1. The course DFA on input aba ends in state –

   ☐ q0

   ☐ q2

   ☐ q1

   ☐ no state

2. In the layered grammar, id + id * id has – parse tree(s)

   ☐ none

   ☐ two

   ☐ one

   ☐ three

3. The halting problem proof builds D(M) that –

   ☐ prints banana

   ☐ rejects everything

   ☐ loops exactly when H says M halts on M

   ☐ always halts

4. The lexer emits –

   ☐ (type, lexeme, position) triples

   ☐ assembly

   ☐ raw characters

   ☐ parse trees

5. The LL(1) table is filled from –

   ☐ FIRST and FOLLOW sets

   ☐ parse trees

   ☐ random choice

   ☐ the lexer

6. The symbol table binds –

   ☐ nothing

   ☐ tokens to trees

   ☐ identifiers to type and scope

7. The pumping lemma shows a^n b^n is not regular because –

   ☐ DFAs reject all strings

   ☐ pumping y breaks the a-b count

   ☐ it has too many symbols

   ☐ it is infinite

8. The leftmost derivation of id + id * id takes – steps

   ☐ six

   ☐ five

   ☐ seven

   ☐ four

9. Rice's theorem covers –

   ☐ syntax errors

   ☐ parser speed

   ☐ every nontrivial semantic property

   ☐ memory leaks

10. The input ifx lexes as –

    ☐ an error

    ☐ KW(if) then IDENT(x)

    ☐ KW(if) then NUM

    ☐ one IDENT ifx

1. **q1**

   q0-a-q1, q1-b-q2, q2-a-q1: ends q1, rejected (q2 accepts, not q1).

2. **one**

   Layering enforces precedence: x binds under T, deeper than +.

3. **loops exactly when H says M halts on M**

   D flips the answer about itself - running D on D contradicts.

4. **(type, lexeme, position) triples**

   Type for the parser, lexeme for messages, position for errors.

5. **FIRST and FOLLOW sets**

   FIRST chooses expansions; FOLLOW handles the empty production.

6. **identifiers to type and scope**

   Name resolution before type checking: scoping first.

7. **pumping y breaks the a-b count**

   y lies in the a-block; pumping changes the count of a without b.

8. **six**

   E, E+T, T+T, id+T, id+T*F, id+id*F, id+id*id - six.

9. **every nontrivial semantic property**

   Nontrivial properties of WHAT the program computes - undecidable.

10. **one IDENT ifx**

    Greedy longest match: the ident regex permits the longer lexeme.