CoCoALib - Bug #1233

factor SEGV -- too large characteristic

08 Nov 2018 09:57 - John Abbott

Status:	Resolved	**Start date:**	08 Nov 2018
Priority:	High	**Due date:**	
Assignee:	John Abbott	**% Done:**	60%
Category:	Maths Bugs	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99880	**Spent time:**	1.70 hour

Description

factor can cause a SEGV if the characteristic is NextProbPrime(2^31).

```
p := NextProbPrime(2^31);
use ZZ/(p)[x];
factor(x^2+1); --> SEGV
```

History

#1 - 08 Nov 2018 09:58 - John Abbott

Obviously I discovered the bug while demoing CoCoA-5 :-/

Ange would like to be able to factorize (univariate polys) over large characteristic.

#2 - 14 Nov 2018 17:20 - John Abbott

- Status changed from New to Resolved

- Assignee set to John Abbott

- % Done changed from 0 to 60

I have found two bugs related to the old code in the factorizer which uses int (for speed?) rather than long.
I have patched the bugs, but cannot yet check in until I fix my VPN settings.

This does prompt some questions:

- make two copies of the old factorizer code (one based on int, and the other based on long) and see if there is a significant speed difference
- if there is a significant speed difference then both versions must be incorporated and a suitable dispatch function added.
- there should also be an impl for "unlimited" characteristic (which is to be used only when native integers cannot cope)

But where to find the time to do all this?

#3 - 26 Feb 2019 17:05 - John Abbott

- Target version changed from CoCoALib-0.99650 November 2019 to CoCoALib-0.99700

#4 - 04 Dec 2019 17:38 - John Abbott

This now gives error rather than SEGV -- so that is an improvement!

#5 - 08 Jan 2020 22:44 - John Abbott

- Target version changed from CoCoALib-0.99700 to CoCoALib-0.99800

#6 - 26 Mar 2020 11:41 - John Abbott

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

#7 - 25 Sep 2020 15:24 - John Abbott

- Subject changed from factor SEGV to factor SEGV -- too large characteristic

#8 - 09 Nov 2020 21:02 - John Abbott

Revised example (now that we're mostly 64 bit)

```
p := NextProbPrime(2^63);
use ZZ/(p)[x];
factor(x^2+1); --> SEGV
```

I have just tested the example: it no longer gives SEGV, but does give a CoCoA error (*cannot convert*).

#9 - 14 Mar 2023 19:45 - John Abbott

- Target version changed from CoCoALib-0.99850 to CoCoALib-0.99880

I have just tried:

```
p := NextPrime(2^31);
use ZZ/(p)[x];
factor(x^2+1);  --> almost instant
p := NextPrime(p);
use ZZ/(p)[x];
factor(x^2+1);   --> SLOW!  stopped it after several minutes.
```

Not sure if it can really be made any faster. It would be nice if the code were interruptible.
Current state is "tolerable", but the issue is not really resolved. Postponing.