Dynamic Symbolic Execution

Question 1. Random Testing (Fuzzing) vs. DSE

a. State one advantage of random testing over DSE.

Answer: Random testing is black-box: it does not require the source/binary code of the program under test. DSE is white-box: it requires the code.

b. State one advantage of DSE over random testing.

Answer: DSE may be more effective on certain programs: it may find a crashing input faster than random testing, it may give higher coverage than random testing in the same time budget, etc.

Question 2. Consider the following program:

```
int main(char[] input) {
  if (input[0] == 'C')
    if (input[1] == 'S')
      if (input[2] == '6')
        if (input[3] == '3')
          if (input[4] == '4')
            if (input[5] == '0')
                return 1 / 0;
  return 0;
}
```

Which of the following two approaches will be more effective at discovering the divide by 0 error present in the `main` function? Explain your answer.

a. Fuzzing the main program with randomly generated strings comprised of exactly six alphanumeric characters.
b. Executing DSE on the `main` function with an arbitrary starting string input. Assume that the solver is capable of solving character equivalence constraints.

Answer: DSE will be more effective. Because the error is triggered when the input string's first six characters are 'CS6340', the fuzzer will take a very long time to randomly generate that exact six character string and trigger the bug. DSE will require one iteration per branch to generate a concrete input satisfying the constraints necessary to trigger the bug, which will execute more quickly than the fuzzer would.

Question 3. Consider running dynamic symbolic execution on each of the following programs. Suppose the starting inputs are `x = 3`, `y = 9`, and suppose the symbolic solver DSE has access to can only solve linear equations and inequalities (in particular, it cannot solve equations and inequalities where variables are squared). Assume integer overflows do not occur.

```
int foo(int x, int y) {
  if (x*x != y*y) {
    return x * y;
  }
  else {
    if (x < y) {
      return x / y;
    }
    else {
      return x + y;
    }
  }
}
```

```
int bar(int x, int y) {
  if (x == y*y) {
    return x / y;
  }
  else {
    if (x < y) {
      return x;
    }
    else {
      return y;
    }
  }
}
```

```
int qux(int x, int y) {
  if (x == y) {
    return x / (x - y);
  }
  else {
    if (x > y) {
      return x / y;
    }
    else {
      return y / x;
    }
  }
}
```

For which of these programs will DSE correctly return that the program has a division-by-zero error? (Select all that apply.)

a. foo
b. bar
c. qux
d. None of the above

Answer: c

For which of these programs will DSE correctly return that the program has <u>NO</u> division-by-zero error? (Select all that apply.)

a. foo
b. bar
c. qux
d. None of the above

Answer: a

For which of these programs will DSE <u>wrongly</u> return that the program has a division-by-zero error? (Select all that apply.)

a. foo
b. bar
c. qux
d. None of the above

Answer: d

For which of these programs will DSE wrongly return that the program has NO division-by-zero error? (Select all that apply.)

a. foo
b. bar
c. qux
d. None of the above

Answer: b