Caesar Cipher

A Caesar cipher is a simple method for encrypting and decrypting messages. The method works in the following way:

1. Pick a fixed number n.
2. Number the letters in the alphabet: `a` is 0, `b` is 1, ..., and `z` is 25.
3. Encode your message from letters to numbers.
4. Add n to each number. If the number is larger than 26, find its remainder when you divide by 26.
5. Decode your message from numbers to letters.

Decryption works by reversing the procedure given above.

Example: For $n = 3$ and message: `zing`,

1. `z i n g` $\rightarrow$ 25 8 13 6
2. 25 8 13 6 $\rightarrow$ 2 11 16 9
3. 2 11 16 9 $\rightarrow$ `c l q j`

The encrypted message is `clqj`.

Goals:

1. Write a function `caesarE(message,n)` that, given a message and a value n, returns the encrypted message given by the Caesar cipher with a shift of size n.
2. Write a function `caesarD(message,n)` that, given an **encrypted** message and a value n, returns the **decrypted** message given reversing the Caesar cipher with shift of size n.
3. **Bonus:** Decipher the following encrypted message:

`mujxufuefbuevjxukdyjutijqjuiydehtuhjevehcqcehufuhvusjkd`
`yeduijqrbyixzkijysuydikhutecuijysjhqdgkybyjofhelytuvehj`
`xuseccedtuvudiufhecejujxuwuduhqbmubvqhuqdtiuskhujxurbui`
`iydwievbyruhjojeekhiubluiqdtekhfeijuhyjoteehtqydqdtuijq`
`rbyixjxyisedijyjkjyedvehjxukdyjutijqjuievqcuhysq`

Test Cases:

1. $n = 27$, decrypted message: `super`, encrypted message: `tvqfs`
2. $n = -3023$, decrypted message: `stargazing`, encrypted message: `lmtkztsbgz`

A solution:

```
alphaToNum = {'a':0,'b':1,'c':2,'d':3,'e':4,'f':5,'g':6,'h':7,'i':8,'j':9,
'k':10,'l':11,'m':12,'n':13,'o':14,'p':15,'q':16,'r':17,'s':18,'t':19,'u':20,
'v':21,'w':11,'x':23,'y':24,'z':25}

numToAlpha = 'abcdefghijklmnopqrstuvwxyz'

def caesarE(message,n):
    s = ''
    for char in message:
        newNum = (alphaToNum[char]+n)%26
        s += numToAlpha[newNum]
    return s

def caesarD(message,n):
    return caesarE(message,-n)
```