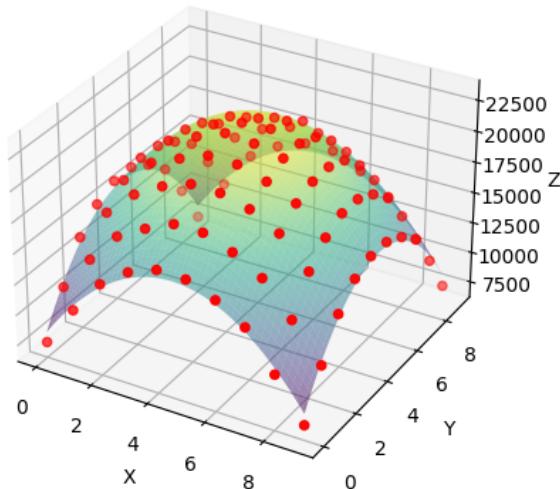


# Yu-Chueh Wang Engineering Portfolio

## table of content

Coding project 1 - Battleship with AI.....	p.1-3
Analytical project 1 - Winning percentage analysis.....	p.4
Analytical project 2 - Volleyball database.....	p.5-6
Solidworks project 1 - Gearbox.....	p. 7
Circuit project 1 - Halfwave rectifier/Diode Clipper.....	p 8 -10
Coding competition - Code.golf / Advent of Code.....	p 11
Automation Project 1 - Digits Solver.....	p 12



battleship AI training. probability analysis.

## Battleship AI with tensorflow and keras

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from replit import db
from sklearn.model_selection import train_test_split
import kerastuner
from kerastuner.tuners import RandomSearch

with open("z.npy","rb") as file:
    in_flat = np.load(file)

input_data = np.array([np.array([int(_) for __ in in_flat[_].split(",")])
                      .reshape((10,10)) for _ in range(len(db))])

with open("input_data.npy","rb") as file:
    out_flat = np.load(file)

output_data = np.array([np.array([int(_) for __ in out_flat[_].split(",")]
                                 .reshape((10,10)) for _ in range(len(db))])

input_data1 = input_data == 1
input_data2 = input_data == 2
input_data3 = input_data < 0

# Split the data into training, validation, and test sets
# train_input1, test_input1, train_input2, test_input2, train_input3, test_input3, train_output,
test_output = train_test_split(
#     input_data1, input_data2, input_data3, output_data, test_size=0.2, random_state=42)
train_input1, val_input1, train_input2, val_input2, train_input3, val_input3, train_output, val_output
= train_test_split(
    input_data1, input_data2, input_data3, output_data, test_size=0.1, random_state=42)

# Define a Hypermodel
class MyHyperModel(kerastuner.HyperModel):
    def build(self, hp):
        inputs = []
        for _ in range(3): # Change this based on the number of input arrays
            inputs.append(keras.Input(shape=(10, 10, 1)))

        conv_outputs = []
        for index, input_layer in enumerate(inputs):
            conv_layer = layers.Conv2D(hp.Int(f'filters_{index}', min_value=16, max_value=64, step=16),
                                      kernel_size=(3, 3), activation='relu')(input_layer)
            max_pooling_layer = layers.MaxPooling2D(pool_size=(2, 2))(conv_layer)
            conv_outputs.append(max_pooling_layer)

        concatenated = layers.concatenate(conv_outputs)

