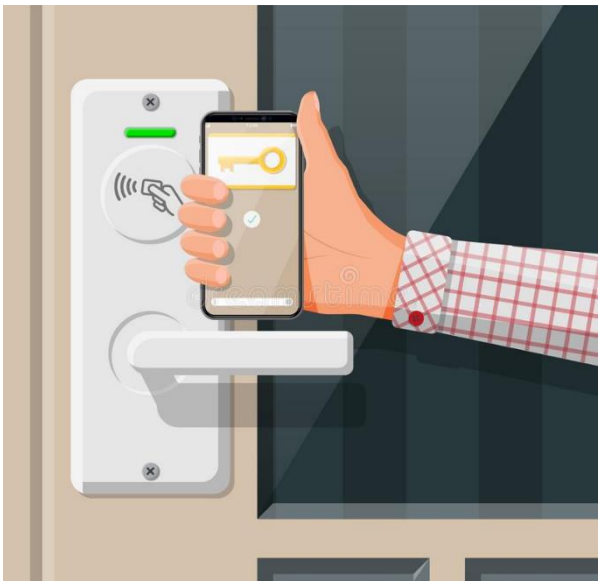
Assignment #5

How Many Possible Passwords?



You arrive at Computer Programming class and are surprised to find a password protected lock on the door. The password was emailed out to your but you lost it.

However, you do remember a few key facts about the password:

- It is a six-digit number.
- The value is within the range of 0 to 999999.
- Two adjacent digits are the same (like 22 in 1*22*345).
- Going from left to right, the digits *never decrease*; they only ever increase or stay the same (like 111123 or 135679).

Examples of good/bad passwords:

- 111111 meets these criteria (double 11, never decreases).
- 223*450* does not meet these criteria (decreasing pair of digits 50).
- 123789 does not meet these criteria (no double).

Your Mission:

Find out How many different passwords within the range given will meet the criteria?

Once you have figured out the answer, adjust your program so that a user can input any range they wish (example: 0 to 888888) and the program will tell them how many possible passwords exist in that new range.

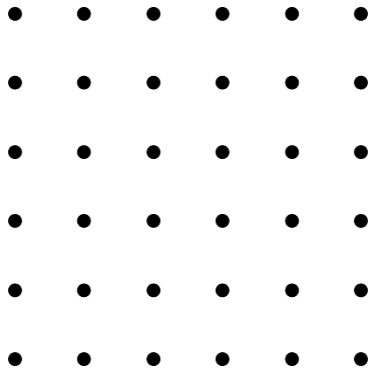
House Grid Deliveries

A worker from Amazon is delivering packages to an infinite **two-dimensional** grid of houses. He begins by delivering packages to the house at his starting location, and then a worker at headquarters calls and tells him where to move next.
His moves are *always exactly one house* to the north (N), south (S), east (E), or west (W). After each move, he delivers another package to the house at his new location.

However, worker at headquarters is new and sometimes gets the routes wrong, so the delivery person ends up visiting some houses more than once.

For examples:

Input: E
Output:
Diliver person goes to to ***2 houses***: one at the starting location, and one house directly to the east.



Input: N E S W N E
Output:
Makes 6 moves times but only hits ***4 houses*** located in a square.

Input: N S N S N S N S
Output:
move locations 8 times but ping pongs between on **2 houses**

Your Misson: create a program that can collect directions from a user and then *determine how many unique houses* were hit on the route.

Bike Shop Database

Create a table using a 2D list format

Create a computer program that can store and display information like the table shown below. You may change *information* stored in database but you must have at least 10 products and use must use the category columns show below.

Bicycle Inventory Data Table

Product Category	Product Name	Purchase Cost	Selling Price	Supplier	Quantity on Hand
Leisure	Blue Moon	$75.29	$105.41	Simpson's Bike Supply	4
Mtn.	Bluff Breaker	$375.00	$495.00	The Bike Path	3
Leisure	Breeze	$89.95	$130.95	The Bike Path	4
Leisure	Breeze LE	$109.95	$149.95	The Bike Path	5
Road	Classic 109	$207.49	$290.49	Bicyclist's Choice	7
Children	Coolest 100	$69.99	$97.98	Bicyclist's Choice	6
Mtn.	Eagle 1	$410.01	$574.01	Bike-One	1
Mtn.	Eagle 2	$401.11	$561.54	Bike-One	2
Mtn.	Eagle 3	$350.52	$490.73	Bike-One	5
Hybrid	Eagle 7	$150.89	$211.46	Bike-One	9
Road	Elegant 210	$281.52	$394.13	Bicyclist's Choice	7

When complete, your program should be able to do all the following:

1. Start with a display that says:

Welcome Employee! Please enter your password:
(your database must be password protected) and should only allow you to enter if you have the correct password)

2. Once a correct password has been entered, the program should ask the user to select from the following options:

1 - View an *individual* product's info by product name (user then enters product by name)
2 - View ***all*** products with just **selling price** and **quantity** on hand.
3 - View ***all*** products with all product information (in alphabetical order by product name).
4 - **Enter a sale** – This option will allow the user to input a sale and then the quantity on hand in the database will automatically be adjusted.
5 - **See finances** this option will let the employee's see how much money the store has maked in profit. This is a running total of: (the sale price of all the products that have been sold) minus (their purchase price).
6 – Any other cool features you wish.

All outputs must be displayed in a user friendly fashion that is easy to read.

Giving Directions Back Home

Jane's family has just moved to a new city and today is her first day of scho
list of instructions for walking from her home to the school. Each instruction describes a turn she must make. For example, the list

```
R
QUEEN
R
FOURTH
R
SCHOOL
```

means that she must turn right onto Queen Street, then turn right onto Fourth Street, then finally turn right into the school. Your task is to write a computer program which will create instructions for walking in the opposite direction: from her school to her home. The input and output for your program will be formatted like the samples below. You may assume that Jane's list contains at least two but at most five instructions, and you may assume that each line contains at most 10 characters, all of them capital letters. The last instruction will always be a turn into the "SCHOOL".

Sample Input 1

```
R
QUEEN
R
FOURTH
R
SCHOOL
```

Sample Output for Sample Input 1

```
Turn LEFT onto FOURTH street.
Turn LEFT onto QUEEN street.
Turn LEFT into your HOME.
```

Sample Input 2

```
L
MAIN
R
SCHOOL
```

Sample Output for Sample Input 2

```
Turn LEFT onto MAIN street.
Turn RIGHT into your HOME.
```