Practical – 6

Output:

```
PS E:\dsa> cd 'e:\dsa\output'
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: (a)
Valid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: a
Valid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: (a,a)
Valid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: (a,(a,a),a)
Valid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: (a,a),(a,a)
Invalid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: a)
Invalid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: (a
Invalid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: a,a
Invalid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: (a,a),a
Invalid string
PS E:\dsa\output> & .\'practical6.exe'
Enter input string: a,
Invalid string
```

Practical – 7

Output:

```
Enter Non terminal for LEFT hand side: 5
Enter Non terminal or terminal for right hand side: A
B
C
Enter Non terminal or terminal for right hand side2: D
Enter Non terminal for LEFT hand side: A
Enter Non terminal or terminal for right hand side: p
Enter Non terminal or terminal for right hand side2: ^
Enter Non terminal for LEFT hand side: B
Enter Non terminal or terminal for right hand side: q
Enter Non terminal or terminal for right hand side2: ^
Enter Non terminal for LEFT hand side: C
Enter Non terminal or terminal for right hand side: (
S
)
Enter Non terminal or terminal for right hand side2: c
Enter Non terminal for LEFT hand side: D
Enter Non terminal or terminal for right hand side: A
C
FIRST(5) = { p , q , ( , c }
FIRST(A) = { p , ^ }
FIRST(B) = { q , ^ }
FIRST(D) = { ( , c }
FIRST() = { p , ( }
FOLLOW(5) = { ) , $ }
FOLLOW(A) = { q , ( , ) , $ }
FOLLOW(B) = { c , ) , $ }
FOLLOW(D) = ) , $ }
FOLLOW() = ) , $ }
PS E:\dsa\output>
```

Practical – 8

Output:

```
PS E:\dsa> cd 'e:\dsa\output'
PS E:\dsa\output> & .\'practical8.exe'
First Sets:
First(Z) = { r ( }
First(Y) = { ϵ q }
First(X) = { ϵ p }
First(S) = { ( r q ϵ p }

Follow Sets:
Follow(Y) = { $ ) }
Follow(Z) = { $ ) }
Follow(X) = { ) ϵ $ q }
Follow(S) = { ) $ }

Parsing Table:
M[Z, r] = r
M[Z, (] = (S)
M[Y, q] = q
M[X, p] = p
M[S, Z] = Z
M[S, X] = XY

Grammar is LL(1).

Testing: pq -> Invalid

Testing: p -> Invalid

Testing: (pq) -> Invalid

Testing: r -> Invalid

Testing: (p) -> Invalid

Testing: qq -> Invalid

Testing: () -> Invalid

Testing: (qr) -> Invalid

Testing: pqrr -> Invalid

Testing: q -> Invalid
```

```
Testing: (pq) -> Invalid

Testing: (pq) -> Invalid
Testing: (pq) -> Invalid


Testing: r -> Invalid

Testing: (p) -> Invalid

Testing: qq -> Invalid

Testing: () -> Invalid

Testing: (qr) -> Invalid

Testing: pqrr -> Invalid

Testing: q -> Invalid
```

Practical – 9

Output:

```
PS E:\dsa> yacc -d 22ce111_firstyacc.y
22ce111_firstyacc.y: conflicts: 1 shift/reduce
PS E:\dsa> flex 22ce111_firstyacc.l
PS E:\dsa> gcc lex.yy.c 22ce111_firstyacc.tab.c -o 22ce111_firstyacc
22ce111_firstyacc.tab.c: In function 'yyparse':
22ce111_firstyacc.tab.c:586:16: warning: implicit declaration of function 'yylex' [-Wimplicit-function-declaration]
 # define YYLEX yylex ()
                ^
22ce111_firstyacc.tab.c:1246:16: note: in expansion of macro 'YYLEX'
       yychar = YYLEX;
                ^~~~~
22ce111_firstyacc.tab.c:1382:7: warning: implicit declaration of function 'yyerror' [-Wimplicit-function-declaration]
       yyerror (YY_("syntax error"));
       ^~~~~~~
```

```
PS E:\dsa> ./22ce111_firstyacc.exe
Enter the string -> ibt

-> Invalid string!!
PS E:\dsa> ./22ce111_firstyacc.exe
Enter the string -> ibtai

-> Invalid string!!
PS E:\dsa> ibtaea
```

```
PS E:\dsa> ./22ce111_firstyacc.exe
Enter the string -> ibtaea

-> Valid string!
PS E:\dsa> ./22ce111_firstyacc.exe
Enter the string -> a

-> Valid string!
PS E:\dsa> ibtibta
```

```
PS E:\dsa> ./22ce111_firstyacc.exe
Enter the string -> ibtibta

-> Valid string!
```