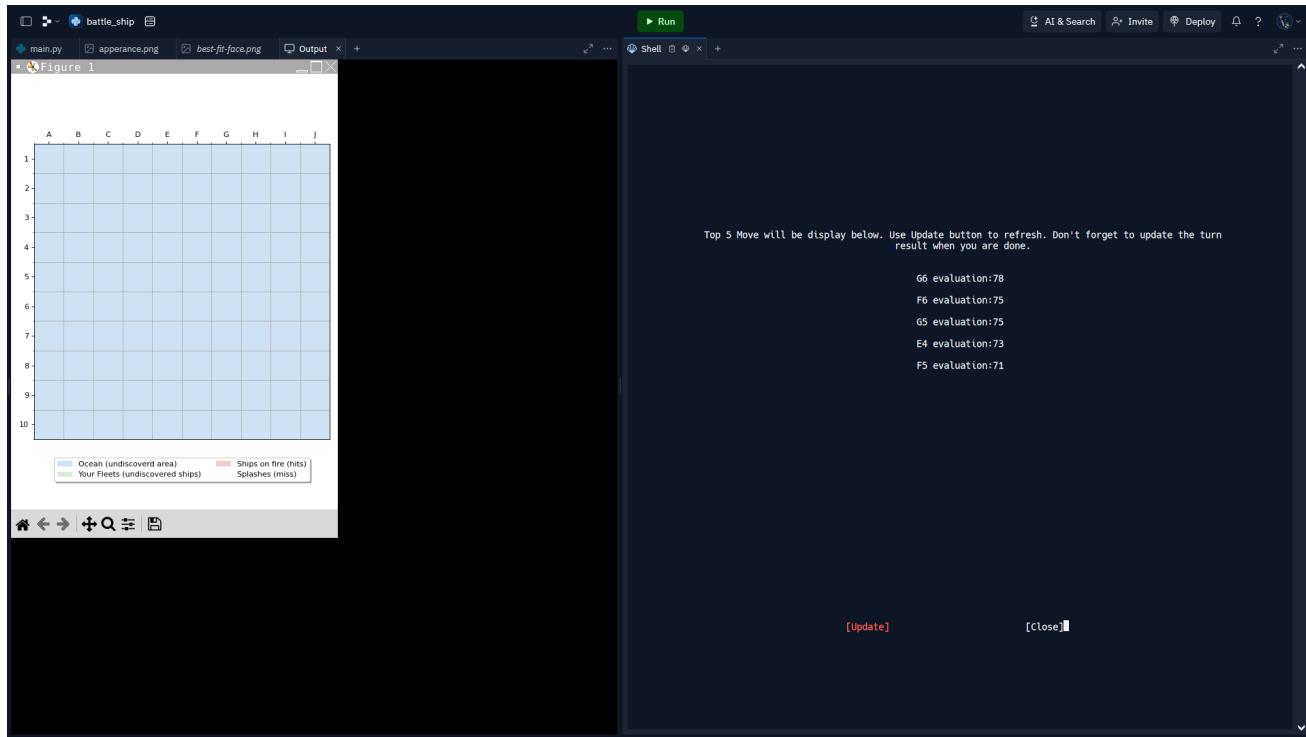
        flattened = layers.Flatten()(concatenated)
        dense_1 = layers.Dense(hp.Int('units_1', min_value=64, max_value=256, step=64),
                               activation='relu')(flattened)
        dropout_1 = layers.Dropout(hp.Float('dropout_1', min_value=0.2, max_value=0.5, step=0.1))
(dense_1)
        dense_2 = layers.Dense(hp.Int('units_2', min_value=32, max_value=128, step=32),
                               activation='relu')(dropout_1)
        dense_output = layers.Dense(10 * 10, activation='linear')(dense_2)
        reshaped_output = layers.Reshape((10, 10))(dense_output)

        model = keras.Model(inputs=inputs, outputs=reshaped_output)

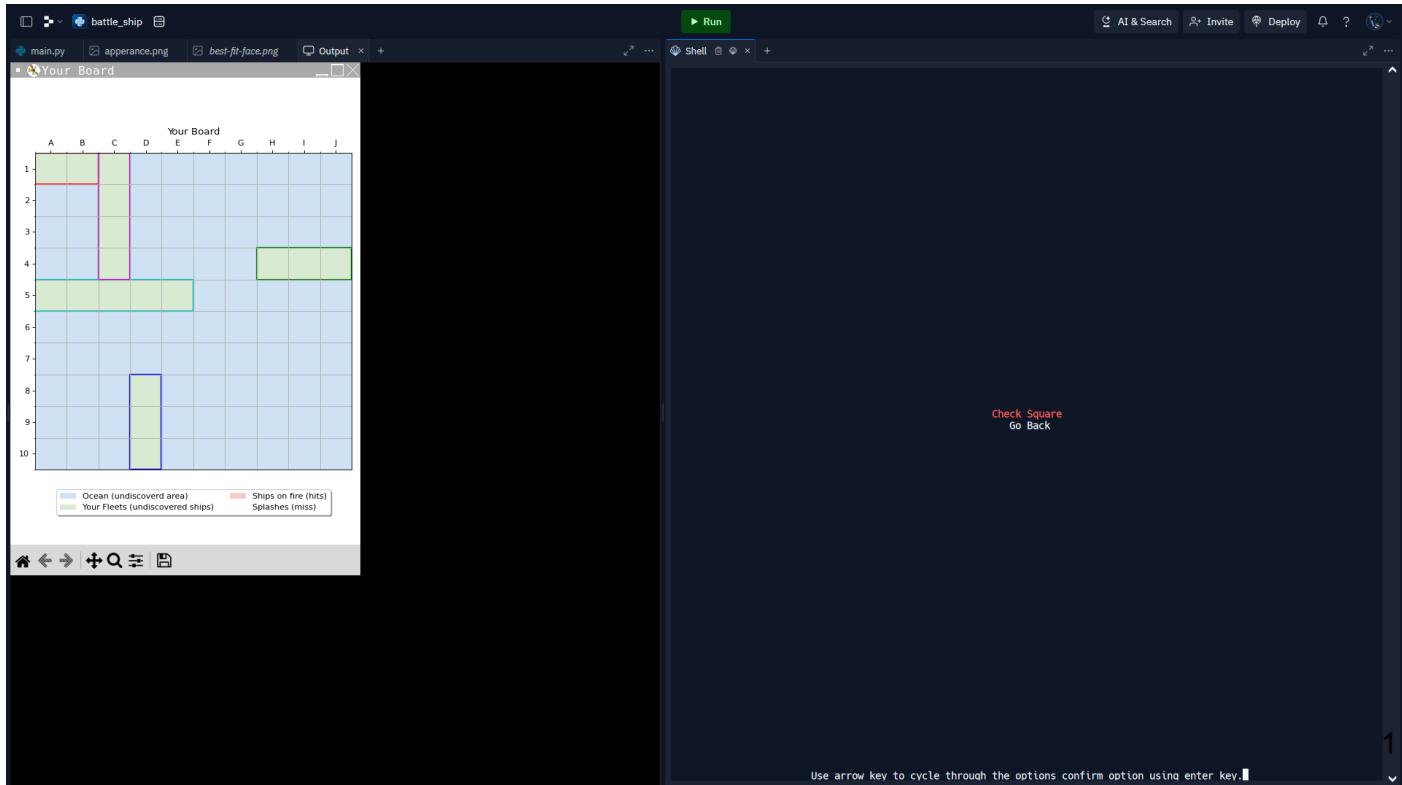
        hp_learning_rate = hp.Choice('learning_rate', values=[1e-2, 1e-3, 1e-4])

