Name: Gokulnath M Prabhu
Class: CS7B
Roll No: 21

Lab Cycle 2 - Experiment 6

Generate a YACC specification to recognize a valid arithmetic expression that uses operators +, – *,/ and parenthesis

Code:

arith_exp.l :

```
%{
/* Definition section*/
#include "y.tab.h"
extern int yylval;
%}
%%
[0-9]+ {
yylval = atoi(yytext);
return NUMBER;
}
[a-zA-Z]+ { return ID; }
[ \t]+ ; /*For skipping whitespaces*/
\n { return 0; }
. { return yytext[0]; }
%%
```

arith_exp.y :

```
%{
/* Definition section */
#include <stdio.h>
int yylex();
int yyerror();
%}
%token NUMBER ID
// setting the precedence
// and associativity of operators
%left '+' '-'
%left '*' '/'
/* Rule Section */
%%
E : T {
printf("Result = %d\n", $$);
return 0;
}
T :
T '+' T { $$ = $1 + $3; }
```

```
| T '-' T { $$ = $1 - $3; }
| T '*' T { $$ = $1 * $3; }
| T '/' T { $$ = $1 / $3; }
| '-' NUMBER { $$ = -$2; }
| '-' ID { $$ = -$2; }
| '(' T ')' { $$ = $2; }
| NUMBER { $$ = $1; }
| ID { $$ = $1; }
%%

int main() {
   printf("Enter the expression\n");
   yyparse();
}
/* For printing error messages */
int yyerror(char* s) {
   printf("\nExpression is invalid\n");
}
```

Output:

```
gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Arithmetic_Exp$ lex arith_exp.l
gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Arithmetic_Exp$ yacc -d arith_exp.y
gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Arithmetic_Exp$ cc lex.yy.c y.tab.c -ll
gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Arithmetic_Exp$ ./a.out
Enter the expression
3*(5-3)+2
Result = 8
gokz1119@gokz-Lenovo:/media/gokz1119/New Volume/S7/CD Lab/Arithmetic_Exp$
```