

## CoCoALib - Feature #1490

### New function: MinusOneToPower

25 Sep 2020 13:35 - John Abbott

<b>Status:</b>	Rejected	<b>Start date:</b>	25 Sep 2020
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	New Function	<b>Estimated time:</b>	0.99 hour
<b>Target version:</b>	CoCoALib-0.99800	<b>Spent time:</b>	0.95 hour
<b>Description</b>			
I have just seen in the code for matrix adjoint <b>power(-1, i+j)</b> . It would be nice to have a simpler function for computing a power of -1.  If so, what should it be called?			

### History

#### #1 - 25 Sep 2020 13:37 - John Abbott

There is a very simple inline impl:

```
long MinusOneToPower(long n)
{
    return (IsEven(n) ? 1 : -1;
}
```

If we do accept this fn, in which (header) file should it go?

#### #2 - 25 Sep 2020 16:21 - Anna Maria Bigatti

John Abbott wrote:

There is a very simple inline impl:  
[...]

If we do accept this fn, in which (header) file should it go?

Just after `power(long, long)`, I'd say

### #3 - 25 Sep 2020 17:44 - John Abbott

- Status changed from New to In Progress
- % Done changed from 0 to 10

It would be tricky to call the function power because I want the return type to be long (or even int) rather than BigInt. The power function (family) in BigIntOps all return BigInt which is a bit wasteful -- more irritating than exorbitantly slow.

Possible names include MinusOnePower or MinusOneToPower; here the "minus one" is in the name rather than as an explicit argument.

JAA thinks there ought to be a better (compact) name... but what?

**UPDATE** nor more compact, but possibly better **PowerOfMinusOne**, at least it starts with Power which is like power.

### #4 - 03 Oct 2020 17:41 - John Abbott

- Status changed from In Progress to Resolved
- Assignee set to John Abbott
- % Done changed from 10 to 70

I have implemented **PowerOfMinusOne** (for both MachineInt and BigInt exponents). Currently they are not inline; should I make them inline? (seems to be harmless)

No doc; no tests.

### #5 - 05 Oct 2020 11:40 - John Abbott

Now I am having doubts about this function...

The normal power function just calls mpz\_power, and I fully expect that this GMP function handles "cleverly" the case of high powers of 1, 0, and -1. The only advantage MinusOneToPower has is that the result is a long rather than a BigInt; but *when is that a genuine advantage?*

So now, I'm considering deleting the impl, and rejecting this issue. What do you think?

### #6 - 06 Oct 2020 09:53 - John Abbott

- Status changed from Resolved to Rejected
- % Done changed from 70 to 100
- Estimated time set to 0.99 h

## REJECTED

There is no need for **MinusOneToPower**; it can be achieved by **SmallPower** (if we want to avoid BigInt) or by power (with the overhead of creating a BigInt).

I have removed the fn defn, and the doc.