* **Introduction:**

- The idea behind this project is to design our own compiler based on Lex & Yacc tools. First we define our BNF grammars then, using these grammars we create our Lex & Yacc files. Lex & Yacc are tools that are used in building up compilers. Lex: is a tool for writing lexical analyzers, Yacc: is a tool for constructing parsers.

Lex reads our input character sets and converts them into tokens that we feed to the Yacc file in order to parse them and do the wanted actions. Since we are designing our own language we can choose the relevant action we wish.

**Solution steps:**

1. Setting up BNF grammars with our imagination and functions that may used to ease the job.
2. Creating Lex file and specifying the patterns (regex) that are going to be passed to the Yacc file as tokens.
3. Constructing Yacc file by defining the tokens, the data types, the grammar rules, and the C functions.
4. Code debugging time.
5. Error checking & handling through the right functions e.g: yyerror() .. etc.

* **BNF grammar:**
* **Prerequisites:**
* **Extended Features:**

Functions:

* 1. print : prints the string value of the object passed as an argument. Prints th string literal value passed as argument.
  2. list\_contents : lists the contents (files, directories) of the passed directory argument.
  3. get\_size : returns the size of the (file, directory) passed as argument.
  4. create : creates a new (file, directory) in the current directory.
  5. copy : makes a copy of the argument passed (file, directory) in the current directory.
  6. compare : compares two given files to check for equality, returns false or true;
  7. connect : connects to the given server as an argument.
  8. delete : deletes the given (file, directory).
  9. rename : renames a given (file, directory) to a name passed as an argument.
  10. move : moves a given (file, directory) to the given new path.
  11. sort : sorts the given directory’s files according to name ordering.
  12. filter: returns all files of the specified type from the given directory.
  13. back\_up: makes a back-up copy of your local files to the cloud.
  14. synchronize: performs a synchronization between local files and files on the cloud to update the cloud files with any new changes.
  15. search: search in the given directory for the given file.
  16. cd : changes the current directory to the given one.
  17. download: downloads the given file using it’s URL.
  18. upload: uploads the specified (file, directory) to the cloud.
  19. open: opens the give file.
  20. properties: returns the properties of the given (file, directory).
  21. move\_back: moves one directory back to the previously opened one.
  22. move\_forward: moves one directory forward, in case of moving back then the need to move forward again.