

CoCoALib - Feature #1472

Idea for for loops

23 Jul 2020 20:36 - John Abbott

Status:	In Progress	Start date:	23 Jul 2020
Priority:	Low	Due date:	
Assignee:		% Done:	10%
Category:	New Function	Estimated time:	0.00 hour
Target version:	CoCoALib-0.99900	Spent time:	0.70 hour
Description			
Victor Shoup told me that he has written a little class in NTL which allows him to write			
<pre>for (long j: range(n)) {...}</pre>			
instead of the old style			
<pre>for (long j=0; j < n; ++j) {...}</pre>			
Do we want something similar in CoCoALib? I thought C++(14) already something like but did not find after a quick search on internet.			
Related issues:			
Related to CoCoALib - Feature #312: LongRange(a,b) returning vector of long a...		Closed	14 Feb 2013
Related to CoCoALib - Design #601: LongRange: in which file should it be decl...		In Progress	31 Jul 2014
Related to CoCoALib - Design #1346: C++14: use the new for loop syntax where ...		In Progress	21 Oct 2019

History

#1 - 23 Jul 2020 20:39 - John Abbott

While most loops in C++ do start from 0, I think I would like to have the ability to say whether to start from 0 or 1.
For example we could have:

```
for (long j: from0to(n)) {...}  
for (long j: from1to(n)) {...}
```

Ahhh! Now I see a slight problem. The from0to range would normally stop at n-1 rather than n. Hmmmm

#2 - 23 Jul 2020 20:40 - John Abbott

- Related to Feature #312: LongRange(a,b) returning vector of long a..b (included) added

#3 - 23 Jul 2020 20:40 - John Abbott

- Related to Design #601: LongRange: in which file should it be declared/defined? added

#4 - 23 Jul 2020 21:00 - Anna Maria Bigatti

range(0,n) ?

#5 - 25 Sep 2020 11:04 - John Abbott

- Status changed from New to In Progress

- Priority changed from Normal to Low

- Target version changed from CoCoALib-0.99800 to CoCoALib-0.99850

- % Done changed from 0 to 10

Anna's suggestion of **range(1,n)** etc. seems simple, clear, flexible.
The intention is that both values are included, right?

We might need to be careful if the max value is the largest possible long.

To use the new "range for" syntax, range(a,b) has to produce a sort of iterator; I wonder how fast that is compared to a classical for loop. Maybe this should not matter; if it is measurably slower (for almost empty loops) then we can always use a classical for loop where needed for speed.

It would be nice to eliminate the existing **LongRange** function, or replace it with this new iterator. Perhaps there could be automatic conversion from this new iterator type to vector<long>?

I have marked this as "low" priority, and postponed to 0.99750 as it is not urgent... rather just "icing on the cake".

#6 - 25 Sep 2020 11:22 - John Abbott

- Related to Design #1346: C++14: use the new for loop syntax where appropriate (like cocoa's foreach) added

#7 - 25 Sep 2020 16:16 - Anna Maria Bigatti

just wondering: (I know it's low priority) is this any better than the standard syntax?

#8 - 16 Feb 2024 09:55 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99900

While it can be difficult to say much positive about Julia, it does offer a convenient way of using "iterators" (and also a very compact notation). Perhaps we can take some inspiration from there?