ECON 381 HOMEWORK-3

1. Which distance metric is usable for distances between keys?

ANSWER:

Manhattan distance is best suited because:

- Measures moves in up/down/left/right directions
- Matches remote control navigation
- Easy to calculate on grid layout
- Accurately represents actual key selection effort

2. Would you need a particular data structure to represent the keyboard layout? Would this structure be needed permanent or once we calculate the distances between keys, could it be replaced by another structure?

```
// Initial layout representation

char[][] keyboard = {

  {'1','2','3','4','5','6','7','8','9','0'},

  {'q','w','e','r','t','y','u','i','o','p'},

  {'a','s','d','f','g','h','j','k','l'},

  {'z','x','c','v','b','n','m'}

};
```

Can be replaced with distance map after initial calculation.

3. Suppose we decided to map each key to a list of valid moves (ie. other keys with 2 to 3 distance). What kind of Java data structure be the best suited for this?

ANSWER:

```
HashMap<Character, List<Character>> validMoves = new HashMap<>();
```

HashMap is ideal because:

- O(1) lookup time
- Easy to store key-to-moves mapping

- Simple to update/access

4. Write pseudocode (or Java code) for creating an 8 character password using the data structure you suggested.

ANSWER:

```
generatePassword(char firstChar):
  password = [firstChar]
  currentChar = firstChar

  while password.length < 8:
    validChars = validMoves.get(currentChar)
    nextChar = getRandomChar(validChars)
    password.add(nextChar)
    currentChar = nextChar

  return password
```

5. Compute the list of valid moves for for the following keys: a, f, h, 8, 0, and p.

ANSWER:

a -> {d, r, 4, w}

f -> {c, t, h, 3}

h -> {e, k, u, 5}

8 -> {5, i, k, 0}

0 -> {7, o, l, p}

p -> {m, i, 7, 0}

These moves are all 2-3 Manhattan distance away from their respective keys.

Elif Işıl Çiçek

23232810007