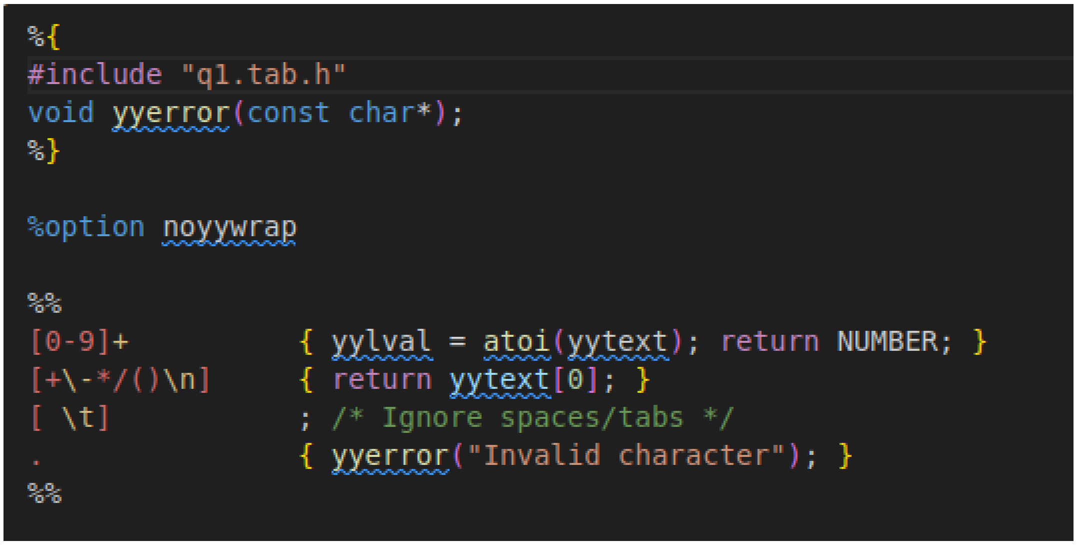
CSE 358 - ASSIGNMENT 7

220002018 - Arnav Jain

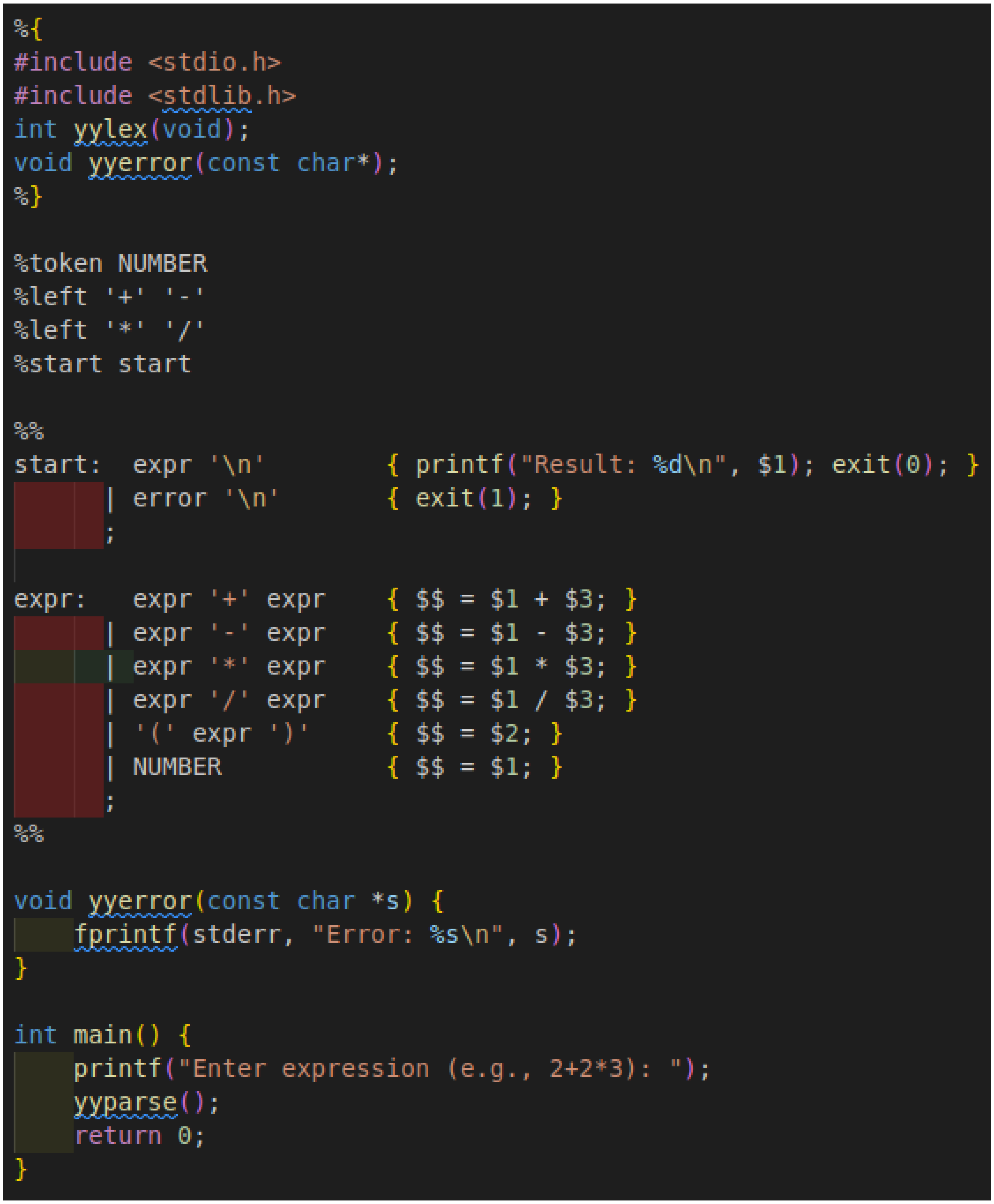
Question 1

Lex:



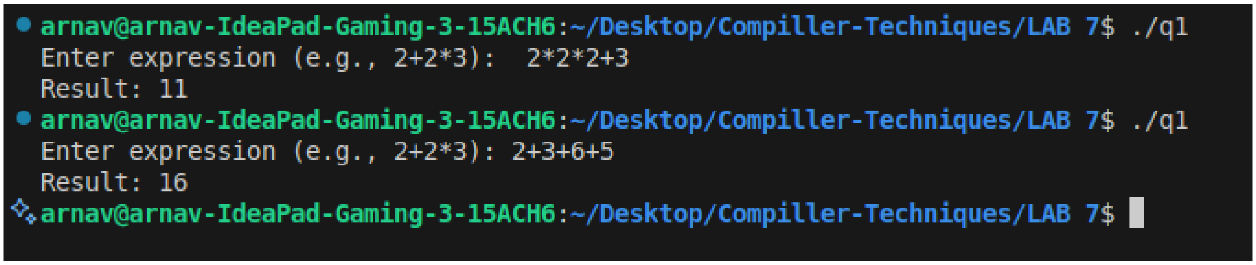
```
%{
#include "q1.tab.h"
void yyerror(const char*);
%}

%option noyywrap

%%
[0-9]+          { yylval = atoi(yytext); return NUMBER; }
[+\-*/()\n]     { return yytext[0]; }
[ \t]           ; /* Ignore spaces/tabs */
.               { yyerror("Invalid character"); }
%%
```

Yacc:



```
%{
#include <stdio.h>
#include <stdlib.h>
int yylex(void);
void yyerror(const char*);
%}

%token NUMBER
%left '+' '-'
%left '*' '/'
%start start

%%
start:  expr '\n'        { printf("Result: %d\n", $1); exit(0); }
      | error '\n'       { exit(1); }
      ;

expr:   expr '+' expr    { $$ = $1 + $3; }
      | expr '-' expr    { $$ = $1 - $3; }
      | expr '*' expr    { $$ = $1 * $3; }
      | expr '/' expr    { $$ = $1 / $3; }
      | '(' expr ')'     { $$ = $2; }
      | NUMBER           { $$ = $1; }
      ;
%%

void yyerror(const char *s) {
    fprintf(stderr, "Error: %s\n", s);
}

int main() {
    printf("Enter expression (e.g., 2+2*3): ");
    yyparse();
    return 0;
}
```

