1. Soln

file = open("csv-sample.csv")

reader = csv.reader(file)

lines= len(list(reader))

1. Soln

I am not at all familiar with docker

1. Soln

Puzzle 1. 679

The code:

from itertools import count

def checknr(num, checknumber, nr\_correct, nr\_correct\_position):

    count\_nr\_correct=0

    count\_nr\_correct\_position =0

    lnumber = str(num).zfill(len(checknumber))

    for i in range(len(checknumber)):

        if lnumber[i] in checknumber:

            count\_nr\_correct += 1

        if lnumber[i] == checknumber[i]:

            count\_nr\_correct\_position += 1

    return nr\_correct == count\_nr\_correct and \

        nr\_correct\_position == count\_nr\_correct\_position

for cnum in range (1000):

    if checknr(cnum, '147', 1,0 ) and \

        checknr(cnum, '189', 1,1 ) and \

        checknr(cnum, '964', 2,0 ) and \

        checknr(cnum, '523', 0,0 ) and \

        checknr(cnum, '286', 1,0 ):

        print("code: "  + str(cnum).zfill(3))

logic:

147 one number is correct and in wrong place - so all three num have possibilities.

189 one num is correct and in place- this eliminates ‘1’

964 two is correct but in wrong place- here we find 9 is the number and its position from 2nd

523 all are incorrect-

286 one is correct in wrong place- 8 is eliminated from 2nd , possible num is 6 as 2 is wrong from above. From ‘964’ 6 is correct but in wrong place and from this 6 is correct but in wrong place. So we have ‘6-9’ so the remaining is 7 from ‘147’.

And is 679.

Puzzle 2. 042

from itertools import count

def checknr(num, checknumber, nr\_correct, nr\_correct\_position):

    count\_nr\_correct=0

    count\_nr\_correct\_position =0

    lnumber = str(num).zfill(len(checknumber))

    for i in range(len(checknumber)):

        if lnumber[i] in checknumber:

            count\_nr\_correct += 1

        if lnumber[i] == checknumber[i]:

            count\_nr\_correct\_position += 1

    return nr\_correct == count\_nr\_correct and \

        nr\_correct\_position == count\_nr\_correct\_position

for cnum in range (1000):

    if checknr(cnum, '682', 1,1 ) and \

        checknr(cnum, '614', 1,0 ) and \

        checknr(cnum, '206', 2,0 ) and \

        checknr(cnum, '738', 0,0 ) and \

        checknr(cnum, '380', 1,0 ):

        print("code: "  + str(cnum).zfill(3))

logic:

‘682’ one num is correct and in position so we have possibilities for all 3 numbers to be correct.

‘614’ this sliminates the 6 as it cant be in the right and wrong place at the same time.

‘206’ since 6 is sliminated 2 and 0 are correct. From ‘682’ 2 is in correct position.

‘738’ all are incorrect.

‘380’ since 38 are eliminated from above 0 is correct. From ‘206’ 0 is in incorrect position and its in incorrect position over here too. So we have ‘0-2’ so far.

From ‘614’ one number is correct and in incorrect position, 6 is eliminated and the position where “1” is in position, it eliminates 1, so 4 is the ans, thus the code is 042.