Programming Assignment 1

1. (10 points) Use lex or flex to implement a lexical analyzer for the miniC language.
   - See an attached file for the lexical rules in details.
   - You are requested to separate the C code and the Lex specification into distinct files.

Guideline:

1. You have to demonstrate your program in person and have the report in paper with you.
2. You will get 30% bonus if you succeed in demonstrating your program in class, while the report in paper still need to be handed-in in the due week. And, 30% penalty will be given for lateness. More precisely, if you get X in demonstration, and Y for the report:
   - Your score = X * 70% + Y * 30%
   - In-class demonstration and on-time report = X * 70% * 1.3 + Y * 30%
   - In-class demonstration but late report = X * 70% * 1.3 + Y * 30% * 0.7
   - On-time demonstration but late report = X * 70% + Y * 30% * 0.7
   - Late demonstration and on-time report = X * 70% * 0.7 + Y * 30%
   - Both late = (X * 70% + Y * 30%) * 0.7
3. Your report have to include the following elements:
   I. A cover page.
   II. The problem description.
   III. Highlight of the way you write the program.
   IV. The program listing.
   V. Test run results.
   VI. Discussion.

LEXICAL RULES

Integer

Sequence of digits denoting an integer number in the range -32768..32767. This should be stored in a two's-complement representation.
(Hint: The value range will be checked in the phase of semantic analysis.)

Identifiers

Sequence of letters, digits and underscores that may only be initiated with underscores or letters, no longer than 16 characters.
(Hint: The length of squence will be checked in the phase of semantic analysis.)

Strings

A string begins with a " and ends with a ". No new-line or " is allowed to appear in a string.
(Hint: Strings are only for the use in printf.)

Reserved words

```
break continue else if int return while printf
```

(Hint: Each reserved word is a token type.)

Special characters

```
+ - * / % ! ? : = , < > ( ) { } || && == " ;
```

(Hint: Each special character or sequence of special characters is a token type.)

Comments:

A comment begins with `//` and goes to the end of the line.

TEST PROGRAM

```
int ComputeFac(int num) {
   int num_aux;
      if (num < 1)
         num_aux = 1;
      else
         num_aux = num * ComputeFac(num-1);
      return num_aux;
}

int main() {
      printf("%d\n", ComputeFac(10));
}
```