Quiz: 050 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]]
```

You may want to use the `tabulate` function on lists:

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

Also recall the `sequence` methods 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] (length :Int) (ra :Rand[A]) :Rand[List[A]] =  {
2    val rand_list = List.tabulate (length) ( _ => ra)
3    sequence (rand_list)
4  }
```

2 points for a correct solution, 1 point for a solution that creates a list of integers instead of a generator of such, or other serious flaws. Zero points otherwise (ask if in doubt).