

# Quiz: State and Rand

no stress, no embarrassment, no consequences

Write a function `random_list` that given a value of type `Rand[A]` (a generator of random values of type `A`) produces a generator of lists of random values of type `A`.

Each list is of a given size `n`:

```
def random_list[A](n: Int)(ra: Rand[A]): Rand[List[A]]
```

**Hints:** You may want to use the `tabulate` or `fill` function on lists:

```
def List[A].tabulate(n: Int)(e: Int => A): List[A].
```

```
def List[A].fill(n: Int)(e: => A): List[A].
```

Also recall the `sequence` function on random number generators (or more generally on states):

```
def sequence[A](fs: List[Rand[A]]): Rand[List[A]]
```

# An example solution

```
1 def random_list[A](n Int)(ra: Rand[A]): Rand[List[A]] =  
2   val rand_list = List.tabulate(n) { _ => ra }  
3   // val rand_list = List.fill(n)(ra)  
4   sequence(rand_list)
```

- 2 points for a correct solution,
- 1 point for a solution that creates a list of generators instead of a generator of a list + other smaller flaws
- No points otherwise (ask if in doubt).