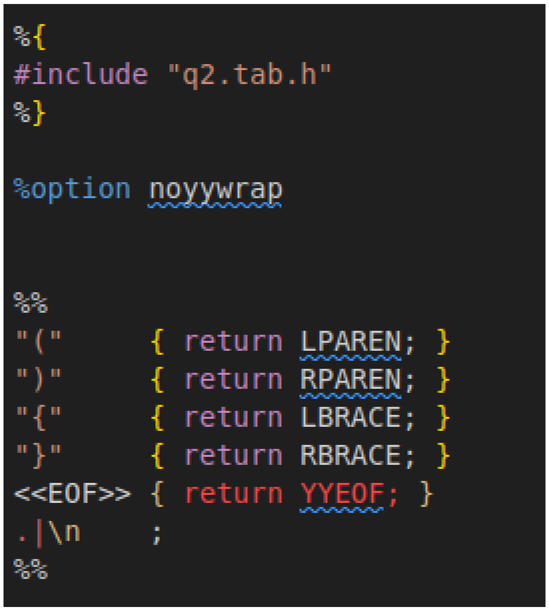
Output:



```
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$ ./q1
  Enter expression (e.g., 2+2*3):  2*2*2+3
  Result: 11
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$ ./q1
  Enter expression (e.g., 2+2*3): 2+3+6+5
  Result: 16
  arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$ 
```

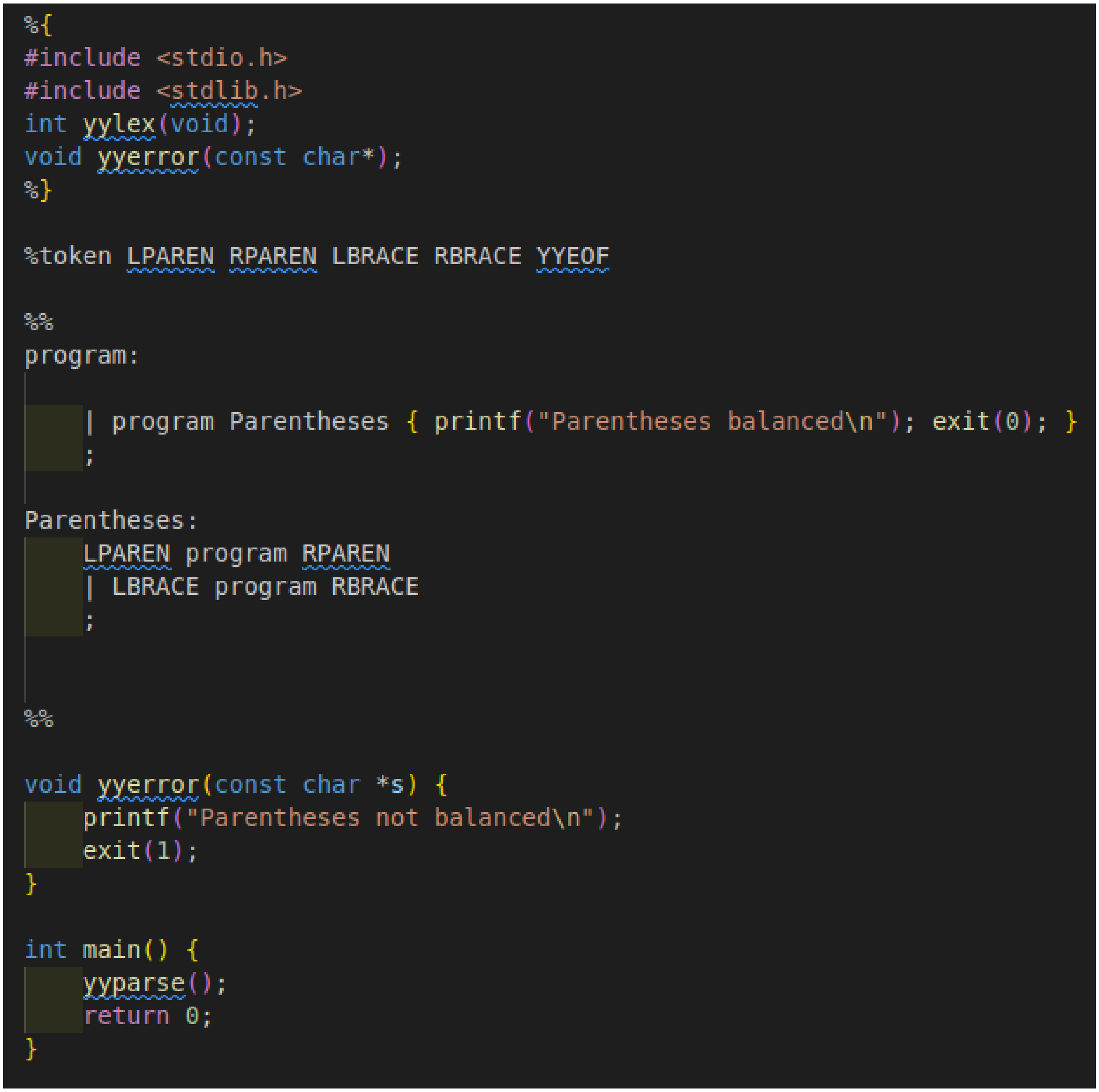
Question 2

Lex:



