Quiz: State and Rand

10 minutes, no stress, no embarrassment, no consequences, but **alone and quietly**

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 a 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].
```

*google for "List scaladoc, if you need API description"

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

```
def random_list[A] (length: Int) (ra: Rand[A]): Rand[List[A]] =  {
  val rand_list = List.tabulate (length) ( _ => ra)
  // val rand_list = List.fill (length) (ra)
  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).