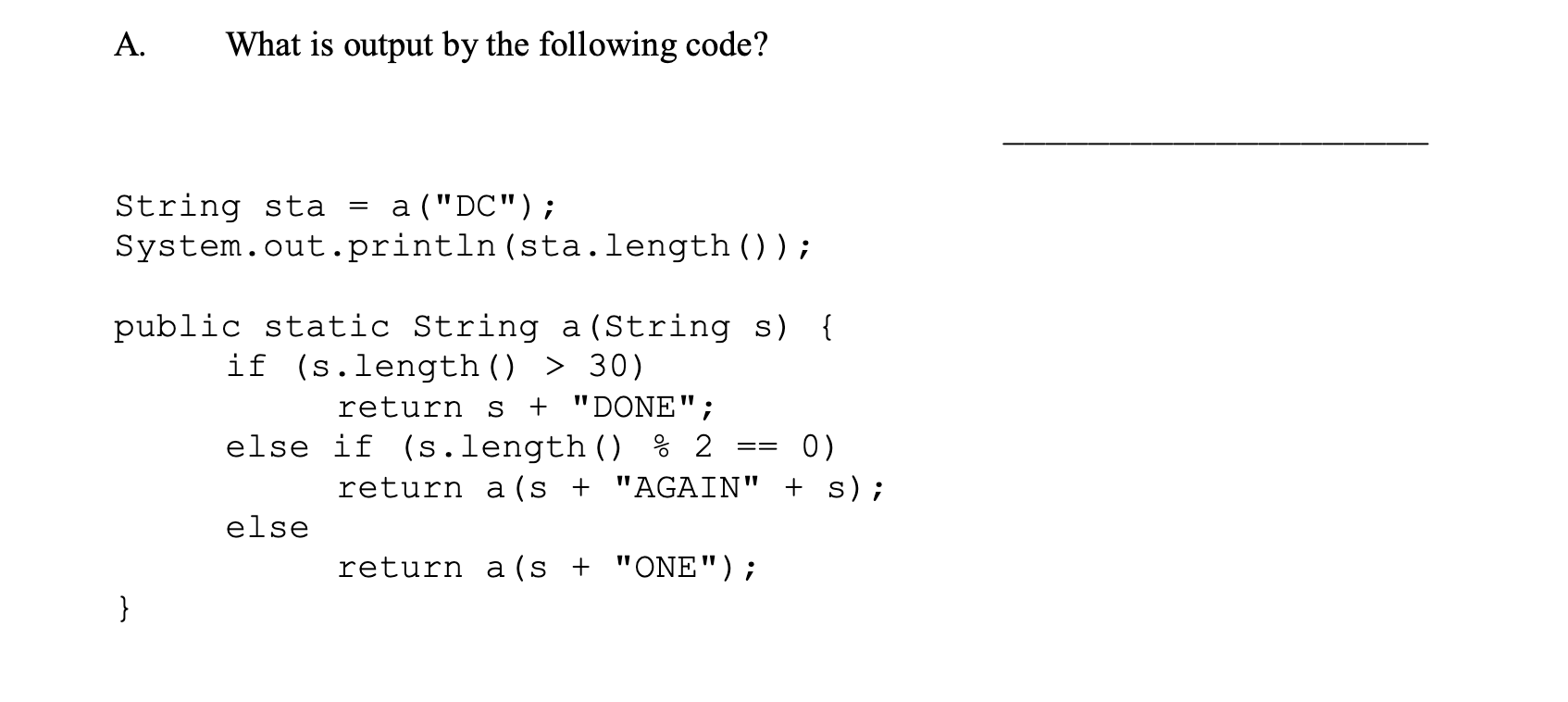
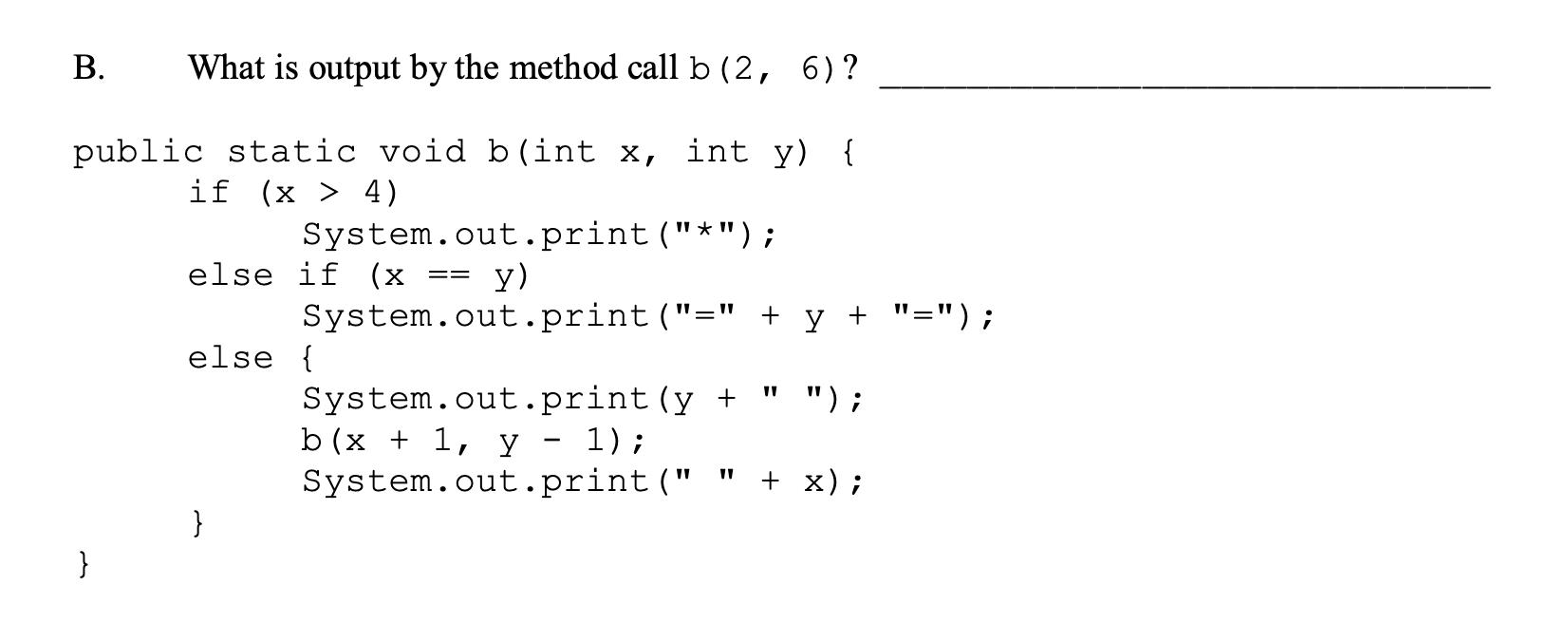
SI: Recursive Backtracking





