This is an example of the annotation interaction. All human annotator inputs are highlighted in blue.

Problem set up. First off, the evaluation is set up by displaying the problem name, the problem statement, the buggy code, the bug descriptions, the ground truth bug fixes, and the suggested bug fixes:

```
Problem:  073-set-matrix-zeroes.py
Here is the problem statement:


---
problem:


Given a m x n matrix, if an element is 0, set its entire row and column to 0. Do it in-place.


Example 1:
Input:
[
  [1,1,1],
  [1,0,1],
  [1,1,1]
]
Output:
[
  [1,0,1],
  [0,0,0],
  [1,0,1]
]


Example 2:
Input:
[
  [0,1,2,0],
  [3,4,5,2],
  [1,3,1,5]
]
Output:
[
  [0,0,0,0],
  [0,4,5,0],
  [0,3,1,0]
]
---
```

Here is the buggy code:

```
---
buggy_code:
1. class Solution(object):
2.     def setZeroes(self, matrix):
3.         row = len(matrix)
4.         col = len(matrix[0])
5.         is_col= False
6.         for i in range(row):
7.             if matrix[i][0] == 0:
8.                 is_col= True
9.             for j in range(0,col):
10.                 if matrix[i][j] == 0:
11.                     matrix[0][j] = 0
12.                     matrix[i][0] = 0
13.
14.         for i in range(1,row):
15.             for j in range(1,col):
16.                 if not matrix[i][0] or not matrix[0][j]:
17.                     matrix[i][j] = 0
18.         if matrix[0][0] == 0:
19.             for j in range(col):
20.                 matrix[0][j] = 0
21.         if is_col:
22.             for i in range(row):
23.                 matrix[i][0] = 0
```

Here is the bug description:

---

bug_desc:

On line 9, using range(0,col) introduces a bug where the code unnecessarily modifies the first column of the matrix. To avoid this mistake, use range(1,col) in the for-loop.

Here are the bug fixes:

---

bug_fixes:

Replace `range(0,col)` with `range(1,col)` on line 9.

Here are the PREDICTED bug fixes:

["['Instead of setting only the first elements of the row and column to zero, we need to set all elements in the same row and same column to zero. This can be done by iterating through the entire row and column whenever a zero element is found.']", "['In lines 11-12, add the following lines after the inner loop to set the first element in the first row and first column to zero when a zero is found in the first column: `matrix[0][0] = 0` and `if i == 0: matrix[i][0] = 0`.', 'Change the starting index of the inner loop from 0 to 1 in lines 14 and 15']"]

Calculating success rate. The evaluator is asked for the success rates of the conceptual bugs and syntactical bugs. In single-bug and one-bug settings, the evaluator has been tasked to not provide any success rate for a non-existing bug type. For example, there is no syntactical bug in this question, and therefore there is no success rate provided:

CONCEPTUAL success rate: **1**

SYNTACTICAL success rate:

Turn level evaluation. For each turn in the conversation, the evaluator is presented with the Instructor question and the Student response, and they have to answer four questions regarding the type of bug the question is targeted towards, relevance, indirectness, and logical flow:

TURN 1/3:

Instructor: Can you explain the purpose of the nested loop starting at line 9 and what it's doing to the matrix?

Student: The nested loop starting at line 9 sets all elements in the same row and same column of a zero element to zero. It does this by changing the value of the first element in the row and the first element in the column to zero. This allows the following loops to identify these rows and columns as having at least one zero element, and set all other elements in those rows and columns to zero.

The question asked by the instructor was related to a syntactical bug (0) or conceptual bug (1): **1**

The question asked by the instructor was relevant to the bugs in the Student code (tag: "bug_fixes"). (0: No; 1: Yes): **1**

The question asked by the instructor does NOT explicitly state any of the bug fixes (tag: "bug_fixes"). (0: No; 1: Yes): **0**

The question asked by the instructor made the conversation flow in a logical manner and guided the user to solve their problem. (0: No; 1: Yes): **1**

TURN 2/3:

Instructor: Can you explain the role of the first column in your current approach and what happens when you iterate over it in the initial loop?

Student: The first column in the current approach plays a special role in identifying the need to set entire rows and columns to zero. When iterating over the first loop, if an element in the first column is found to be zero (line 7), the variable 'is_col' is set to True. If later during the second loop (lines 14-17), we encounter a zero element in the same column (or same row), we set that element to zero and also set the entire row and column to zero. Therefore, the first column serves as a marker to identify zero elements in the entire row or column. However, the current implementation has a bug where setting the first element in the first row and first column to zero is missed (lines 11-12). Therefore, an additional check and set to zero is required for the first element in the first row and first column.

The question asked by the instructor was related to a syntactical bug (0) or conceptual bug (1): **1**

The question asked by the instructor was relevant to the bugs in the Student code (tag: "bug_fixes"). (0: No; 1: Yes): **1**

The question asked by the instructor does NOT explicitly state any of the bug fixes (tag: "bug_fixes"). (0: No; 1: Yes): **1**

The question asked by the instructor made the conversation flow in a logical manner and guided the user to solve their problem. (0: No; 1: Yes): **1**

TURN 3/3:

Instructor: That's not entirely correct, let me rephrase the question. Can you describe the impact of starting the inner loop from the first column instead of the second column in the initial iteration? How might this affect the rest of the matrix?

Student: Starting the inner loop from the first column instead of the second column in the initial iteration means that the first row will be processed twice. In the first iteration of the outer loop, if the first element of the first row is zero, then the entire first row and first column will be set to zero. However, in the subsequent iteration of the inner loop for the first column, the same element will be encountered again, and it will set the first column to zero once more. This would result in an additional unwanted zero entry in the matrix, affecting the other entries in the first row and column.

The question asked by the instructor was related to a syntactical bug (0) or conceptual bug (1): **1**

The question asked by the instructor was relevant to the bugs in the Student code (tag: "bug_fixes"). (0: No; 1: Yes): **1**

The question asked by the instructor does NOT explicitly state any of the bug fixes (tag: "bug_fixes"). (0: No; 1: Yes): **1**

The question asked by the instructor made the conversation flow in a logical manner and guided the user to solve their problem. (0: No; 1: Yes): **1**

This is repeated for all conversations across all settings and baselines.