1. PL Concepts
   1.1. “Goodbye!”, “Goodbye!”
   1.2. It translates a program from one language to another (1/2 point given for answer B)
   1.3. Ruby
   1.4. “It determines reachability incrementally, as the program executes” AND “Memory can be freed immediately once it becomes unreachable”
   1.5. “A Java interface only applies to classes that explicitly declare it, but a Rust trait can be applied to any type”
   1.6. “uniq should return a list of the same length whether given a list or its reverse”, “List of small integers”

2. Ruby
   2.1. #1: “yield grade”
        #2: “0”
   2.2. x["3xx"] = [330, 351]
   2.3. /^a((0-9a-zA-Z)[2])z$/
   2.4. #1: collect OR map
        #2: val/2
   2.5. #1: attr_reader :n, :m
        #2: "#{n} + #{m}"

3. OCaml
   3.1. fun a -> 0
   3.2. fun a b -> match b with [] -> failwith “” | h::_ -> [h a]
   3.3. error, z is used both as int and bool
   3.4. x+1 (many correct answers)
   3.5. [6;0;4]
   3.6. #1: min x lst
       #2: []
       #3: h::t

4. DFA/NFA
   4.1. (ab+)+|(a*b*)
   4.2. ab, aba
   4.3. 2, 3
   4.4. 2, 3, 4
   4.5. 4
   4.6. X, Y, Z
   4.7. The third one
5. **CFG and Parsing**
   5.1. False
   5.2. False
   5.3. \[ S \rightarrow T \rightarrow TT \rightarrow TTT \rightarrow cTT \rightarrow ccT \rightarrow ccc \]
   \[ S \rightarrow SS \rightarrow TS \rightarrow cS \rightarrow cSS \rightarrow cTS \rightarrow cTT \rightarrow ccT \rightarrow ccc \]
   5.4. \[ \text{Exp} \rightarrow \text{Add} \rightarrow \text{Mul} \rightarrow \text{Primary} \ast \text{Mul} \rightarrow \text{ID} \ast \text{Mul} \rightarrow x \ast \text{Mul} \rightarrow x \ast \text{Primary} \]
   \[ \rightarrow x \ast \text{Int} \rightarrow x \ast 2 \]

6. **Security**
   6.1. Buffer overflow
   any number >49
   add “if (n>49){n=49}” before line 5
   6.2. The size of the buffer is part of the type, and the compiler adds bounds checks before memory is accessed

7. **Rust**
   7.1. #1: any int type (i32, u32, etc)
   #2: mut
   #3: i -= 1
   7.2. if let Flavor::Savory = self.flavor { “savory” } else { “sweet” }
   7.3. a
   7.4. the value is dropped (this question is not counted due to the mistake)
   7.5. make my_string mutable by adding “mut” keyword, pass “&mut my_string”
   to do_stuff, and change arg type of do_stuff to be “&mut String”
   7.6. 4, 7
   7.7. 9, 10

8. **Lambda Calculus**
   8.1. A, D
   8.2. A
   8.3. B
   8.4. It never reaches a normal form
   8.5. To implement recursion