

Quiz: Property Based Testing

Find and fix the bug in the second property test below.

```
1 class BrokenSpec extends AnyFreeSpec with PropertyChecks with Matchers {
2
3   // Our usual sequence Option-List. No surprises
4   def sequence[A] (aos: List[Option[A]]): Option[List[A]] =
5     aos.foldRight[Option[List[A]]] (Some (Nil)) {
6       (oa, z) => z flatMap (l => oa map (_::l)) }
7
8   "Returns Some if the list has no failures" in {
9     implicit def arbList[A] (implicit arb: Arbitrary[List[A]]) =
10       Arbitrary[List[Option[A]]] (arb.arbitrary map { _ map (Some (_)) })
11
12     forAll { (l: List[Option[Int]]) =>
13       sequence (l).isDefined shouldBe true}
14   }
15
16   "Returns None if the list has one failure" in {
17     forAll { (l: List[Option[Int]]) => sequence (l).isEmpty shouldBe true}
18   }
19 }
```

Quiz: Property Based Testing

Find and fix the bug in the second property test below.

```
1  implicit def arbList[A] (implicit arb: Arbitrary[List[A]]) =
2    Arbitrary[List[Option[A]]] (arb.arbitrary map { _ map (Some (_)) })
3
4  forAll { (l :List[Option[Int]]) =>
5    sequence (l) shouldNot be (None) }
6  }
7
8  "Returns None if the list has one failure" in {
9    implicit def arbFailingList[A] (implicit arb: Arbitrary[List[Option[A]]]) =
10      Arbitrary (arb.arbitrary filter { _ exists (_.isEmpty) })
11
12    forAll { (l: List[Option[Int]]) => sequence (l) shouldBe None }
13  }
14
15 }
```

Have seen the problem:1pt + Have fixed the problem:1pt = Max total:2pt