

CoCoALib - Design #1608

Domain of definition of NextPrime (and PrevPrime)

16 Sep 2021 22:03 - John Abbott

<b>Status:</b>	Closed	<b>Start date:</b>	16 Sep 2021
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	John Abbott	<b>% Done:</b>	100%
<b>Category:</b>	Tidying	<b>Estimated time:</b>	1.01 hour
<b>Target version:</b>	CoCoALib-0.99800	<b>Spent time:</b>	1.00 hour
<b>Description</b>			
Not so important: what is the domain of defn of NextPrime and that of PrevPrime?			
I tried to compute NextPrime(0) but obtained an error saying the arg must be (strictly) positive. Also PrevPrime(2) in CoCoA-5 gave an unhelpful/misleading error message (ArgTooBig).			
Decide defn, and improve error mesgs.			

History

#1 - 20 Sep 2021 13:24 - John Abbott

- Description updated
- Status changed from New to In Progress
- Assignee set to John Abbott
- % Done changed from 0 to 10

I am now inclined to allow 0 as arg to NextPrime; perhaps this is related to my preference to consider 0 as a "natural number".

I have doubts about PrevPrime returning 0 when given 2 (or 1 or 0) as input. Surely an error would be more appropriate?

I wonder why I wrote the code the way I did...? (Documentation? Pah!)

#2 - 20 Sep 2021 14:08 - John Abbott

- Status changed from In Progress to Feedback
- % Done changed from 10 to 90

I have spoken to Anna who agrees to change the behaviour:

- both fns throw BadArg (or similar) if given (strictly) negative args
- PrevPrime throws OutOfRange if arg is  $0 \leq n \leq 2$ .
- NextPrime returns 0 if no next small prime exists
- Similar mods to NextProbPrime and PrevProbPrime.

Modified doc. Modified test-NumTheory1.C.

#3 - 20 Oct 2021 22:09 - John Abbott

- Status changed from Feedback to Closed
- % Done changed from 90 to 100

- Estimated time set to 1.01 h