**Question 1 - multiple choice, shuffle**

Does fuzz.py identify a crash in wisdom-alt? In how many iterations?

\*A: Identifies a crash, one iteration

Feedback: Radamsa immediately generates a bogus input; indeed, any input that is not a "1" or a "2" is going to cause problems.

B: Does not identify a crash

C: Identifies a crash, 103 iterations

D: Identifies a crash, 44 iterations

**Question 2 - multiple choice, shuffle**

Does fuzz.py identify a crash in wisdom-alt2? In how many iterations?

\*A: Does not identify a crash

Feedback: Now that we've fixed the first problem, it's much harder for radamsa to find an input that overruns the buffer, which is the second bug in the file. It tried many inputs, to no avail.

B: Identifies a crash, 1 iteration

C: Identifies a crash, 800 iterations

D: Identifies a crash, 133 iterations

**Question 3 - text match**

Name one symbolic variable that was set in the path condition identified by KLEE that crashes wisdom-alt2.

\*A: buf

\*B: r

Default Feedback:

**Question 4 - text match**

Name another symbolic variable set in the path condition identified by KLEE that crashes wisdom-alt2.

\*A: buf

\*B: r

Default Feedback:

**Question 5 - multiple choice, shuffle**

What was the data content of the buf object?

\*A: '\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'

Feedback: This is the content that KLEE generates; the contents don't matter so much as its length matters, but in any case we want you to tell us what KLEE selected.

B: '\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xFF\x00\x00\x00\x00\x00\x00\x00\x00\xAA'

C: '\xFF\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'

D: '\x00\x00\x00\xFF\x00\x00\x00\x00\x00\x00\xBB\x00\x00\x00\x00\x00\x00\x00\x00\xEE'

**Question 6 - text match**

After executing the symbolic maze, what was the data value of the 'program' object? (Hint: it will be a string of the lowercase letters **s**, **d**, **w**, and **a**.)

\*A: ssssddddwwaawwddddssssddwwww

Feedback: The program object contains the solution to the maze, modulo the bug. So any solution is an acceptable answer. Just depends on the paths that KLEE takes in its search.

\*B: ssssddddwwaawwddddsddw

Feedback: The program object contains the solution to the maze, modulo the bug. So any solution is an acceptable answer. Just depends on the paths that KLEE takes in its search.

\*C: sddwddddssssddwwww

Feedback: The program object contains the solution to the maze, modulo the bug. So any solution is an acceptable answer. Just depends on the paths that KLEE takes in its search.

\*D: sddwddddsddw

Feedback: The program object contains the solution to the maze, modulo the bug. So any solution is an acceptable answer. Just depends on the paths that KLEE takes in its search.

Default Feedback: The program object contains the solution to the maze, modulo the bug. So any solution is an acceptable answer. Just depends on the paths that KLEE takes in its search.

**Question 7 - numeric**

If you run the symbolic maze program so that it finds all solutions, not just one, how many are there?

\*A: 4.0

Default Feedback:

**Question 8 - numeric**

There was a bug in the maze program that allows the player to walk through walls. What line in $$\color{red}{\verb|maze-sym.c|}$$ is the bug on? (If there are multiple lines, pick one of them.)

\*A: 113.0

Feedback: The extra condition in the conditional whose guard covers lines 111-113 is incorrect (but the part of the conditional on line 111 is not wrong). Lines 112-113 should just be commented out and the conditional guard closed off, to make the problem go away.

\*B: 112.0

Feedback: The extra condition in the conditional whose guard covers lines 111-113 is incorrect (but the part of the conditional on line 111 is not wrong). Lines 112-113 should just be commented out and the conditional guard closed off, to make the problem go away.

Default Feedback: The extra condition in the conditional whose guard covers lines 111-113 is incorrect (but the part of the conditional on line 111 is not wrong). Lines 112-113 should just be commented out and the conditional guard closed off, to make the problem go away.