Assignment-4

Due : 1^{st} February, 2023

The assignment for today has two parts:
Part 1: Design a syntax analyzer using Bison which will accept the grammar given below:
Non-terminals : $\{E\}$
Terminals : $\{+ * id\}$
Production rules :
$S : ID = E$
$E : E + E$
$E : E * E$
$E : id$
Start symbol : E
Observe the output once you compile the grammar. Observe the output if you give / operator in the input stream. Change the grammar so that the problem (if any) is alleviated.

Part 2: Design a syntax analyzer using Bison which will accept an input code written in a toy programming language. For this assignment, the syntax analyzer accepts some declaration statements written in a function. You have already designed a lexical analyzer which tokenizes the inputs in assignment 2. Please return appropriate tokens to the parsers for the input lexemes. Some example tokens, the symbols and the context-free grammar are given below.

Terminals :

Expression	**Symbol in grammar**
abc	ID
12	$INTEGER\_CONSTANT$
1.2	$FLOAT\_CONSTANT$
int	INT
float	$FLOAT$
;	$SEMICOLON$
,	$COMMA$
=	$ASSIGN$
if	IF
else	$ELSE$
&&	AND
||	OR
!	NOT
==	EQ
>=	GE
<=	LE
<	LT
>	GT
!=	NE
while	$WHILE$
return	RETURN

Non-terminals :

prog funcDef type argList declList stmtList argList arg type decl varList stmtList

Start symbol : prog

Production Rules :

$prog \rightarrow funcDef$
$funcDef \rightarrow type\ id$ '(' $argList$ ')' '{' $declList\ stmtList$ '}'
$argList \rightarrow arg$ ',' $arg \mid epsilon$;
$arg \rightarrow type\ ID$;
$declList \rightarrow declList\ SEMICOLON\ decl | epsilon$;
$decl \rightarrow type\ varList$;
$varList \rightarrow ID\ COMMA\ varList | ID$;
$type \rightarrow INT | FLOAT$;
$stmtList \rightarrow epsilon$;

A sample accepted input is given below.

```
int main()
{
    int a, b;
    float d;
    char  c;
}
```