

COSC 251 – Programming Languages

Project 4

Spring 2018

Objective: The esoteric languages strike again! Build an interpreter for an esoteric language in LISP.

Your Task:

For this project, you and your group will create an interpreter for the Ook! esoteric language. The language has a small number of potential operations, listed below. For your interpreter, you will need to take in a filename, read the file and interpret the Ook! code in that file, outputting as necessary.

NOTE: We have not, nor will we discuss file operations in LISP. You're on your own there.

Ook!

Ook! is a trivial iteration of a language called brain*ck. The language utilizes an infinite length tape (a one-dimensional array), indexed from 0, with every entry in the tape initialized to 0. The pointer referenced by the commands below is initially pointing at the tape entry at index 0.

Commands:

Ook!	Description
Ook. Ook?	Move the pointer to the right.
Ook? Ook.	Move the pointer to the left.
Ook. Ook.	Increment the memory cell under the pointer.
Ook! Ook!	Decrement the memory cell under the pointer.
Ook! Ook.	Output the character signified by the cell at the pointer.
Ook. Ook!	Input a character and store it in the cell at the pointer.
Ook! Ook?	Jump past the matching Ook? Ook! if the cell under the pointer is 0.
Ook? Ook!	Jump back to the matching Ook! Ook?

Errors:

You are required to catch and report any errors dealing with the file I/O operations or errors in the source itself. Your error report should be descriptive enough for a programmer to debug effectively. Errors that should be caught and reported:

- No file found
- Incorrect Ook syntax (for instance, Ook% or Ook? Ook?)
- Dangling Ook (odd number of Ooks)
- Ook? Ook. call while at the first cell.

Restrictions: You may not use any third party libraries. The file can be any length and may contain syntax errors that you must catch and report. You may not translate the source code to any other language (other than LISP). You may not set an arbitrary length for the tape.

Implementation details: you must implement the interpreter in a single file, with an entry function called `ook` that takes in a single string parameter. Calls to your function will look like:

(ook "source.txt")

You may work in groups of three on this project. Teams must be sent to us via email by 5pm April 5th. Failure to do so and continuing to work together as a group will result in a 30 point deduction.

Expectations: The code should be clean, concise, well-commented and correct. If you use an outside source, be sure to document that source (we are aware of the `Ook!` LISP interpreter on github). Significant use of outside sources will result in a deduction. Example sources will be provided soon. Grading rubric will be provided a week ahead of the due date.

DUE: April 20th, 11:59pm via Blackboard.