Mathematica Workshop - Problem Set

The problem set for the workshop “Quick intro to Mathematica, CloudLab’s symbolic language.”
(By Alfredo Gonzalez-Espinoza and Huajin Wang @ Carnegie Mellon University. September 2024.)

The Context

We are designing an experiment that requires four reagents. Each reagent has a name and a corresponding volume as follows:

template: 3 uL
mastermix: 2.5 uL
primer1: 5 uL
primer2: 5 uL

In the following problem set, we want to explore how to store the information of these reagents in variables and extract information as needed, using the programming language Mathematica.

1. List Creation and Manipulations.

1.1. Create two Lists to store the reagent names and their corresponding volumes. Let’s name the two Lists `reagentList` and `volumeList`, respectively.

```
reagentList =
volumeList =
```

1.2. List splicing. From the Lists created above, let’s take a look at the third ingredient and its volume using List splicing.

```

```

1.3. Organize the output from 1.2 in a list so that it’s easier to manage.

```

```

1.4. Nested List. Building on the result from 1.3, we'd like to organize all the ingredients and their their volumes in a Nested List, so that we can conveniently look up each ingredient and their corresponding volume. Let’s call this Nested List `reagentVolList`.

```
reagentVolList = {


    }
```

1.5. Now, extract the information about the third ingredient and its volume.

```

```

2. Association

Now, we want to further improve the way we organize all the ingredients and their their volumes so we'd be able to look up the volume used for each ingredient by their names. Let's use Association as the data structure to accomplish this.

2.1. Create an Association named `reagentVolAssociation`, that uses reagent names as keys and volumes as values. Hint: use the Lists we created in the beginning of the exercise to create keys and values. Remember, Association works like "dictionaries": key (word) -> value (meaning).

```
reagentVolAssociation =


```

2.2. Now, let's use the Association you created to look up what volume to use for "primer1"

```

```

3. Automation With Loops [Bonus]

In the exercises above (1.4 and 2.1), we repeated the same action four times to create the Nested List or the Association, with the only difference being the indexes. This will become tedious if we are dealing with many more ingredients. Fortunately, we can "automate" this process with a Loop.

3.1. Table. Complete exercise 1.4 with a loop, using the "Table" function. Let's give the resulted Nested List a new name, `newReagentVolList`, so that we don't overwrite the previous result.

```
newReagentVolList = Table[


]
```

3.2. For loop. Complete exercise 2.1 with a "For loop". Let's name the resulted Association `newReagentVolAssociation`. To do this, first, create an empty Association, then, iterate through the indexes in the `reagentList` and `volumeList`, then, add one key /value pair per iteration with `AppendTo[x,elem]` command. Finally, it would be good to print out the value of `newReagentVolAssociation` so you can view the result.

```
(*initializing with an empty dictionary*)
newReagentVolAssociation = Association[]
For[  , , ,
(*AppendTo adds a new entry to an existing Association *)
    AppendTo[  ,  ]
    ]

newReagentVolAssociation
```

The End.