Program Number :- 01

1BY18CS177 6 C

Tarush J. Reddy

Question:

Write a Lex Program to recognise valid automatic expression. Id's are only no. & operators could be + and * count the identifiers & print them.

Answer

```
%{
#include<stdio.h>
int op=0, id=0, flag=0;
%}
%%
[0-9][0-9]* {id++; printf("\n Identifier Integers:")
ECHO;}
[\+\*] {op++; printf("\n operator :"); ECHO }
";" {flag=1;}
.\n {return 1;}
%%
int main()
{
printf("enter the expression:\n");
yylex();
if((op+1)==id && flag==0)
{ printf("\n identifiers are: %d\n", id);
printf("operators are: %d\n",
}
```

```
else
{
printf("\n Expression is invalid.\n");
}
}

int yywrap()
{
return 1;
}
```

Output:-

```
$ lex 12.l
$ gcc lex.yy.c -ll
$ ./a.out

enter the expression
2+3
identifiers integers : 2
operator : +
identifier integers : 3
identifiers are : 2
operators are : 1
expression is valid.
```

Program Number :- 018

Question

Answer.

16.l

```
%{
#include "y.tab.h"
extern yylval;
%}
%%
[0-9]+ {yylval = atoi(yytext); return num;}
[\+\-\*\/] {return yytext[0];}
[>] {return yytext[0];}
[<] {return yytext[0];}
. {;}
\n {return 0;}
%%
```

16.y

```
%{
#include<stdio.h>
#include<stdlib.h>
%}
%token num
%left '+' '-'
%left '*' '/'
%%
input: exp {printf("result is %d\n", $$); exit(0);}
exp: exp '+' exp {$$ = $1 + $3}
```

```
| exp '-' exp {$$ = $1 - $3}.
| exp '*' exp {$$ = $1 * $3}.
| exp '/' exp {if ($3 == 0) {printf ("Divide by 0 invalid
expression \n"); exit(0);} else $$ = $1 / $3 ; }
| '(' exp ')' {$$ = $2 ;}.
| num {$$ = $1}
%%
int yyerror()
{
printf ("error invalid expression \n"); exit(0)
}
int main()
{
printf ("Enter an expression : \n"); yyparse();
}
```

Output:

```
$ lex 16.l
$ yacc -d 16.y
$ gcc lex.yy.c y.tab.c -w
$ ./a.out
enter the expression
3 + 1
result is 4
$ ./a.out
enter an expression
4 - 7
result is 16
```

```
student@student-virtual-machine:~$ lex sample.l
student@student-virtual-machine:~$ gcc lex.yy.c -ll
student@student-virtual-machine:~$ ./a.out
Enter the expression:
2+3

Identifier/integers:2
Operartor:+
Identifier/integers:3
identifiers are: 2
operators are:1
Expression is valid
student@student-virtual-machine:~$ ./a.out
Enter the expression:
2++

Identifier/integers:2
Operartor:+
Operartor:+
Expression is invalid
```

```
student@student-virtual-machine:~$ lex 1b.l
student@student-virtual-machine:~$ yacc -d 1b.y
student@student-virtual-machine:~$ gcc lex.yy.c y.tab.c -ll
student@student-virtual-machine:~$ ./a.out
Enter an Expression:
3+1
result is 4
student@student-virtual-machine:~$ ./a.out
Enter an Expression:
4-7
result is -3
student@student-virtual-machine:~$ ./a.out
Enter an Expression:
8*2
result is 16
student@student-virtual-machine:~$ ./a.out
Enter an Expression:
5/0
Divide by Zero. Invalid expression.
student@student-virtual-machine:~$ ./a.out
Enter an Expression:
9/3
result is 3
```