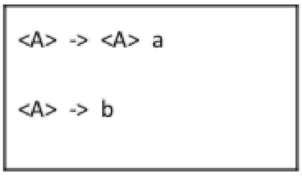
PROGRAMMING LANGUAGES CSE-351
TERM PROJECT REPORT

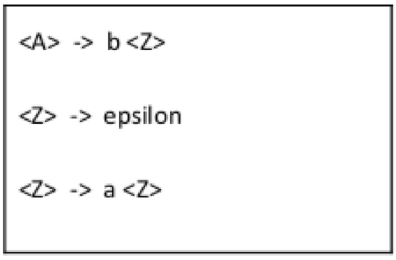
This Project is about excepting any BNF grammar and eliminate left recursions.



```
<A> -> <A> a

<A> -> b
```

---->



```
<A> -> b <Z>

<Z> -> epsilon

<Z> -> a <Z>
```

I did this project in 4 phases:

1) Writing lex and general structure of yacc files:

- I wrote lex file which can parse BNF rule. It returns each NONTERMINAL, TERMINAL, ASSIGN, EPSILON, NEWLINE seperatly.
- I wrote BNF rule for accepting rules in given form. Then i wrote C++ functions for file operations:

```
void openFile(string);
void writeLinetoFile(string);
void closeFile();
```

2) Accepting BNF rules with no left recursion:

- I wrote, BNF rules with no left recursion directly into output file.
- I wrote C++ functions to write directly into output file.

```
void writeDirectly(string)
```

3) Accepting BNF rules with 1 left recursion:

- I wrote a class for left recursions.And field is like this:

```
class leftRecursion
{
public:
      string ident;
      string rhsAfter;
      vector<string> otherLines;
      vector<string> otherLRrhsAfter;
};
```

- Then i wrote C++ functions to accept 1 left recursion and save it:

```
void addLR(string)
int isLRExist(string)
void eliminateLR()
```

4) Accepting BNF rules with more than 1 left recursion:

- Then i rearranged my functions to accept more than 1 left recursions and eliminate them.