Puzzle no 3: Padlock

I wrote the solution of the given padlock problem using python. This program checks the given conditions with the numbers starting from zero (0) till the required number is reached.

The solution I got was (617) in the given padlock problem.

# Solving the puzzle 3 padlock problem

def checknumbers(number, con, correct\_digit, correct\_position):

count\_correct\_digit = 0

count\_correct\_position = 0

checknumber = str(number).zfill(len(con))

for i in range(len(con)):

if checknumber[i] in con:

count\_correct\_digit +=1

if checknumber[i] == con[i]:

count\_correct\_position += 1

if correct\_digit == count\_correct\_digit and correct\_position == count\_correct\_position:

return True

def solution(con1:str, con2:str, con3:str, con4:str, con5:str):

#con1 -> one digit right but in wrong place

#con2 -> one digit right and in right place

#con3 -> two digits correct but in wrong place

#con4 -> all digits are wrong place

#con5 -> one digit right but in wrong place

start = 0

for number in range(1000):

if checknumbers(number,con1, 1, 0) and checknumbers(number,con2, 1, 1) and checknumbers(number,con3, 2, 0) and checknumbers(number,con4, 0, 0) and checknumbers(number,con5, 1, 0):

print(f'The number is {number}')

break

solution("147", "189", "964", "523", "286")

Puzzle no 4: Padlock problem

I wrote the solution of the given padlock problem using python. This program checks the given conditions with the numbers starting from zero (0) till the required number is reached.

The solution I got was (042) in the given padlock problem.

# Solving the puzzle 4 padlock problem

def checknumbers(number, con, correct\_digit, correct\_position):

count\_correct\_digit = 0

count\_correct\_position = 0

checknumber = str(number).zfill(len(con))

for i in range(len(con)):

if checknumber[i] in con:

count\_correct\_digit +=1

if checknumber[i] == con[i]:

count\_correct\_position += 1

if correct\_digit == count\_correct\_digit and correct\_position == count\_correct\_position:

return True

def solution(con1:str, con2:str, con3:str, con4:str, con5:str):

#con1 -> one digit right but in right place

#con2 -> one digit right and in wrong place

#con3 -> two digits correct but in wrong place

#con4 -> all digits are wrong place

#con5 -> one digit right but in wrong place

start = 0

for number in range(1000):

if checknumbers(number,con1, 1, 1) and checknumbers(number,con2, 1, 0) and checknumbers(number,con3, 2, 0) and checknumbers(number,con4, 0, 0) and checknumbers(number,con5, 1, 0):

print(f'The number is {str(number).zfill(len(con1))}')

break

solution("682", "614", "206", "738", "380")