Assignment 8

1. Grammar expansion

Task 1.1

We simply look for the rule corresponding to the symbol in the map. If we find a rule we return its value otherwise we return the symbol unchanged.

Task 1.2

The function `expandOnce` is really straightforward. We iterate on the given string and call `expandSymbol` on the current character to create the new string and return it.

Task 1.3

We iteratively call `exapndOnce` `num_iters` times and return the final string.

2. Drawing

Task 2.1

Here we use a `switch` statement on each symbol of the given string to Perform the drawing. We are using two stacks. One for the pen's current position and another one for the current angle. If the symbol isn't one of `+`, `-`, `[` or `]` we assume it is the drawing symbol.

3. Understanding the expansion rules

Task 3.1

`left_plant`

We simply looked at iteration 0 and iteration 1 and noticed how one line changed and it luckily was only that.

`crossout`

Same as `left_plant`

`hexerode`

Same as `left_plant`

`flower_plant`

This one was trickier because the evolution isn't as trivial so we tried using 2 symbols instead of only one and after some thinking and some trial and error we found a result that looked correct.

4. Stochastic systems

Task 4.1

The main problem here was to respect the distribution of the probabilities (assuming they all sum up to 1).
We roll a die, which gives us a random number between 0 and 1. Then our algorithm works with ranges. Each rule has a probability and thus has a proportionate part of that [0,1] range assigned to it, then we see in which rule's range the die is rolled in and this gives us the rule to apply.

Example using 3 rules:

- Rule 1 has probability 0.5 and its range is $[0, 0.5]$
- Rule 2 has probability 0.3 and its range is $]0.5, 0.8]$
- Rule 3 has probability 0.2 and its range is $]0.8, 1]$

The rules' probabilities must sum to 1 so the total range will always be $[0, 1]$

We roll the die we get $x = 0.9324234$
Thus the rule we chose is number 3.

But instead of computing these ranges we proceed iteratively be summing the precedent rules' probabilities.

Workload

- Lucien Michaël Iseli, 274999: **33.33%**
- Lucas Strauss, 272432: **33.33%**
- Joachim Dunant, 262314: **33.33%**