

# User Manual – Course Project Phase 1

Liam Wrubleski – 10168579

## Usage

### Compilation

Run the following commands on a system with Flex installed (assuming Linux, tested on Kali Linux [variant of Debian]):

```
flex a1.flex
```

```
gcc lex.yy.c -lfl -o parser
```

### Running

Run the program from the folder in which it was compiled by running the command:

```
./parser [-v] [-o outputfile] [filename]
```

Flag	Param	Effect
<b>-v</b>	none	(optional) prints to stdout in addition to outputfile.
<b>-o</b>	Outputfile (mandatory)	(optional) prints to outputfile. Default output file is parser.res.
<b>[none]</b>	filename	(optional) selects file to parse. If no file is passed, parses stdin.

Note: I got tired of looking inside a file, so I added the -v flag. Then I realized I didn't know what the file output should be called, so I added the -o flag. Finally, just for fun, I made the parameters order-agnostic.

## Functionality

### Working

The parser detects all reserved words, special symbols, and other tokens as specified in the Lexical Conventions of C- table. It detects the tokens by category, then filters the category to the specific token. The parser also detects multiline comments correctly, including the error checking for multiline comments. I also slightly added to the requested error reporting, for the purposes of showing errors due to non-printing characters.

### Could be improved

The printToken function currently only uses the actual token when the category is an end token or an error token. The language documentation specifies that whitespace should separate IDs, NUMs, and reserved words, but this is not currently verified.