

# Re: New symbolic/numeric/dynamic/intuitive programming language

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2008-02/msg03589.html>

---

- *From:* Michael Press <[rubrum@xxxxxxxxxxx](mailto:rubrum@xxxxxxxxxxx)>
  - *Date:* Thu, 21 Feb 2008 16:47:04 -0800
- 

In article <[rem-2008feb21-004@xxxxxxxxxx](mailto:rem-2008feb21-004@xxxxxxxxxx)>, [rem642b@xxxxxxxxxx](mailto:rem642b@xxxxxxxxxx) (Robert Maas, see <http://tinyurl.com/uh3t>) wrote:

From: "Mark Nudelman" <[ma...@xxxxxxxxxxxxxxxxxxxxxxxx](mailto:ma...@xxxxxxxxxxxxxxxxxxxxxxxx)>  
This seems like an instance of the COBOL school of language design, the philosophy being that the closer a program looks to ordinary English, the better it is.

COBOL actually had at least one good idea: PICTURE clauses.

In COBOL's formative years, 80-column Hollerith cards, with all data laid out in fixed columns, were the standard for input and output. Even printed reports used fixed format.

When a data layout was specified once, it could be used both to parse incoming data and to generate outgoing data in the same format. The modern equivalent for variable-length fields with nested structure would be BNF. (The more popular regular expressions, aren't powerful enough, IMO.) I haven't seen any programming language that directly used BNF for its standard way to specify input/output parsing/formatting, which I see as a deficiency in *\*all\** programming languages to date.

Are you saying that lex and yacc fail this test?

—

Michael Press

.