Session 7

Recap

In the previous session we covered the following:

- TDD - revisited
- Pair Programming
- Mobbing

Pair Programming - Continued

For the best part of the day we will be looking more at doing several challenges in a TDD pair programming manner.

Challenge: Football Score

Write a program that calculates the number of points obtained by a football team over a given season

A score is denoted by “X:Y” where X is the home score and Y is the away score. Assume we played our games at home all season.

Key:

Home Score		Away Score	Points
x	>	y	3
x	<	y	0
x	=	y	1

Your input can be assumed to be an array of 38 strings.

All score combinations should be covered.

Bonus: Allow for your program to let the user state which teams points are required, i.e. home teams or the away teams.

Challenge: Weighting

Nick has been attending a fitness club and every week weights are checked to see who the best and worst performers are. Nick's weight is 100kg and so his weight would be the worst given the others in the club have lost more than him. Luckily for Nick, he is in charge of gathering the group's weights this week, so he has come up with a way where the weights are given an ordering "weight" number such that his own weight is no longer last.

For example, the weight `99` will be given the "weighted" number of `18` (because 9 + 9 = 18). A weight of `100` will give you a "weighted" number of `1`, therefore 100 will come before 99.

A possible example is:

```
input: `56 65 74 100 99 68 86 180 90`
as ordered "weight" becomes `100 100 90 56 65 74 68 86 99`.
```

When two numbers have the same "weight", they are ordered as if they were strings, such that `100` comes before `180` because their ordered "weights" are `1 (for 100)` and `9 (for 180)`, and `180` would come before `90` since `180` comes before `90` when they are treated as strings.

Write a function that orders weights in order by their summed values in ascending order.

All numbers provided are positive integers but the list can also be empty.