Exp-3(Project)

Group member-IT004,IT005,IT006

PROJECT TITLE

1. X++: used to indicate two increments to “X”
2. X--: used to indicate two decrements to “X” integer variable.
3. X**: used to indicate X variable should be doubled.
4. X=+Y12: Does the addition of X and Y12 and stores value back in “X”.
5. X=-Y: Does the subtraction of X and Y and stores value back in “X”.
6. X=<integer or float value>: To assign some value to the variable
7. <datatype> <variable name list separated by semicolon>: To define the variables with a particular datatype.

● Variables can be of type: “simple” or “float”
● Variable names should start with capital alphabet and can be followed by maximum 2 digits.
● Sentence end is indicated by “:”

- Find the RE and regular definition of your language and write it in flex

Regular Expression:

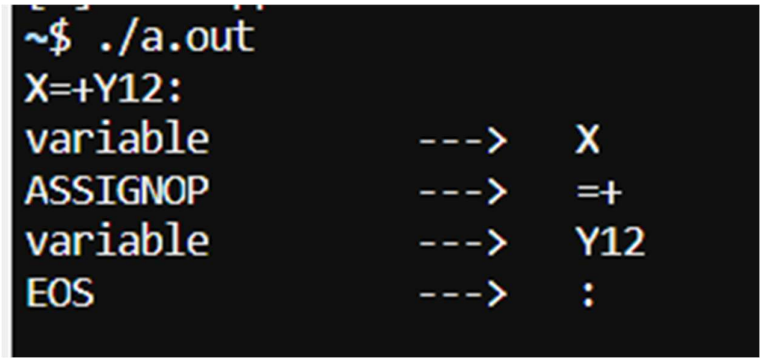
cletter → [A-Z]
letter → [A-Za-z]
digit → [0-9]
op → ++ | -- | **
assignop → =+ | =- | =
simple → simple
float → float
id → cletter(digit?)(digit?)(letter)*
num → digit+((.)(digit+))?
ws → (blank | tab | newline) +

flex program:

```
%{

#include <stdio.h>

%}

printf("\n\n");
```

```
%%
";"                     printf("VS          --->\t%s\n",yytext);
":"                     printf("EOS          --->\t%s\n",yytext);
"simple"|"float"            printf("datatype      --->\t%s\n",yytext);
"++"|"--"|"**"              printf("OP          --->\t%s\n",yytext);
"="|"=+"|"=-"               printf("ASSIGNOP      --->\t%s\n",yytext);
[A-Z][[0-9]?[0-9]?]?[a-z|A-Z]*  printf("variable      --->\t%s\n",yytext);
[+-]?([0-9]+([.][0-9]+)?)     printf("NUM          --->\t%s\n",yytext);
[ \t\r]+          /*NOP*/    print("");
%%
int yywrap(){
return 1;
}
int main()
{

yylex();
return 0;
}
```

OUTPUT-:



```
~$ ./a.out
A12abc=20.21:
variable          --->   A12abc
ASSIGNOP          --->   =
NUM               --->   20.21
EOS               --->   :
```

```
X++:
variable          --->   X
OP                --->   ++
EOS               --->   :
```

```
~$ flex lt1.l
~$ cc lex.yy.c
~$ ./a.out
simple X;Y12a;Z=20:
datatype          --->   simple
variable          --->   X
VS                --->   ;
variable          --->   Y12a
VS                --->   ;
variable          --->   Z
ASSIGNOP          --->   =
NUM               --->   20
EOS               --->   :
```