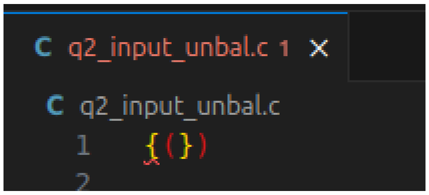
```
%{
#include "q2.tab.h"
%}

%option noyywrap


%%
"("     { return LPAREN; }
")"     { return RPAREN; }
"{"     { return LBRACE; }
"}"     { return RBRACE; }
<<EOF>> { return YYEOF; }
.|\n    ;
%%
```

Yacc:

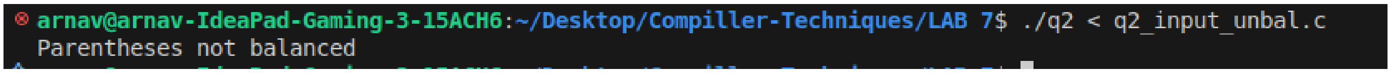


```
%{
#include <stdio.h>
#include <stdlib.h>
int yylex(void);
void yyerror(const char*);
%}

%token LPAREN RPAREN LBRACE RBRACE YYEOF

%%
program:

    | program Parentheses { printf("Parentheses balanced\n"); exit(0); }
    ;

Parentheses:
    LPAREN program RPAREN
    | LBRACE program RBRACE
    ;


%%

void yyerror(const char *s) {
    printf("Parentheses not balanced\n");
    exit(1);
}

int main() {
    yyparse();
    return 0;
}
```

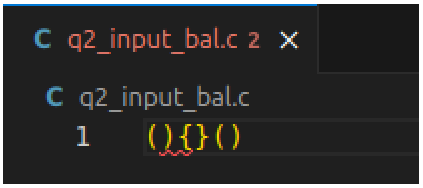
Output:



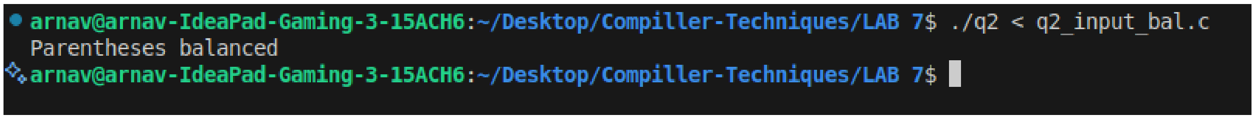
```
{(})
```



```
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$ ./q2 < q2_input_unbal.c
Parentheses not balanced
```



