Challenge project: Heat flow in a wire

We model one-dimensional heat movement in a wire by describing "temperatures" at a large number of locations along the wire. Think of them as closely-spaced points $x_0, x_1, \ldots, x_n$ along a wire. We store the temperatures at these locations in an array `temps[]` of type `double`. The value in `temps[k]` represents the temperature at location x_k. For the purposes of this assignment, we keep the number of points small; set $n = 10$.

We create a primitive way of visualizing the temperature distribution. Do this by taking the value of `temps[k]`, rounding it down to the nearest integer, and then printing out that many `X`'s. For example, if the temperature is 2.5 everywhere, we print out

```
XX
XX
XX
...
XX
```

Write a program that does just this. There should be one loop that fills the array with all values 2.5 and another array that does the primitive printing.

Now that we have a way to visualize heat at any given time, we want to make the heat move about. We move the heat about in small steps. In each step:

- 10% of the heat at each location moves to the left
- 10% of the heat at each location moves to the right
- the remaining heat stays where it is.

This happens everywhere except at the endpoints, where we need to make a choice about what happens to the heat. Assume that heat can leave the wire, but not enter.

Add a loop that adjusts the values of the temperature distribution according to these rules. I found it helpful to create a second "temporary" array.

Finally, have your program take in an integer n and iterate the heat-moving process that many times. What happens if the heat in initially a spike (all in one place)? Run with

```
./a.out <<< "5"
```

and replace "5" by various values. Or print out various time steps all at once.