Project 2

Lexical Analyzer (Scanner)

Due: October 16, 2022

Write a scanner/lexical analyzer for MYC++. You will use `ocamllex` to create the scanner.

Read more about ocamllex:

- **https://caml.inria.fr/pub/docs/manual-ocaml/lexyacc.html**
- **https://courses.softlab.ntua.gr/compilers/2015a/ocamllex-tutorial.pdf**

The scanner will read a .mycpp file and return a set of tokens or an error message. For example, given the following source file:

```
begin
      x := 1;
      while(x <= 10) /* 10 iterations */
      begin
            a := a + c;
            x := x + 1;
      end
end
```

the output should be as follows:

```
KEYWORDTOK: begin
IDENTTOK: x
OPTOK: :=
NUMTOK: 1
DELIMTOK: ;
KEYWORDTOK: while
DELIMTOK: (
IDENTTOK: x
OPTOK: <=
NUMTOK: 10
DELIMTOK: )
KEYWORDTOK: begin
IDENTTOK: a
OPTOK: :=
IDENTTOK: a
OPTOK: +
IDENTTOK: c
DELIMTOK: ;
IDENTTOK: x
OPTOK: :=
IDENTTOK: x
OPTOK: +
NUMTOK: 1
DELIMTOK: ;
KEYWORDTOK: end
KEYWORDTOK: end
```

Whitespace characters and comments are ignored, so they will not show up in the output.

If there is any unrecognized character(s), the lexical analyzer should display the appropriate error message with the line number. For example, for the given source code:

```
begin
      x := 1;
      while{x <= 10) /* 10 iterations */
      begin
            a := a + c;
            x := x + 1;
      end
end
```

The lexical analyzer should display the following (you do not have to show any recognized tokens):

```
KEYWORDTOK: begin
IDENTTOK: x
OPTOK: :=
NUMTOK: 1
DELIMTOK: ;
KEYWORDTOK: while
*Error* Unrecognized character (line 3): {
```

Here is another example:

```
begin
      x := 1;
      while(x <= 10) /* 10 iterations */
      begin
            a := a + c;
            x := x + 11.;
      end
end
```

The output should be (Notice, there is an ‘s’ in ‘characters’ in this error message):

```
KEYWORDTOK: begin
IDENTTOK: x
OPTOK: :=
NUMTOK: 1
DELIMTOK: ;
KEYWORDTOK: while
DELIMTOK: (
IDENTTOK: x
OPTOK: <=
NUMTOK: 10
DELIMTOK: )
KEYWORDTOK: begin
IDENTTOK: a
OPTOK: :=
IDENTTOK: a
```

```
OPTOK: +
IDENTTOK: c
DELIMTOK: ;
IDENTTOK: x
OPTOK: :=
IDENTTOK: x
OPTOK: +
*Error* Unrecognized characters (line 6): 11.
```

Here is a description of the syntax of our MYC++ language:

Whitespaces and Comments:

Newline, space, and tab characters are supposed to be deleted from the source file. You may need to keep line numbers in case of error messages. The Lexical Analyzer skips over these characters, but does not return any token for them. Comments are contained between the characters /* and */ , or they begin with // for one-line comments. Comments of course cannot be nested.

Operators and Delimiters:

Operators of MYC++ are: + - * / := == <> < <= >= >

Delimiters of MYC++ are: ; : ( ) [ ]

Reserved words:

The keywords of MYC++ are:
begin end if then else goto while label do integer real string

Identifiers:

Identifiers begin with a letter, followed by any number of letters, digits, and/or underscore. The length of MYC++ identifiers should not exceed 25 characters. Longer identifiers should not be accepted.

Note that ***MYC++ doesn't accept any capitalized letters except if there are part of strings.***

Numbers:

Unsigned Numbers must begin with a digit; if there is a decimal point, it must be followed by at least one digit.

For simplicity, don't consider any numeric overflow or underflow. If there is an error, your program should display an error message accordingly. For example, the number 123. is not accepted floating point number as there is no digits after the decimal point.

Strings:

A string contains a collection of characters surrounded by double quotes.

```
       string msg = "Hello"
```

Testing:

You may test your program using the attached files `test1.mycpp, test2.mycpp`, `test3.mycpp` and `test4.mycpp`. Your scanner will be tested with a larger number of test cases.

Submit only the ocamllex .mll file