Author: Spera Danilo (xsperad00)

The Makefile is present in the project, to run it on Merlin you need to:

1. enter in the project directory
2. run the command **make**

After this there are two option to run the project:

1. run the command **./main** to run all the tests present on the file test.txt (I inserted some of the tests I made during the development), it will run 36 tests of which the first 4 are the examples provided.
2. run the command
   ./main "input string1" "input string2" … "input stringN", in this case the project will not read the file with the test and will instead read the input strings. Some example can be:
   - **./main "add(1,2)"**
   - **./main "sub(1,3)" "mul(2,9)"**

Implementation details:

For the project I used the C language, the libraries I used are:

- stdio.h
- stdlib.h → for the dynamic allocation of memory
- string.h → for some string function like strlen
- ctype.h → for the function isspace to check if a character is a withespace

I manually implemented all the structures needed (stack, tree, token).

The project is subdivided in more file, the most important are:

- **Scanner.c:** It contains the implementation of the scanner, to create it I was inspired by the algorithms provided during the lesson on PDF 5.
- **Parser.c**: Ic contains the implementation of the scanner, to create it I used a CFG and a precedence table, I inserted the precedence table inside a matrix.
- **main.c:** The main calls the function scan and then if no error occurs cals the function parse on the output of the function scan, like I said before the main can be run in two different ways, reading the test file or not.

Another file present is the test.txt, it can be updated easily to insert new test cases.

After all the test is possible to write the command **make clean**.