5-year-old Shinchan had just started learning mathematics. Meanwhile, one of his studious classmates, Kazama, had already written a basic calculator which supports only three operations on integers: *multiplication, addition, and subtraction*. Since he had just learned about these operations, he didn't know about operator precedence, and so, in his calculator, all operators had the same precedence and were left-associative (explained below).

As always, Shinchan started to irritate him with his silly questions. He gave Kazama a list of integers and asked him to insert one of the above operators between each pair of consecutive integers such that the result obtained after feeding the resulting expression in Kazama's calculator is divisible by **101**. At his core, Shinchan is actually a good guy, so he only gave lists of integers for which an answer exists. Can you help Kazama create the required expression? If multiple solutions exist, print any one of them.

**Input Format**

A List<Integer> object to represent the list of numbers that needs to be computed.

**Output Format**

Print a single line containing the required expression. You may insert spaces between operators and operands.

**Note**

* You are not allowed to permute the list (change the order of the numbers in the list).
* All operators have the same precedence and are left-associative, e.g. a + b \* c –d \* e, is interpreted as ((((a + b) \* c) – d) \* e)
* Unary plus and minus are not supported, e.g., statements like -a, a \* -b, or –a \* b + c are invalid.

**Sample Input 0**

22 79 21

**Sample Output 0**

22\*79-21

**Explanation 0**

*Solution 1:* 22 \* 79 – 21= 1717, where 1717 / 101 = 17, so it is perfectly divisible by 101.   
*Solution 2:* 22 + 79 \* 21 = (22 + 79) \* 21 = 2121, which is also divisible by 101.

**Sample Input 1**

55 3 45 33 25

**Sample Output 1**

55+3-45\*33-25

**Explanation 1**

55 + 3 – 45 \* 33 – 25 = ((((55 + 3) – 45) \* 33) - 25) = 404 which is divisible by 101.

// No need to check for preconditions.

// Method is void; you are supposed to print your answer.

// Don’t forget you can use helper methods if you don’t want to // have a recursive function with a list as the argument.

public void calculate(List<Integer> nums) {