Net-centric Programming

Lab: Introduction to Golang

I. Objective:

- Get familiar with Go Programming Language

II. Questions:

1. (Hamming) Calculate the Hamming Distance between two DNA strands.

Your body is made up of cells that contain DNA. Those cells regularly wear out and need replacing, which they achieve by dividing into daughter cells. In fact, the average human body experiences about 10 quadrillion cell divisions in a lifetime!

When cells divide, their DNA replicates too. Sometimes during this process, mistakes happen and single pieces of DNA get encoded with incorrect information. If we compare two strands of DNA and count the differences between them, we can see how many mistakes occurred. This is known as the "Hamming Distance".

We read DNA using the letters C,A,G and T. Two strands might look like this:

```
GAGCCTACTAACGGGAT
CATCGTAATGACGGCCT
^ ^ ^  ^ ^    ^^
```

They have 7 differences, and therefore the Hamming Distance is 7.
The Hamming distance is only defined for sequences of equal length, so an attempt to calculate it between sequences of different lengths should not work.
You should generate 1,000 pairs of random DNA samples of a given length and then run the tests.

2. (Scrabble score) Given a word, compute the Scrabble score for that word.

Letter Values (ignore case):

Letter	Value
A, E, I, O, U, L, N, R, S, T	1
D, G	2
B, C, M, P	3
F, H, V, W, Y	4
K	5
J, X	8
Q, Z	10

Examples

"cabbage" should be scored as worth 14 points:
3 points for C
1 point for A, twice
3 points for B, twice
2 points for G
1 point for E
And to total:
3 + 2*1 + 2*3 + 2 + 1
= 3 + 2 + 6 + 3
= 5 + 9
= 14

Compute Scrabble score of a string containing multiple words

3. (Luhn) Given a number, determine whether or not it is valid per the Luhn formula.

The Luhn algorithm is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers

Validating a Number
Strings of length 1 or less are not valid. Spaces are allowed in the input but should be stripped before checking. All other non-digit characters are disallowed.

Example 1: valid credit card number
4539 3195 0343 6467

The first step of the Luhn algorithm is to double every second digit, starting from the right. We will be doubling
4_3_ 3_9_ 0_4_ 6_6_

If doubling the number results in a number greater than 9 then subtract 9 from the product. The results of our doubling:

8569 6195 0383 3437
Then sum all of the digits:

8+5+6+9+6+1+9+5+0+3+8+3+3+4+3+7 = 80
If the sum is evenly divisible by 10, then the number is valid. This number is valid!

Example 2: invalid credit card number
8273 1232 7352 0569
Double the second digits, starting from the right

7253 2262 5312 0539
Sum the digits

7+2+5+3+2+2+6+2+5+3+1+2+0+5+3+9 = 57

57 is not evenly divisible by 10, so this number is not valid.

4. (Minesweeper) Add the mine counts to a completed Minesweeper board.

Minesweeper is a popular game where the user has to find the mines using numeric hints that indicate how many mines are directly adjacent (horizontally, vertically, diagonally) to a square. (https://g.co/kgs/zEmYcs)

Given a rectangle minefield with an empty square is represented by a '.' and a mine is represented by '*'. Counts the number of mines adjacent to a given empty square and replaces that square with the count. If a given space has no adjacent mines at all, leave that square blank.

Create a random 20 x 25 minefield with 99 mines, then mark the board.
Example with 5x4 board:

```
.*.*.
..*..
..*..
.....
```

□

```
1*3*1
13*31
.2*2.
.111.
```

5. (Matching Brackets) Given a string containing brackets[ ], braces { }, parentheses ( ). Verify that all pairs are matched and nested correctly.
   Example: fmt.Println(a.TypeOf(xyz)){[ ]} □ correct

The end.