HASH-MAPS & HASH-SETS

POETRY OF PROGRAMMING – CLOJURE ASSIGNMENTS

(1) Write a function `replace-chars` that takes a character-to-character hash-map and a string, and returns a string that has characters replaced according to the supplied lookup table.

```
(replace-chars {\t \p, \p \t} "pot")
"top"
(replace-chars {\e \a, \l \p, \o \y} "hello")
"happy"
```

(2) Write a function that takes two sets and returns the shared elements.

```
(shared #{1 2 3} #{2 3 4 5})
#{2 3}
```

(3) Write a function that takes a collection of sets (at least one) and returns the elements shared by all sets. Hint: `shared` from the previous problem might be a suitable reducing function.

```
(shared-by-all [#{1 2 3} #{2 3 4 5} #{1 3 5}])
#{3}
```

Note: functions for dealing with sets are available in the `clojure.set` namespace. This exercise is a reimplementation of `clojure.set/intersection`.