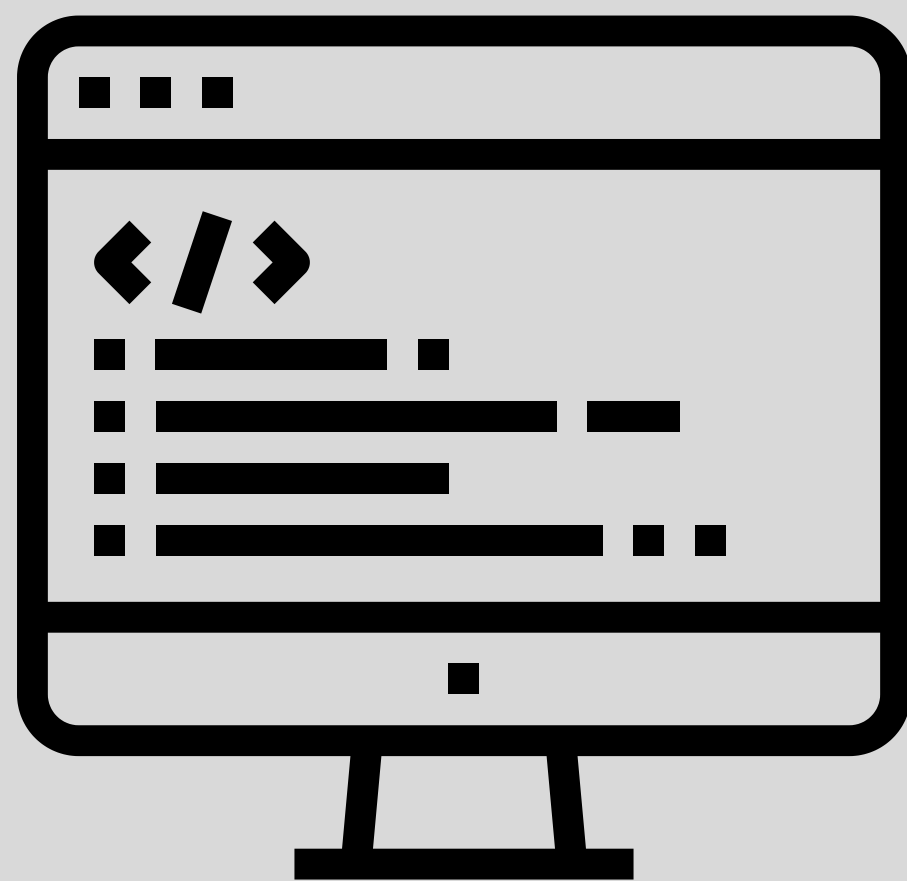




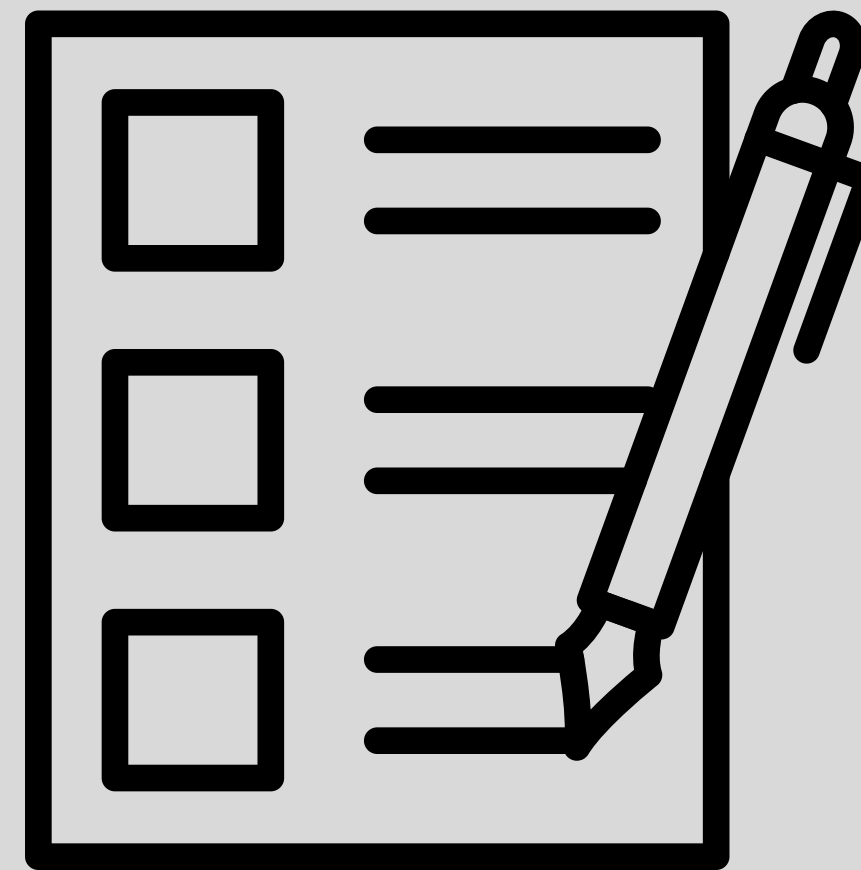
```
1  def encode_text(text: str, key: str) -> str:
2      key = key[::-1]
3      encode_text = ''
4      key_index = 0
5      for letter in text:
6          encode_text += chr(((ord(letter) + ord(key[key_index])) % 127) + 33)
7          key_index += 1
8          if key_index >= len(key):
9              key_index = 0
10
11     return encode_text
```




```
1  def decode_text(text: str, key: str) -> str:
2      key = key[::-1]
3      decode_text = ''
4      key_index = 0
5      for letter in text:
6          decode_text += chr(((ord(letter) - ord(key[key_index])) - 33) % 127)
7          key_index += 1
8          if key_index >= len(key):
9              key_index = 0
10
11     return decode_text
```



Testes





Dúvidas?

Obrigada

