1. Transform the FORTRESS grammar (see Figure 1 at the end of document) in order to: (a) Remove unproductive and/or unreachable variables, if any; (b) Make the grammar non-ambiguous by taking into account the priority and the associativity of the operators. Table 1 shows these priorities and associativities: operators are sorted by decreasing order of priority (with two operators in the same row having the same priority). Please note that you do not have to handle priority if there is no ambiguity. (c) Remove left-recursion and apply factorisation where needed;

```
[1]  <Program>     → BEGIN [ProgName] <Code> END
[2]  <Code>        → <Instruction> , <Code>
[3]                → ε
[4]  <Instruction> → <Assign>
[5]                → <If>
[6]                → <While>
[7]                → <Print>
[8]                → <Read>
[9]  <Assign>      → [VarName] := <ExprArith>
[10] <ExprArith>   → [VarName]
[11]               → [Number]
[12]               → ( <ExprArith> )
[13]               → - <ExprArith>
[14]               → <ExprArith> <Op> <ExprArith>
[15] <Op>          → +
[16]               → -
[17]               → *
[18]               → /
[19] <If>          → IF (<Cond>) THEN <Code> END
[20]               → IF (<Cond>) THEN <Code> ELSE <Code> END
[21] <Cond>        → <ExprArith> <Comp> <ExprArith>
[22] <Comp>        → =
[23]               → >
[24]               → <
[25] <While>       → WHILE (<Cond>) DO <Code> END
[26] <Print>       → PRINT ([VarName])
[27] <Read>        → READ ([VarName])
```

Figure 1: The FORTRESS grammar.

a) removing unproductive symbols

i	V_i
0	$\emptyset$
1	{Code, ExprArithm, op, Comp, Print, Read}
2	{Code, ExprArithm, op, Comp, Print, Read, Program, Instruction, Assign, Cond}
3	{Code, ExprArithm, op, Comp, Print, Read, Program, Instruction, Assign, Cond, If, While}

⇒ all variables $\in V_3$ ⇝ no possible simplification

b) removing unreachable variables.

i	V_i
0	{Program}
1	{Program, Code}
2	{Program, Code, Instruction}
3	{Program, Code, Instruction, Assign, If, While, Print, Read}
4	{Program, Code, Instruction, Assign, If, While, Print, Read, ExprArith, Cond}
5	{Program, Code, Instruction, Assign, If, While, Print, Read, ExprArith, Cond, op, comp}

all variables $\in V_5$ ⇝ no unreachable variables ⇝ no possible simplification

First and Follow