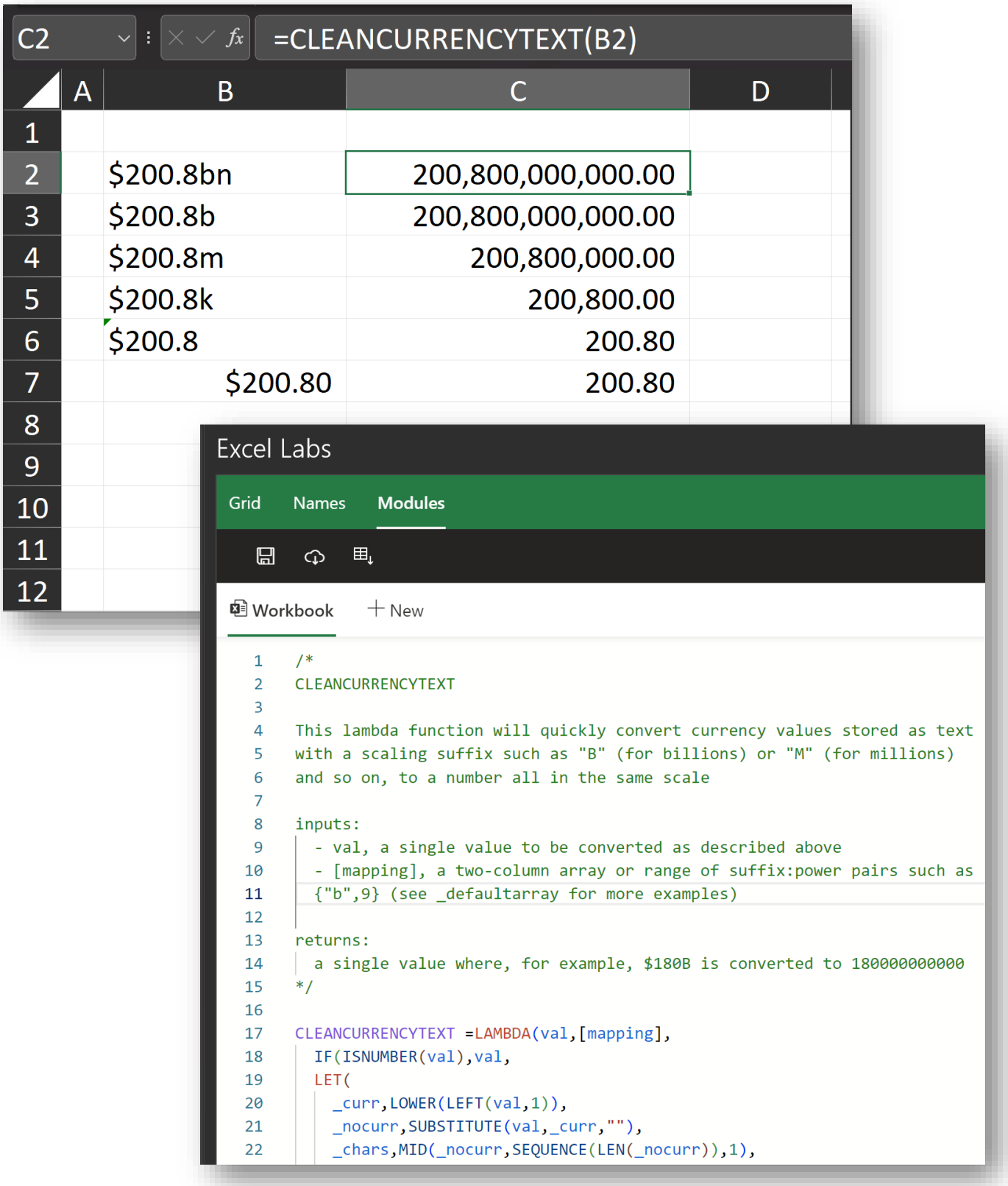
Excel LAMBDA function to standardize currency values



