Puzzle no 1: Count the number of rows in the given csv file

The source code is written in python and it is as follows:

import csv

csv\_file = open("puzzlesV2/csv-sample.csv")

csv\_read = csv.reader(csv\_file)

rows\_count = 0

for i in csv\_read:

rows\_count+=1

print(f'The number of rows present in the given csv file is {rows\_count}')

Puzzle no 2: Build a docker image

I created a simple RESTAPI based on django rest framework and it returns the name and email in response. Please enter localhost:8000/api/email in the browser after you run the docker image.

The link of my docker image in docker hub is : <https://hub.docker.com/repository/docker/khsasbin/simple-docker-application>

In order to pull the docker container and run follow the following commands:

| docker pull khsasbin/simple-docker-application:latest  docker run -it --rm -p 8000:8000 khsasbin/simple-docker-application:latest |
| --- |

Click the link [http://localhost:8000/api/email/](http://0.0.0.0:8000/api/email/) to view the response

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")