Practical – 10

Output:

```
PS E:\dsa> yacc -d 10_22CE111.y
PS E:\dsa> flex 10_22CE111.l
PS E:\dsa> gcc lex.yy.c 10_22CE111.tab.c -o 10_22CE111
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: (3 + 5) * 2
=16
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 + 5 * 2
=13
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 + 5 * 2 ^ 2
=23
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 + (5 * 2)
=13
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 + 5 ^ 2 * 2
=51
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 * ( 5 + 2)
=21
```

```
Enter an arithmetic expression: 3 + 5 ^ 2 * 2
=51
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 * ( 5 + 2)
=21
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: (3 + 5) ^ 2
=64
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 ^ 2 ^ 3
=6561
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: 3 ^ 2 + 5 * 2
=19
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: (3 + 5 * 2
Error: syntax error
PS E:\dsa> ./10_22CE111
Enter an arithmetic expression: (3 + 5 * 2 ^ 2 - 8) / 4 ^ 2 + 6
=6
```

Practical – 11

Output:

```
PS E:\dsa\output> & .\'practical11.exe'
Enter an arithmetic expression: 9+42*8

Postfix Form: 9 42 8 * +

Quadruple Representation:
Operation   Operand1    Operand2    Result
-------------------------------------------------
*           42          8           T1
+           9           T1          T2
PS E:\dsa\output> & .\'practical11.exe'
Enter an arithmetic expression: 5+6-3

Postfix Form: 5 6 + 3 -

Quadruple Representation:
Operation   Operand1    Operand2    Result
-------------------------------------------------
+           5           6           T1
-           T1          3           T2
```

```
PS E:\dsa\output> & .\'practical11.exe'
Enter an arithmetic expression: 7-(8*2)

Postfix Form: 7 8 2 * -

Quadruple Representation:
Operation   Operand1    Operand2    Result
-------------------------------------------------
*           8           2           T1
-           7           T1          T2
PS E:\dsa\output> & .\'practical11.exe'
Enter an arithmetic expression: (9-3)+(5*4/2)

Postfix Form: 9 3 - 5 4 * 2 / +

Quadruple Representation:
Operation   Operand1    Operand2    Result
-------------------------------------------------
-           9           3           T1
*           5           4           T2
/           T2          2           T3
+           T1          T3          T4
```

```
PS E:\dsa\output> & .\'practical11.exe'
Enter an arithmetic expression: (3+5*2-8)/4-2+6

Postfix Form: 3 5 2 * + 8 - 4 / 2 - 6 +

Quadruple Representation:
Operation   Operand1    Operand2    Result
-------------------------------------------------
*           5           2           T1
+           3           T1          T2
-           T2          8           T3
/           T3          4           T4
-           T4          2           T5
+           T5          6           T6
```

Practical – 12

Output:

```
PS E:\dsa> cd 'e:\dsa\output'
PS E:\dsa\output> & .\'practical12.exe'
Enter an arithmetic expression: 5+x-3*2
Original: 5+x-3*2
Optimized: ((5 + x) - 6.000000)
PS E:\dsa\output> & .\'practical12.exe'
Enter an arithmetic expression: 2+3*4-1
Original: 2+3*4-1
Optimized: 13.000000
PS E:\dsa\output> & .\'practical12.exe'
Enter an arithmetic expression: x+(3*5)-2
Original: x+(3*5)-2
Optimized: ((x + 15.000000) - 2)
PS E:\dsa\output> & .\'practical12.exe'
Enter an arithmetic expression: (22/7)*r*r
Original: (22/7)*r*r
Optimized: ((3.142857 * r) * r)
```