Owen Price

Explanation - 1

A single-cell value holding some number which you want to standardize

An optional two-column array or range of suffix:power pairs such as {“bn”,9}. See _defaultarray for more examples.

```
CLEANCURRENCYTEXT =LAMBDA(val,[mapping],
  IF(ISNUMBER(val),val,
  LET(
    _curr,LOWER(LEFT(val,1)),
    _nocurr,SUBSTITUTE(val,_curr,""),
    _chars,MID(_nocurr,SEQUENCE(LEN(_nocurr)),1),
    _nonnumeric,FILTER(_chars,ISERR(INT(_chars))),
```

If **val** is a number, just return it.

Otherwise, let:

- **_curr** be the first character in the text (assumed to be the currency)
- **_nocurr** be the result of replacing the currency with an empty string
- **_chars** be the array of characters in **_nocurr**
- **_nonnumeric** be the non-numeric characters in **_chars**

	A	B	C	D	E	F
1		val	_curr	_nocurr	_chars	_nonnumeric
2		$200.8bn	$	200.8bn	2	.
3					0	b
4					0	n
5					.	
6					8	
7					b	
8					n	

Explanation - 2

```
_filtered,FILTER(_nonnumeric,
                 (_nonnumeric<>".")*(_nonnumeric<>",")),
_joined,TEXTJOIN("",TRUE,_filtered),
_suffix,IFERROR(_joined,"nope"),
_defaultmapping,{"b",9;"bn",9;"bns",9;
                 "m",6;"mm",6;"mn",6;
                 "k",3;"nope",0},
```

Further, let:

- **_filtered** be the non-numeric characters which aren't either a period or a comma. The assumption is this returns the characters that form the suffix.
- **_joined** be the result of joining suffix characters found in **_filtered**
- **_suffix** be the replacement of an error in **_joined** with any text (here I've used "nope")
- **_defaultmapping** be a two-column array of common suffix multipliers

	A	B	C	D	E	F	G
1		val	_filtered	_joined	_suffix	_defaultmapping	
2		$200.8bn	b	bn	bn	b	9
3			n			bn	9
4						bns	9
5						m	6
6						mm	6
7						mn	6
8						k	3
9						nope	0

Explanation - 3

```
    _mapping,IF(ISOMITTED(mapping),
                _defaultmapping,mapping),
    _multiplier,POWER(
                    10,
                    XLOOKUP(
                      _suffix,
                      INDEX(_mapping,,1),
                      INDEX(_mapping,,2),
                      0)),
    _nosuffix,SUBSTITUTE(_nocurr,_suffix,""),
    _output,_nosuffix*_multiplier,
```

Further, let:

- **_mapping** be the mapping passed to the function or the **_defaultmapping** if mapping was omitted
- **_multiplier** be the Nth power of 10 where N is the value returned from **_mapping** by looking for **_suffix**
- **_nosuffix** be the result of replacing suffix in **_nocurr** with an empty string
- **_output** be the result of multiplying **_nosuffix** by **_multiplier**

	A	B	C	D	E	F	G
1		val	_mapping		_multiplier	_nosuffix	_output
2		$200.8bn	b	9	1000000000	200.8	200800000000.00
3			bn	9			
4			bns	9			
5			m	6			
6			mm	6			
7			mn	6			
8			k	3			
9			nope	0			

But why?

Now that we've converted the numbers to a common scale, we can use them in calculations!

	A	B	C	D
1				
2		$200.8bn	200,800,000,000.00	=CLEANCURRENCYTEXT(B2)
3		$200.8M	200,800,000.00	=CLEANCURRENCYTEXT(B3)
4		$200.8k	200,800.00	=CLEANCURRENCYTEXT(B4)
5		$200.80	200.80	=CLEANCURRENCYTEXT(B5)
6		$200.8	200.80	=CLEANCURRENCYTEXT(B6)
7				
8		$200.80	201,001,001,201.60	
9		=SUM(B2:B6)	=SUM(C2:C6)	
10				

Grab the function