A. Designing a topdown parser: Consider the following grammar.

```
prog      : stmt-list
stmt-list : stmt-list stmt | stmt
stmt      : PRINT expr | PRINT string
string    : BEGINQUOTE charlist ENDQUOTE
charlist  : charlist LETTER | ε
expr      : expr + term | expr - term | term
term      : term * factor | term / factor | factor
factor    : ( expr ) | NUM
```

The nonterminal symbols are:

prog, stmt-list, stmt, string, charlist, expr, term, factor

The terminal symbols are:

PRINT, BEGINQUOTE, ENDQUOTE, LETTER, +, -, *, /, (,), NUM

(A1) The grammar is left-recursive. Transform it into a non-left recursive grammar using the following ranking of the nonterminals.

prog, stmt-list, stmt, string, charlist, expr, term, factor

(A2) Further modify the grammar so it is left factored.

(A3) Compute the FIRST sets of each nonterminal symbols.

(A4) Compute the FOLLOW sets of each nonterminal symbols

(A5) Compute the predictive parsing table of the grammar.

(A6) Consider the following program.

```
PRINT ( NUM + NUM ) * NUM
```

Use the predictive parsing table to construct the parse tree. At each step, show the production used to expand the nodes.

B. Consider the following source code for a general purpose programming language.

```
declare x : Integer
declare y : Integer
x := 0
y := 10
while( x < 10 and y > 0 )
do
      x := x + 1
      y := ((y - 1) * (2 / x))
end while
```

B.1 Design a grammar for such programs. You do not need to support any programming constructs beyond the ones used by the program. Express it as a ANTLR grammar, named A2.g4

B2. Provide an accompanying program to parse user specified filename. The application should be invoked as follows:

```
java ParseFile <filename>
```

Submission:

1. report.pdf: The written report should clearly answer questions A1 - A6.
2. B.zip: contains at least the following source files:
   a. A2.g4
      The grammar in Antlr v4 format.
   b. ParseFile.java
      The application that invokes the parser to parse the source code of the user specified file.

Marking scheme:

A1	10
A2	5
A3	15
A4	15
A5	10
A6	5
B1	20
B2	20
Total	100