Details :

- Name : P K Navin Shrinivas
- Section : D
- SRN : PES2UG20CS237

Lexer :

```
%{
    #include "quad_generation.c"
    #include <stdio.h>
    #include <stdlib.h>
    #include <string.h>

#define YYSTYPE char*

void yyerror(char* s);
// error handling function
    int yylex();
// declare the function performing lexical analysis
    extern int yylineno;
// track the line number

FILE* icg_quad_file;
    int temp_no = 1;
%}


%token T_ID T_NUM

/* specify start symbol */
%start START
```

```
%%
START : ASSGN   {
      printf("Valid syntax\n");
                YYACCEPT;
// If program fits the grammar, syntax is valid
            }

/* Grammar for assignment */
ASSGN : T_ID '=' E      {
      quad_code_gen($3," ","=",$1);
      }
   ;

/* Expression Grammar */
E : E '+' T     {
  char* temp = new_temp();
  quad_code_gen($1,$3,"+",temp);
  $$ = temp;
  }
   | E '-' T    {
  char* temp = new_temp();
  quad_code_gen($1,$3,"-",temp);
  $$ = temp;
     }
   | T
   ;


T : T '*' F     {
  char* temp = new_temp();
  quad_code_gen($1,$3,"*",temp);
  $$ = temp;
}
  | T '/' F     {
  char* temp = new_temp();
  quad_code_gen($1,$3,"/",temp);
  $$ = temp;
```

```
}
    | F
    ;

F : '(' E ')'    {$$ = $2;}
   | T_ID                  {$$ = $1;}
    | T_NUM       {$$=$1;}
    ;

%%


/* error handling function */
void yyerror(char* s)
{
    printf("Error :%s at %d \n",s,yylineno);
}


/* main function - calls the yyparse() function which
will in turn drive yylex() as well */
int main(int argc, char* argv[])
{
    icg_quad_file = fopen("icg_quad.txt","w");
    yyparse();
    fclose(icg_quad_file);
    return 0;
}
```

Quad generation.c:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "quad_generation.h"

void quad_code_gen(char* a, char* b, char* op, char* c)
```

```
{
        //use fprintf to output the quadruple code to
icg_quad_file
   fprintf(icg_quad_file,"%s,%s,%s,%s \n",op,a,b,c);
}

char* new_temp()        //returns a pointer to a new
temporary
{
        char* temp = (char*)malloc(sizeof(char)*4);
        sprintf(temp, "t%d", temp_no);
        ++temp_no;
        return temp;
}
```

Outputs :

```
➜ lab6 git:(main) ×./a.out < test_input_1.c
Valid syntax
➜ lab6 git:(main) ×cat icg_quad.txt
/,9,2,t1
+,t1,a,t2
-,t2,b,t3
=,t3,,x
➜ lab6 git:(main) ×./a.out < test_input_2.c
Valid syntax
➜ lab6 git:(main) ×cat icg_quad.txt
/,c,6.7,t1
+,t1,12.45,t2
*,a,1234.0,t3
-,t2,t3,t4
=,t4,,b
➜ lab6 git:(main) ×
```