

# Advent of Code 2022: Day 1 - Report

Adrian Hjert

17/Feb/2023

## Introduction

The aim of this report is to describe the thought process and the implementation of code that solves the problem presented by the Advent of Code: Day 1 from 2022. The task is rather simple; it is to take a large number of new line separated integers from a text file which signify how many calories of food different elves have with them on a trip. The task is to find how many calories the elf with the most calories of food has with them.

## Methods

First, we need to figure out how to read from the `.txt` file. In Elixir, this can be done using `File.stream!("fileName.txt")`. We also want to separate the integer values from the `.txt` file into chunks so we can easily add them up. For this, we use `Stream.chunk_while/4` which groups the values into chunks in a specified way, in this case we make it separate the values based on if there is an empty line before or after them or not. Then we add all the values in the chunk and move on to the next group of values. This continues until all the values from the `.txt` file have been separated into their respective chunks and added up. The following code snippet shows how this is implemented.

```
Stream.chunk_while(File.stream!("cals.txt"),
  0, fn "\n", acc ->
    {:cont, acc, 0} #0 is the initial value of the accumulation
  str, acc ->
    {:cont, String.to_integer(String.trim(str, "\n")) + acc}
  end, fn _acc -> {:cont, []} end
)
```

Now that the values have been summed up (hopefully) correctly, we need to make it easier to find the largest value and to double check that it is correct. For this we use

```
|> Enum.sort(:desc) |> IO.inspect()
```

This takes all our previous values thanks to the pipe operator and sorts them in a descending order, again, using the pipe operator, takes the sorted list of summed up values and prints them with **IO.inspect/2**. It is worth noting that this only prints some of the values as it has a max width of 80 characters by default, but for our purposes this does not need to be changed as we are only looking for the max value which should be first regardless. We can then use

```
|> Enum.take(1)
```

This, again, using the pipe operator will take our previous result and use it as a first input, so the function takes the first element in the sorted list and returns it as a list with 1 element. If we wanted to take the top 3 elements for example, we would replace the 1 with a 3 and it would return the first 3 elements as a list of 3 elements. Examples of this can be seen in the **Results and Conclusions** section.

## Results and Conclusions

As stated previously the **Enum.take/2** function will return and print the element(s) specified, so, taking

```
|> Enum.take(1)
```

will give the first element, which in our case is 70116. This is confirmed to be the correct answer for the generated list of values given to me for this assignment from the Advent of Code website. The number returned gives the amount of calories the elf carrying the most calories is carrying with them on their trip. Were we to replace the 1 with a 3 in the clause it would return [70116, 68706, 67760] which would then show how many calories the top 3 elves were carrying with them.