

```

/*
 * Paul Heidelman
 * Case Western Reserve University
 * EECS 337 Fall 2009
 * Homework 2
 */

#####
FILES
#####

Makefile
    Used to compile the project with the 'make' command.
lex.l
    Contains the code for the lexical analyzer.
linked_list.c
    Contains the updated and modified code for the linked list problem in HW1.
linked_list_hw2.c
    Code that is used in the lexical analyzer to make a linked list for the symbol table.
linked_list_hw2.h
    Header file for linked list functions/structures.
readme
    This file.
test
    The test sample code that is run through the final lexical analyzer as a test.

#####
REGULAR EXPRESSIONS
#####

id:  [a-zA-Z_][a-zA-Z0-9_]*
basic: (int|float)
num:  [+]?[0-9]+
real:  [+]?[0-9]*\.[0-9]+(E[+]?[0-9]+)?

#####
SYMBOL TABLE
#####

Modified Linked List Ouptut:

paul[heidelmanp2]$ gcc linked_list.c
linked_list.c: In function 'main':
linked_list.c:173: warning: passing argument 1 of 'list_init' from incompatible pointer type
paul[heidelmanp2]$ ./a.out
Creating double_list...
found: 2.000000
found: 4.000000
found: 6.000000
not found
found: 4.000000

Creating string_list...
found: 2.000000
found: 6.000000
not found
not found
found: 6.000000

#####
LEXICAL ANALYZER
#####

To compile:
    Be in the directory "heidelmanp2"
    Run "make"

```

To run:

Be in the directory "heidelmanp2"

Run "./myprog test" where test is the file you wish to input.

If no file is specified, it defaults to stdin.

Output:

```
paul[heidelmanp2]$ make
flex lex.l
gcc -c lex.yy.c -o lex.yy.o
lex.l: In function 'yylex':
lex.l:84: warning: assignment makes pointer from integer without a cast
lex.l: In function 'installID':
lex.l:129: warning: return makes integer from pointer without a cast
lex.l:129: warning: function returns address of local variable
gcc -c linked_list_hw2.c -o linked_list_hw2.o
gcc lex.yy.o linked_list_hw2.o -o myprog
paul[heidelmanp2]$ ./myprog test
```

Using file test

Search: list is empty!

```
{ comment basic <id, i>; basic <id, j>; basic <id, v>; basic <id, x>; basic [<num, 100>] <id, a>;
while( true ) {
    do <id, i> = <id, i><num, +1>; while( <id, a>[<id, i>] < <id, v>);
    do <id, j> = <id, j><num, -1>; while( <id, a>[<id, j>] > <id, v>);
    if( <id, i> >= <id, j> ) break;
    <id, x> = <id, a>[<id, i>]; <id, a>[<id, i>] = <id, a>[<id, j>]; <id, a>[<id, j>] = <id, x>;
}
}
id's:
i
j
v
x
a
paul[heidelmanp2]$
```

Notes:

It was difficult to find an acceptable value for yylval and the return statements. Because they won't be needed until Homework 3, I am not too worried about them being missing/incorrect. I will correct them when I can actually use and test them in HW3.

list_search() outputs an error on the first run, because we search for something on an empty list.

#####

SOURCE CONTROL

#####

I set up a git repo on unfuddle.com

To Print:

```
design notes
how to compile and regex
linked_list.h file
linked_list test file
output from ^
lex file
output form lex file
text file
```