CS3300: Compiler Design **Odd Sem**

Lab Assignment #3

Lexing & Parsing Deadline: 19/08/2018, 11:55PM

Calculator

Build a simple calculator using **lex** and **yacc**.

- The calculator will contain a series of statements : assignments or expressions or variable declarations.
- There will be only one variable declared per line.
- Supported data types - **int**
- Supported binary operations:
  - Addition: '+'
  - Subtraction: '−'
  - Multiplication: '∗'
  - Integer Division: '/'
- Supported unary operations: '+' and '−'
- Expressions might contain parentheses as well.
- The precedence order for the above operators is Parentheses > Unary Operators(+, -) > Division > Multiplication > Subtraction = Addition
- All the operators are left-associative, except unary operators.
- Every statement ends with a semicolon.
- There will be only one statement per line. There will not be any empty lines.

Output format

The output should be as follows:

- If there is an invalid input (eg: parse error): Output - "Invalid Input" without quotes and terminate the program.
- If any of the variables are assigned without declaration or used without defining: Output: "Invalid Statement <line number>" without quotes and terminate the program.
- If variables are redeclared: Output: "Invalid Statement <line number>" without quotes and terminate the program.
- In case of divide by zero errors: Output: "Divide By Zero <line number>" without quotes and terminate the program.
- Where, <line number> is the line number of the corresponding invalid statement.
- Else, output the value of the expression encountered.
- All valid expression encountered before any of the above cases should be evaluated and corresponding outputs should be printed.

Examples

Example 1:
Input:

```
int a;
int b;
a = 1;
b = a + 3;
int c;
c = a + b;
a * c + b;
int d;
d = c / b;
d * (a + b);
```

Output:

```
9
5
```

Example 2:
Input:

```
int a;
int b;
a = 1;
b = a + 3;
c = a + b;
a * c + b;
int d;
d = c / b;
d * (a + b);
```

Output:

```
Invalid Statement
```

Reason: Variable c is not declared.

Example 3:
Input:

```
int a;
float b;
a = 1;
b = a + 3;
```

Output:

```
Invalid Input
```

Reason: Data type: float is not supported.

Example 4:
Input:

```
int a;
int b;
a = 4;
b = 2;
int c;
a + b - 6;
a + b - c;
a = a + 2;
```

Output:

```
0
Invalid Statement
```

Reason: Variable c is not initialized.

Example 5:
Input:

```
int a;
int b;
a = 4;
b = 2;
int c;
a + b = c;
```

Output:

```
0
Invalid Input
```

Reason: Parse Error.