Laboratory 09: LLVM parser

Grammar

For this laboratory we will implement the following grammar:

```
Z      ::= E_AS.
E_AS   ::= E_MDR ('+' | '-' E_MDR)*.
E_MDR  ::= T ('*' | '/' | '%' T)*.
T      ::= i | '(' E_AS ')'.
```

E_AS = **E**xpression with **A**dditions and **S**ubtractions

E_MDR = **E**xpression with **M**ultiplications, **D**ivisions and **R**emainders

Code generation

The `CodeGenTopLevel` function will now display the LLVM IR code from the given AST and it will also create a new file, **output.ll** that will also contain the generated IR code. This will be then used by an interpreter program to check the output of the code.

Running the code

In order to run the code we will run first the `make` command. This time the make command will not automatically run the executable file. For this, we will use a special command that will run the executable file generated then will run the intermediate representation generated by the executable file and it will print the result on the screen.

The command for running the executable and printing the result on the screen is

```
./main; lli-17 output.ll; echo "Result is: $?"
```

This command can also be found as a comment in the makefile.