(spp\_)misc\_code\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

#### 22/10/2025

7:55 PM AEDT

* The software engineering manager at the company who posted this challenge emailed me to inform me that they have now filled the vacant positions in the software team
  + He thanked me for my efforts and recent success this week in determining the correct answer
  + Upon receiving this news, I am moving the continuation of this exercise to my backlog
    - I committed and pushed my last changes to the active development branch, then PR’d and merged my changes to the main branch
    - I was looking forward to a peer review with some experienced software engineers so it's a shame that I wasn't ready in time
      * Achieving the 200 lines of code target was going to be a challenge for this week
      * I do see some value in still continuing with this challenge until it is peer-reviewable, but I will be moving this task into the backlog for the time being
        + I think I would like to attempt this code clean-up and refinement before seeking feedback on my code from someone/something like ChatGPT
  + I don't think I've ever attempted anything on this scale before, so I have some further up-skilling to do
    - I did learn how to implement several things like memoization, C++ standard arrays, and C++ standard vectors
    - I don't think my efforts have been all for naught, I challenged myself as an aspiring software engineer and I did solve the problem, so that does give me confidence going forward

#### 20/10/2025

* Today I completed Solution Attempt No. 3, which involved completing the top-down memoization method to calculate a total number of valid 10-key sequences, given the constraints specified by the coding challenge:
  + First the program generates a “possibilities raw” array, which finds all the possible key-to-key transitions (which was determined to be 60 (valid))
  + Then the program creates a “possibilities” vector, which is a filtered copy of the “possibilities raw” array, with only the (60) valid transitions
  + The program copies the possibilities vector into another vector, the “combination list” vector
    - The possibilities vector provides the first 2 characters of the combination
  + Using the possibilities vector, the combination list vector is resized and the next character in the combination is filled
    - The resize and fill operation occurs until all characters have been processed
    - As the character is iterated, the vowel check is performed, erasing combinations that have 3+ vowels
  + This approach was inspired by a Fibonacci solution that implemented top-down memoization
  + As of the solution attempt no. 3 development branch merge to main branch, my program’s answer was 1013398
    - This number was return upon the program’s first successful attempt to calculate the total number of 10 character combinations
    - I asked the Company’s Software Engineering Manager whether this value was correct, in which they responded that it indeed was
    - I also simultaneously asked ChatGPT to determine the total number by showing them the coding challenge, and it also returned the same number
  + Since the program now functions correctly and completely solves the problem, my focus is to prepare the code for peer review
    - This will be solution attempt no. 4
      * Changes in this attempt should not change the list of combinations nor the total number of combinations itself
      * As initially discussed below, the focus of this activity will be:
        + removing redundant code
        + modularising where possible
        + improving readability, clarity, and simplicity
        + reducing line count, ideally to below 200
        + improving efficiency if possible

#### 18/10/2025

Relevant text message exchange this week:

<SMS sent by me on Wednesday 15/10/2025 11:22 PM>

Hi <Company’s Software Engineering Manager>, I'm still working on my memoization based implementation, just a quick question regarding the chess moves, moves such as F to 1 , J to 3 , 2 to K , or 2 to O , are illegal ?

This is my current cache of valid transitions for reference:

AL

AH

BM

BK

BI

CN

CL

CF

CJ

DO

DM

DG

EN

EH

FC

FM

G2

GD

GN

H3

H1

HA

HK

HE

HO

I2

IB

IL

JC

JM

KB

KH

K2

LC

LA

LI

L3

MD

MB

MF

MJ

NE

NC

NG

N1

OD

OH

O2

1H

1F

1N

2I

2G

3J

3H

3L

Possibility Count: 56

<SMS sent by <Company’s Software Engineering Manager> on Thursday 16/10/2025 10:50 AM>

Hi <SchponzeePlusPlus>, yes, the moves F1… J3 etc (in the first part of your message) are valid. I didn't check the list you provided.

<SMS sent by me on Thursday 16/10/2025 12:05 PM>

Hi <Company’s Software Engineering Manager>, thank you for clarifying that, I'm glad I asked otherwise I would be short a few results.

I'm fairly confident with the other key, press transitions so all good in that regard!

#### 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)