to solve the game mastermind using quantum computing we will use these following steps

~~it we~~ first to explain the steps I will applying the an 3Qubit states

$|000\rangle$ $n = 3 \rightarrow N = 2^3 = 8$ states overall

so the keepers keeps a secret key

lets the secret key $= |010\rangle$

and now the guesser should pick a guess randomly from the $N = 4$ possible candidates

guess $= |111\rangle$. and now the guess grade $= 1$

1- for the first step we need to form a superposition of all candidates

$\rightarrow$

using (H) gate

$$|000\rangle \xrightarrow{H} \frac{1}{2\sqrt{2}}\Big(\overbrace{|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle+|111\rangle}^{\text{all candidtates}}\Big)$$

2-

now for the second steps we will add two states to this superposition $|000\rangle$ state for storing not-Xor state and $|00\rangle$ to store the grade of the candidate → now every candidate will be

$$\longrightarrow \underbrace{|000\rangle}_{\text{candidate}}\underbrace{|00\rangle}_{\substack{\text{grade,}\\ \text{comparing}\\ \text{to gues}}}\underbrace{|000\rangle}_{\substack{\text{not-Xor}\\ \text{of candidate}\\ \text{with guess}}}$$

$$\rightarrow \frac{1}{2\sqrt{2}}\Big(|000\rangle+\dots+|111\rangle\Big)|00\rangle|000\rangle$$

now let create the gate that does not-XOR between two states ~~and sums how many~~ and returns the resulting not-XOR state then sums the ~~number~~ 1 bits of that new state: lets call this gate as E

example:

candidate $|101\rangle$ ↘
$\boxed{E} \rightarrow |101\rangle$ and we have $\underset{|01\rangle}{\overset{2}{\shortparallel}}$ one bits
guess $|111\rangle$ ↗

now we apply this gate E for each candidate comparing with the guess state

$$\longrightarrow \frac{1}{2\sqrt{2}}(|000\rangle + \dots + |111\rangle)|00\rangle|000\rangle \xrightarrow{E}$$

$$\xrightarrow{E} \frac{1}{2\sqrt{2}}\Big(|000\rangle|00\rangle|000\rangle + \underbrace{|001\rangle}_{\text{cand.}}\underbrace{|01\rangle}_{\substack{\shortparallel \\ 1 \\ \text{grade}}}\underbrace{|001\rangle}_{\text{not-xor}} + |010\rangle|01\rangle|010\rangle + \dots\dots\Big)$$

3- now we need to create another gate P the gate will take the input

$|000\rangle\,|00\rangle\,|000\rangle$ and the grade of the guess

~~[illegible]~~

example

~~$|000\rangle|00\rangle|000\rangle$~~

$$|000\rangle|00\rangle|000\rangle \longrightarrow \boxed{P} \longrightarrow |100\rangle|01\rangle|100\rangle$$

$$gg = \text{guess grade} = 1 \longrightarrow \boxed{P}$$

checks the difference of grades

$|000\rangle\,|00\rangle\,|000\rangle$

$\quad\quad\;\; \parallel$

$\quad\quad\;\; 0 - gg = -1$ so if minus 1s

the difference it flips the first one from the left bit in not-xor until the diff. is 0

If the difference is positive it flips the first 1 bit from the left in not-xor until the difference is 0 in b

→ ~~the~~ applying the P gate will make us have less candidates call them b now if the number of ~~new~~ the states after P gate is only 1 state then we found the secret if not the we need to continue by taking another guess from the original 16 candidates ~~a~~ but we need to take a candidate that it's grade ~~be~~ is not the same as the initial guess's grade now we callculat the grade of the new guess comparing it to the secret key now we ~~do~~ input the new states b and the new guess and it's grad to P which will result in shortening the number of states in b ~~b~~ so repeat this process untill b ~~b~~ has only 1 state after using a new guess and ~~i~~ inputting b ~~b~~ and the guess to f each time

when we reach to only 1 state the we found the secret key.

<u>note:</u> it is very important ~~#~~ that when choosing a new guess from the original $16 = 2^3 = 2^n$ states we have to make sure that the new guess's ~~#~~ score comparing it with the older guess is not the original guess's score