CoCoALib - Bug #1811

Check include guards

28 Mar 2024 23:00 - John Abbott

Status:	Feedback	**Start date:**	28 Mar 2024
Priority:	Normal	**Due date:**	
Assignee:	John Abbott	**% Done:**	90%
Category:	Tidying	**Estimated time:**	0.00 hour
Target version:	CoCoALib-0.99880	**Spent time:**	0.70 hour

Description

Some files have include guards without the CoCoA_ prefix

```
TmpJBAlgorithm.H:#ifndef TMPJBALGORITHM_H_
TmpMayerVietorisTree.H:#ifndef TmpMVT_H
TmpMonomialFunctions.H:#ifndef TmpMonFun_H
TmpPPVector.H:#ifndef TmpPPVector_H
```

Rectify

History

#1 - 28 Mar 2024 23:04 - John Abbott

- Status changed from New to Resolved

- % Done changed from 0 to 80

Also the interpreter has CPP symbols without the CoCoA prefix. Is that right?

#2 - 29 Mar 2024 08:24 - Anna Maria Bigatti

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

#3 - 31 Mar 2024 20:35 - Nico Mexis

Just a small question to throw in the room: Might it be worth considering to migrate to #pragma once?
Despite not being standardized, Wikipedia (https://en.wikipedia.org/wiki/Pragma_once#Portability) has a list of compilers and their support status (none of the recent versions specify "no").
Then, such issues (and also name clashes) cannot become a problem in the future anymore.

I also searched for any reasons not to use it and only found this response on StackOverflow: https://stackoverflow.com/a/34884735/5894824
However, a little more research reveals that the linked GCC bug 11569 has been fixed since 2003, and even on Reddit there are heated discussions about whether the points made in the SO post are correct (yet, nobody seems to have found failing scenarios).

Just ignore this comment, if it is not of interest/out of scope, by the way!

#4 - 22 Apr 2024 20:15 - John Abbott

Thanks to Nico for the suggestion about #pragma. I read also the discussion about its pros and cons -- it is not clear what should happen if one tries to do "acrobatics" with the preprocessor... [not relevant to our use]

I am hesitating because, as Nico wrote, the #pragma is not standard (though recognized by "every" compiler).
At this point it requires less time just to check & correct the few include-guard symbols (already done) than to change every header file to use #pragma.

As for the CPP symbols in the interpreter... strictly that is not part of CoCoALib, so is not constrained by the coding standards of CoCoALib: in other words, there is no obligation to use the CoCoA_ prefix... but it might be a good idea anyway? Perhaps CoCoA5_ is a better prefix?

#5 - 22 Apr 2024 20:15 - John Abbott

- Status changed from Resolved to Feedback

- Assignee set to John Abbott

- % Done changed from 80 to 90