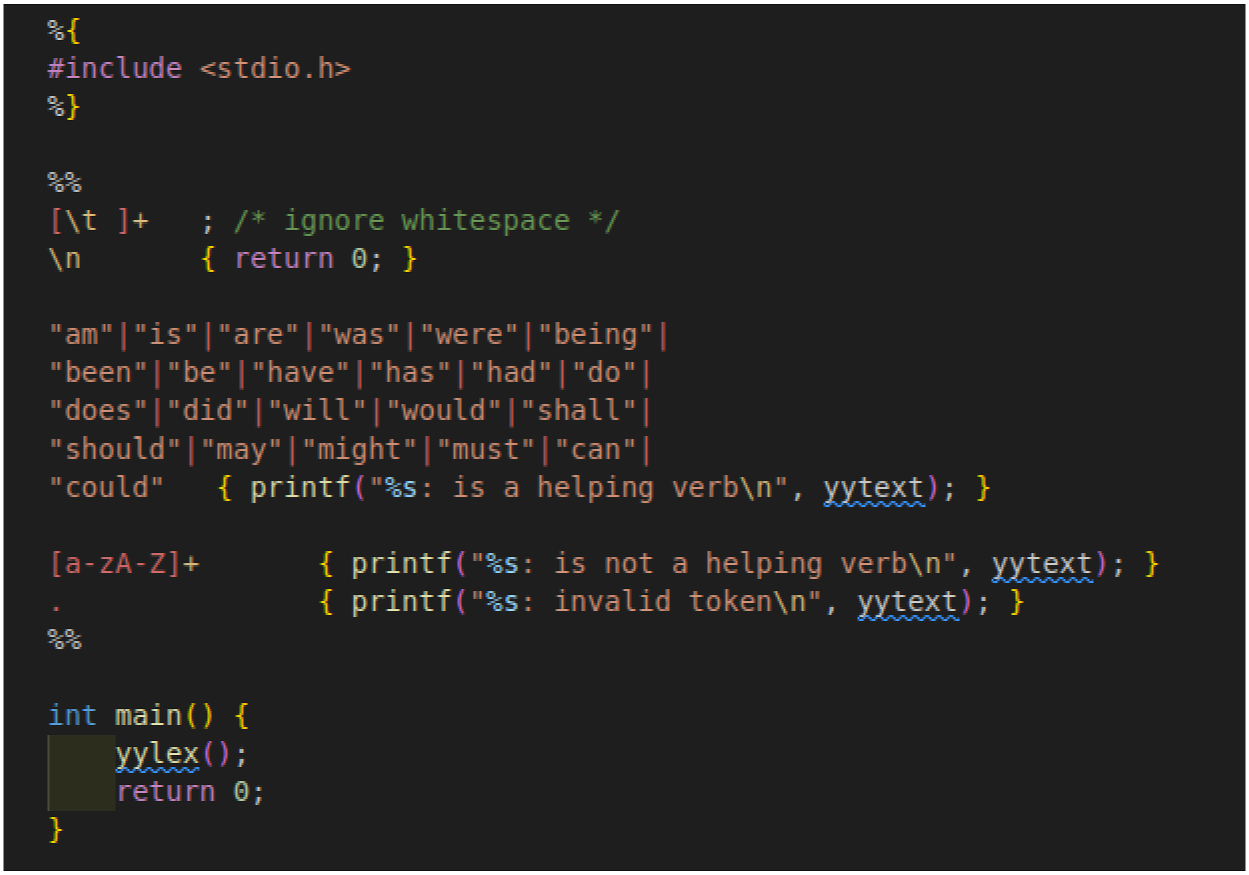
```
(){}()
```



```
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$ ./q2 < q2_input_bal.c
Parentheses balanced
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$
```

