C\Cpp\Exercises\exercise\_20250922\_1314\solutn\_evaluatn.docx

This document is a semi-hiddem document for detailing evaluation, feedback, and results.

# Log

## 2025

### 2025 / October

#### 13/10/2025

Relevant text message exchange last week;

<SMS sent by me on Thursday 9/10/2025 9:27 AM>

Hi <Company’s Software Engineering Manager>,

Apologies that I didn't get back to you yesterday!

I haven't been able to dedicate as much time as I liked to the challenge. However, I did solve the very large array issue so that I could at least execute the program with a 10 character long combination, but then discovered fatal memory leak issues.

've also solved that but now my program is taking impossibly long to produce an answer. If it's ok with you, I wish to continue with the challenge so that I can reduce the calculation time and make it cleaner for peer review.

Are you able to provide a hint on how long the good solutions you've seen take to execute?

Kind regards,

<SchponzeePlusPlus>

<SMS sent by Company’s Software Engineering Manager on Thursday 9/10/2025 9:54 AM>

Hi <SchponzeePlusPlus>, good solutions will typically run in less than a second. Hint: look into 'memoisation' to avoid doing repeated calculations.

<SMS sent by me on Thursday 9/10/2025 9:58 AM>

Hi < Company’s Software Engineering Manager >, thanks for the hint, I will look into it!

* So the bottom line from this conversation is for me to investigate memoization
  + to reduce repetitive calculations
  + it is a concept term that I’ve never heard before, but upon initial searches for it, it does look like an important technique
  + I think my current brute force combination generator tactic is just too repetitive and inefficient, 20^10 different combinations being checked, most being invalid
    - Don’t know how much help it would be to keep optimising this method
    - one of the challenge authors are suggesting a different approach so it is worth following that advice

#### 07/10/2025

* Last week
  + On the 29th of September I sent an email that could be regarded as a partial submission
    - I attached my Attempt No. 1 of the solution and the complimenting solution justification
    - I asked them if there was a deadline for this task
      * The first email presenting the challenge inferred that they would accept submissions a week from that email
        + Did they expect candidates to take only a week to develop a solution ready for submission? I was not sure
    - I asked them whether I should continue with my solution or submit my work as is
      * I said that my solution is not currently at the standard I would've liked due to the following:
        + The algorithm cannot work with 10 digit sequences, only a maximum of 5 due to using large dynamic arrays
        + It is messy and not the easiest to read
        + I would still like to explore implementing advanced data structures and algorithms that I am not currently comfortable using but believe that they might make the program more efficient
        + Results from my attempt no. 1:

3 Digit Sequence, Valid Combinations: 188

4 Digit Sequence, Valid Combinations: 598

5 Digit Sequence, Valid Combinations: 1899

* I spoke to the software engineering manager of the company who presented me with this challenge on Wednesday the 1st of October
  + They gave me the following feedback on my code
    - For starters it does not return the correct value because it is not complete
    - If it did return the correct value and was submitted for further peer review it would be graded poorly
      * Too verbose (and I would personally add also too convoluted)
      * Too messy
      * That my variable names that, while meaningful, are too long and hard to read
      * Lots of repetitive code looks like it can be abstracted into reusable functions
      * Reckons that the chess move 01 and 02 don’t need to be done the way they are – better to abstract the difference (into an array ?)
      * Some functions are too long
        + Perhaps he was referring to the shift or increment functions
      * My attempt no. 1 is 845 lines in total, too long for the expected solution
        + The software engineering manager has seen solutions around 60 lines
        + Says that the problem is not complicated, that it should not be over 200 lines (200 lines maximum, the less the better)
        + That a non-explicitly stated criteria of the challenge is simplicity

Better code is simpler, simpler to read as well

That it is a simple algorithm that does not need to implement any object-oriented concepts

* + Hinted that the answer is over 1 million
  + Said that there was no concrete deadline for the challenge, it was more dependant on the first candidate/s to get it submitted
    - Open until the (junior) role is fulfilled
  + Checked in with me, wanted to make sure that I wasn’t wasting my time, but offered to give me another week for me to work on it and then check in there
* So based on this feedback, will be criteria for my attempt no. 2 (or 3)