

```
In [1]: import string # importing string function to get lowercase and uppercase letters
# encode_rot function
def encode_rot(text = str, key = int):
    encodeText = []
    lower = string.ascii_lowercase #lowercase Letters
    upper = string.ascii_uppercase #uppercase Letters
    for char in text:
        if char.islower() and char in lower: #checking for lower
            if key > 0:
                encodeText.append(lower[(lower.index(char)+key) % 26]) #moving the Lett
            elif key < 0:
                encodeText.append(lower[(lower.index(char)-key) % 26]) #movign the Lett
            else:
                encodeText.append(char) #appending the char if key == 0
        elif char.isupper() and char in upper: #checking for upper
            if key > 0:
                encodeText.append(upper[(upper.index(char)+key) % 26]) #moving the Lett
            elif key < 0:
                encodeText.append(upper[(upper.index(char)-key) % 26]) #moving the Lett
            else:
                encodeText.append(char) #appending the char if key == 0
        else:
            encodeText.append(char)
    return ''.join(encodeText)
```

```
In [2]: clear_text = "Machine CAN learn 2 !!!"

encode_rot(clear_text, 28)
```

```
Out[2]: 'Ocejkpg ECP ngctp 2 !!!'
```

```
In [3]: import string
def decode_rot(cipher_text):
    upper = string.ascii_uppercase #Lowercase Letters
    lower = string.ascii_lowercase #uppercase Letters
    keys = list(range(0, 26)) #iterating the keys from 0 to 25
    decoded_list = {}
    for key in keys:
        decoded = ""
        for char in (cipher_text):
            if char != " " and not char.isdigit() and (char in upper or char in lower):
                if char.isupper():
                    shifted_idx = ((upper.index(char)) - key) % 26
                    decoded += (upper[shifted_idx])
                if char.islower():
                    shifted_idx = (lower.index(char) - key) % 26
                    decoded += (lower[shifted_idx])
            else:
                decoded += (char)
        words = decoded.split()
        decoded_list[decoded] = 0
        for word in words:
            if word in open('dictionary.txt').read():
                decoded_list[decoded] += 1
    #
    print(f'{decoded_list},{decoded},{words},{key}')
```

```
final_string = [key for key, value in decoded_list.items() if value == max(decoded_
return final_string
```

In [4]:

```
# Test case
cipher_text = "Tqjq yi byau fuefbu, ydjuhewqju yj xqht udekwx qdt yj mybb jubba oek mxq
print(decode_rot(cipher_text))
```

Data is like people, interrogate it hard enough and it will tell you whatever you want to hear.

In [ ]:

In [ ]: