Puzzle no 3: Padlock 1

The solution of the given padlock problem was done 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 1 problem

def checknums(num, con, right\_number, right\_pos):

count\_right\_number = 0

count\_right\_pos = 0

checknum = str(num).zfill(len(con))

for i in range(len(con)):

if checknum[i] in con:

count\_right\_num +=1

if checknum[i] == con[i]:

count\_right\_pos += 1

if right\_num == count\_right\_num and right\_pos == count\_right\_pos:

return true

def numbers(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 checknums(num,con1, 1, 0) and checknums(num,con2, 1, 1) and checknums(num,con3, 2, 0) and checknums(num,con4, 0, 0) and checknums(num,con5, 1, 0):

print('The required solution is {num}')

break

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

Puzzle no 3: Padlock 2 problem

The solution of the given padlock problem was done 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 3 padlock 2 problem

def checknums(num, con, right\_num, right\_pos):

count\_right\_num = 0

count\_right\_pos = 0

checknum = str(num).zfill(len(con))

for i in range(len(con)):

if checknum[i] in con:

count\_right\_num +=1

if checknum[i] == con[i]:

count\_right\_pos += 1

if right\_num == count\_right\_num and correct\_pos == count\_right\_pos:

return true

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

#con1 -> one digit is right and in its place

#con2 -> one digit is right but in the wrong place

#con3 -> two digits are right but both are in the wrong place

#con4 -> all digits are wrong

#con5 -> one digit is right but in the wrong place

start = 0

for num in range(1000):

if checknums(num,con1, 1, 1) and checknums(num,con2, 1, 0) and checknum(num,con3, 2, 0) and checknum(num,con4, 0, 0) and checknum(num,con5, 1, 0):

print('The required number for padlock is {str(num).zfill(len(con1))}')

break

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