# CSE 358 - ASSIGNMENT 6

*Arnav Jain - 220002018*

## Question 1

LEX Code:

```
 1  %{
 2  #include "q1.tab.h"          // Required for tokens
 3  extern void yyerror(const char *s);  // Explicit declaration
 4  %}
 5
 6  %%
 7  A    { return A; }
 8  B    { return B; }
 9  \n   { return 0; }
10  .    { yyerror("Invalid character detected"); }
11  %%
12
13  int yywrap() { return 1; }
```

YACC Code:

```
q1.y                    ×

q1.y
 1    %{
 2    #include <stdio.h>
 3    int yylex(void);                 // Explicit declaration
 4    void yyerror(const char *s);     // Const-correct prototype
 5    %}
 6
 7    %token A B
 8
 9    %%
10    start: a_seq b_seq ;
11
12    a_seq: /* empty */ | a_seq A ;
13    b_seq: /* empty */ | b_seq B ;
14    %%
15
16    void yyerror(const char *s) {
17        printf("ERROR: %s\n", s);
18    }
19
20    int main() {
21        yyparse();
22        return 0;
23    }
24
```

Output:

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   POLYGLOT NOTEBOOK   GITLENS   SPELL CHECKER   COMMENTS

● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 6$ ./q1
  ABBB
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 6$ ./q1
  AAAAABBBBBB
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 6$ ./q1
  ABBAA
  ERROR: syntax error
● arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 6$ ./q1
  ABCCC
  ERROR: Invalid character detected
  ERROR: Invalid character detected
  ERROR: Invalid character detected
⟲ arnav@arnav-IdeaPad-Gaming-3-15ACH6:~/Desktop/Compiller-Techniques/LAB 6$
```

1

# Question 2

LEX Code:

```
1    %{
2    #include "q2.tab.h"
3    extern int operands, operators;
4    extern int has_error;
5    extern void yyerror(const char *s);
6    %}
7
8    %%
9    [a-zA-Z_][a-zA-Z0-9_]* { operands++; return ID; }
10   [0-9]+(\.[0-9]*)?       { operands++; return NUMBER; }
11   "+"                     { operators++; return ADD; }
12   "-"                     { operators++; return SUB; }
13   "*"                     { operators++; return MUL; }
14   "/"                     { operators++; return DIV; }
15   \n                      { return END; }
16   [() \t]                ;
17   .                       { yyerror("Invalid token"); }
18   %%
19
20   int yywrap() { return 1; }
```

YACC Code:

```
q2.y                    ×

q2.y
  1   %{
  2   #include <stdio.h>
  3   int operands = 0, operators = 0;
  4   int has_error = 0;
  5   int yylex(void);
  6   void yyerror(const char *s);
  7   %}
  8
  9   %token ID NUMBER END
 10   %token ADD SUB MUL DIV  // Individual operator tokens
 11   %left ADD SUB
 12   %left MUL DIV
 13   %left UMINUS
 14
 15   %%
 16   start: expr END {
 17       if (!has_error) {
 18           printf("Valid Expression\nOperands: %d\nOperators: %d\n", operands, operators);
 19       }
 20   };
 21
 22   expr: expr ADD expr
 23       | expr SUB expr
 24       | expr MUL expr
 25       | expr DIV expr
 26       | SUB expr %prec UMINUS
 27       | '(' expr ')'
 28       | ID
 29       | NUMBER
 30       ;
 31   %%
 32
 33   void yyerror(const char *s) {
 34       has_error = 1;
 35       printf("Error: %s\n", s);
 36   }
 37
 38   int main() {
 39       yyparse();
 40       return 0;
 41   }
```

Output:



## For full code , refer GitHub

https://github.com/arnavjain2710/Compiller-Techniques/tree/main/LAB%206