Two's-Complement

The two's-complement representation of a non-negative integer is simply its standard representation in base 2. However, the two's-complement representation of a negative integer is obtained by first finding the base-2 representation of its absolute value, then flipping each bit, and finally adding the number 1.

This problem has two parts.

- First, write a function called `TcToNum` that takes as input a string of 8 bits representing an integer in two's-complement, and returns the corresponding integer. Notice here that since the input string is always exactly 8 bits long, we will often have leading 0s in the input, as in `"00000001"`. Here is sample input and output:

```
>>> TcToNum("00000001")
1
>>> TcToNum("11111111")
-1
>>> TcToNum("10000000")
-128
>>> TcToNum("01000000")
64
```

- Next, write a function called `NumToTc` that takes as input an integer N, and returns a string representing the two's-complement representation of that integer. You should assume that exactly 8 bits are being used, so only a certain range of numbers can be represented! The output should be exactly 8 bits long in all cases. If the input N is such that it cannot be represented in two's-complement with 8 bits, the function should return "Error". Here is some sample input and output:

```
>>> NumToTc(1)
'00000001'
>>> NumToTc(-128)
'10000000'
```

```
>>> NumToTc(200)

'Error'
```