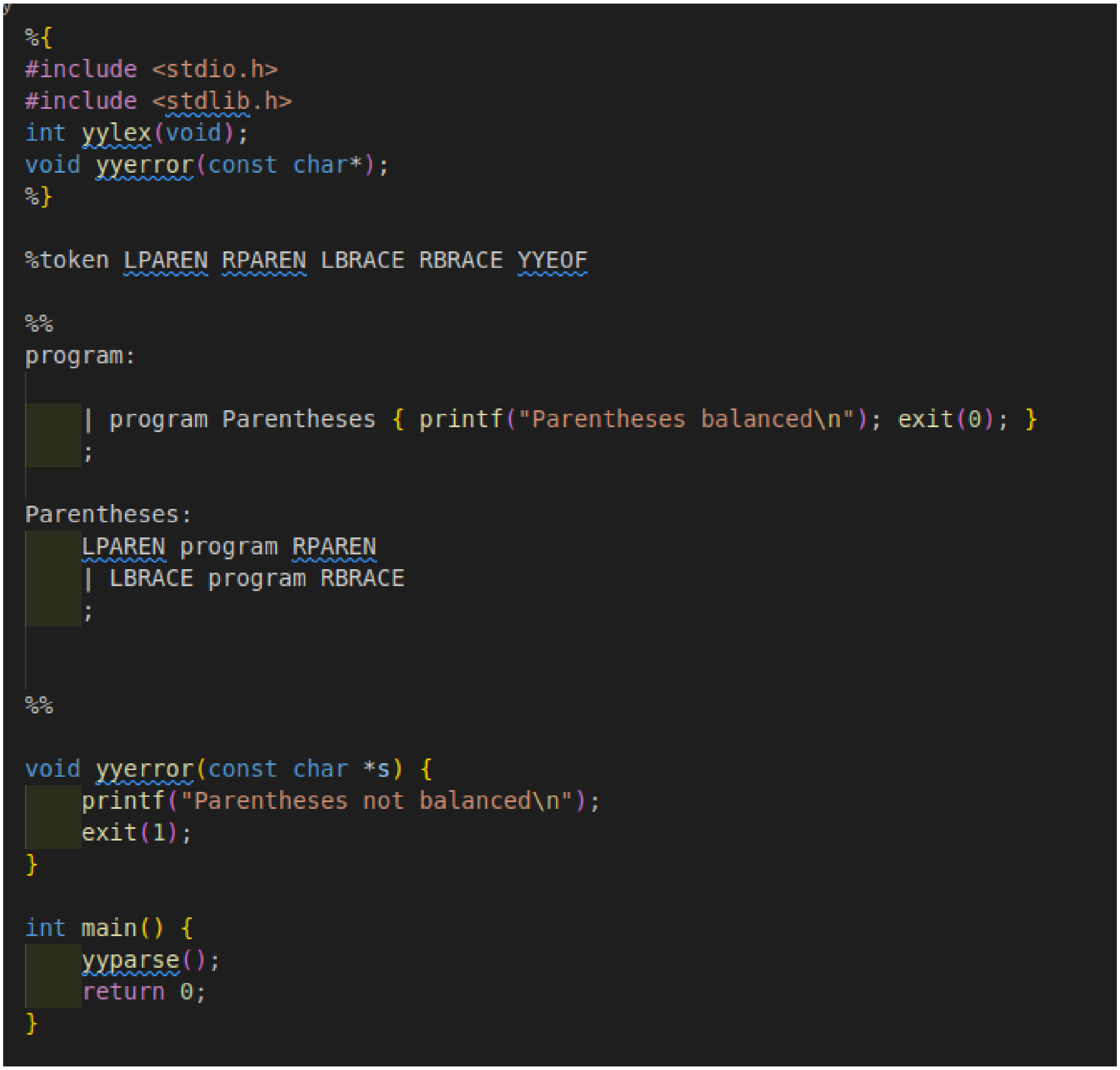
Question 3

Lex:



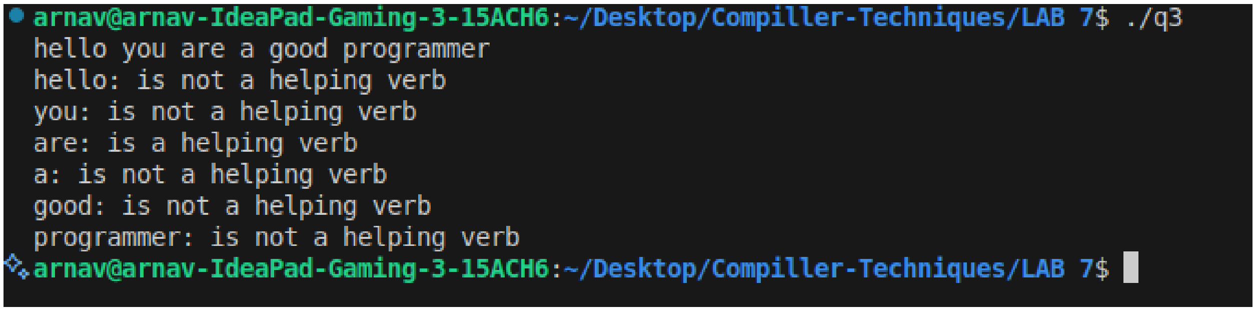
```
%{
#include <stdio.h>
%}

%%
[\t ]+   ; /* ignore whitespace */
\n       { return 0; }

"am"|"is"|"are"|"was"|"were"|"being"|
"been"|"be"|"have"|"has"|"had"|"do"|
"does"|"did"|"will"|"would"|"shall"|
"should"|"may"|"might"|"must"|"can"|
"could"   { printf("%s: is a helping verb\n", yytext); }

[a-zA-Z]+       { printf("%s: is not a helping verb\n", yytext); }
.               { printf("%s: invalid token\n", yytext); }
%%

int main() {
    yylex();
    return 0;
}
```

Yacc:



```
%{
#include <stdio.h>
#include <stdlib.h>
int yylex(void);
void yyerror(const char*);
%}

%token LPAREN RPAREN LBRACE RBRACE YYEOF

%%
program:

    | program Parentheses { printf("Parentheses balanced\n"); exit(0); }
    ;

Parentheses:
    LPAREN program RPAREN
    | LBRACE program RBRACE
    ;


%%

void yyerror(const char *s) {
    printf("Parentheses not balanced\n");
    exit(1);
}

int main() {
    yyparse();
    return 0;
}
```

Output:



```
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$ ./q3
hello you are a good programmer
hello: is not a helping verb
you: is not a helping verb
are: is a helping verb
a: is not a helping verb
good: is not a helping verb
programmer: is not a helping verb
arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 7$ 
```

Code (GitHub)

https://github.com/arnavjain2710/Compiller-Techniques/tree/main/LAB%207