

# Advent of Code

Lucas Larsson

Spring Term 2022

## Introduction

Pretty straight forward, not much to say regarding the assignment, fun and easy to do.

## Description

It is a prerequisite that the reader has read the story of the game. From there it's 2 rules for the game.

1. If the current value is greater than the previous then it is a plus one.
2. If it is lower, then it is 0 or nothing.

Part Two: simply put, instead of measuring for each element we measure the sum of three at a time.

## Implementation

Below is all the code needed to solve the problem, simply when a the second parameter on the first clause is an empty list we have traversed through the whole list. Otherwise compare the heads of the list if the the second is greater add one to the count that shall be returned otherwise call `measure/2` again.

`measure/2` is called with the list as the first parameter and the tail as the second so that we can compare the consecutive element.

```
def measure([_], []) do 0 end
def measure([h|t], [h2|t2]) do
  cond do
    h2 > h -> 1 + measure(t, t2)
    true -> measure(t, t2)
  end
end
```

The second part of the puzzle was just as easy as the first one, we already got a well detailed explanation of what we are supposed to do in the instructions. The way I did it follows as below.

```
def 3sum(list, list2) do
  [_|t]= list
  cond do
    Enum.at(list, 2) == nil -> list2
    true -> [ Enum.at(list,0) +
              Enum.at(list,1) +
              Enum.at(list,2) | 3sum(t, list2)]
  end
end
```

In simple terms the method `3sum/2` sums up the first three elements in the first parameter `list` and appends them to the second parameter `list2`.