Arithmetic expression compiler

Developed by Mojtaba Farokhi(9632443) and Mohsen Pakzad(9632445)

It was asked to convert an arithmetic expression into a specific form as following:

Input	Output
12925	Assign (OneTen_Two)Tou_NinHun_TwoTen_Fiv to t1 Print t1
2840*(106+5)	Assign OneHun_ZerTen_Six Plu Fiv to t1 Assign (Two)Tou_EigHun_FouTen_Zer Mul t1 to t2 Print t2

Interpretation is done based on below table:

Zer	0		Sev	7		Plu	+
One	1		Eig	8		Min	-
Two	2		Nin	9		Mlu	*
Thr	3		Ten	10		Div	/
Fou	4		Hun	100			
Fiv	5		Tou	1000			
Six	6						

There are 3 phases needed for the arithmetic expressions' compiler:

1. Lexical analysis
2. Syntax analysis
3. Intermediate code generation

The first phase which is Lexical one is responsible for tokenizing the input expression.
We had the following tokens for that:

- NUMBER
- PLUS
- MINUS
- MULTIPLY
- DIVIDE
- LEFT_PARENTHESES
- RIGHT_PARENTHESES
- NEW_LINE

To handle conversion from numerical to string version we define these regular definitions as following:

- digit
- one_digit_number
- two_digit_number
- three_digit_number
- four_digit_number
- five_digit_number
- six_digit_number
- +
- -
- *
- /
-)
- (
- \n

And for each of our digit numbers we wrote an action to convert that to the string version using our

```
void getDigitName(char *result, char digit)
```

function which converts a digit to its corresponding string format according to the above table.

The second phase is for parsing the tokens and turning them into the intermediate code.
Our grammars are as the following: (start symbol is stmts)

```
stmts → ε | stmt stmts
stmt → expr NEW_LINE | NUMBER NEW_LINE
expr → expr PLUS term |  expr MINUS term | term
term → term MULTIPLY factor | term DIVIDE factor | factor
factor → NUMBER | LEFT_PARENTHESES expr RIGHT_PARENTHESES
```

To simply using strings in c and running away from char* complexities we defined a struct called String in our string.h file and to let flex and yacc to work with that by setting the datatype to string in our shared.h file as following:

```
#define YYSTYPE struct String
```

There are also batch/bash files available in our project to simplify compiling and running in both Linux and Windows.