**Ma. Adriana Nicole Trinidad BSCS 3A**

**LOLCODE** is an esoteric programming language created in 2007 by Adam Lindsay, inspired by Internet "lolcat" memes. Despite its humorous syntax and unconventional structure, LOLCODE is Turing-complete, meaning it can theoretically perform any computation that other programming languages can.

All LOLCODE programs begin with HAI (followed by an optional version number) and end with KTHXBYE. Comments are denoted by BTW for single-line comments and OBTW...TLDR for multi-line comments.

LOLCODE supports several primitive data types:

* **YARN**: String literals (enclosed in double quotes)
* **NUMBR**: Integer values
* **NUMBAR**: Floating-point numbers
* **TROOF**: Boolean values (WIN for true, FAIL for false)
* **NOOB**: Undefined value

Variables are declared using the syntax I HAS A variableName with optional initialization via ITZ.

LOLCODE uses verbose syntax for operations:

**Arithmetic:** SUM OF x AN y (addition), DIFF OF x AN y (subtraction), PRODUKT OF x AN y (multiplication), QUOSHUNT OF x AN y (division), MOD OF x AN y (modulo)

**Comparison:** BOTH SAEM x AN y (equality), DIFFRINT x AN y (inequality), BIGGR OF x AN y (maximum), SMALLR OF x AN y (minimum)

**Boolean Logic:** BOTH OF x AN y (logical AND), EITHER OF x AN y (logical OR), WON OF x AN y (logical XOR), NOT x (logical NOT)

Control structures in LOLCODE include conditionals with O RLY?, YA RLY, MEBBE, NO WAI, and OIC; loops using IM IN YR loop and IM OUTTA YR loop; and switch statements with WTF?, OMG, and OMGWTF. Input/output operations are handled with VISIBLE for output and GIMMEH for input.

Functions are defined using HOW IZ I functionName YR param1 AN YR param2 and return values with FOUND YR result, while being terminated with IF U SAY SO. Function calls use the syntax I IZ functionName YR arg1 AN YR arg2.

Building a LOLCODE interpreter involves:

1. **Lexical Analysis**: Tokenizing the unique syntax and keywords
2. **Parsing**: Building an abstract syntax tree (AST) from tokens
3. **Semantic Analysis**: Validating the program structure
4. **Execution Engine**: Interpreting or compiling the AST

Since LOLCODE's specification is somewhat loose, interpreter implementations may vary in how they handle edge cases and language features, making standardization a challenge for serious development.