        model.compile(optimizer=keras.optimizers.Adam(learning_rate=hp_learning_rate),
                      loss='mean_squared_error',
                      metrics=['accuracy'])
        return model

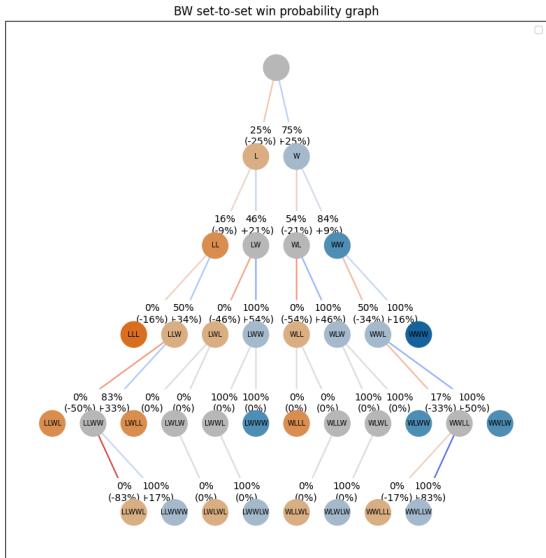
# Create a tuner
tuner = RandomSearch(
    MyHyperModel(),
    objective='val_loss'
```



## user interface for the AI

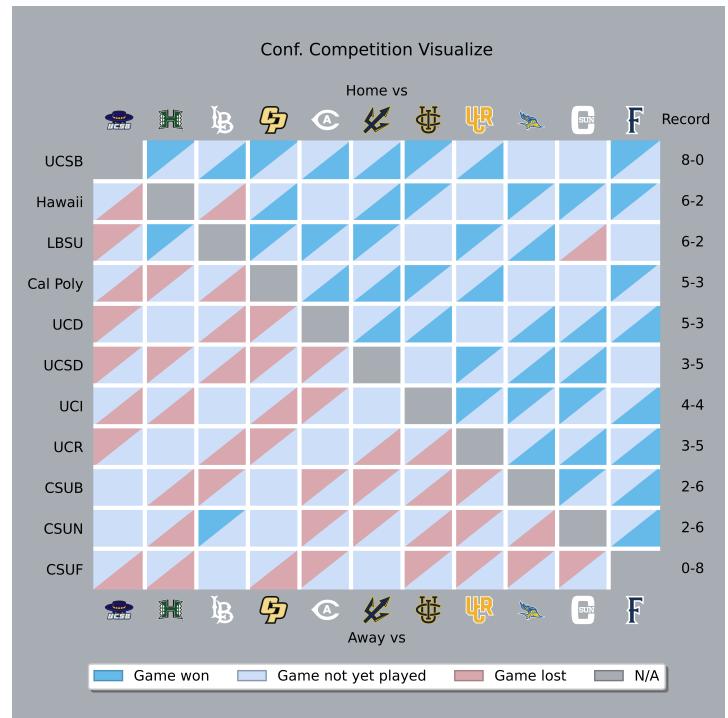


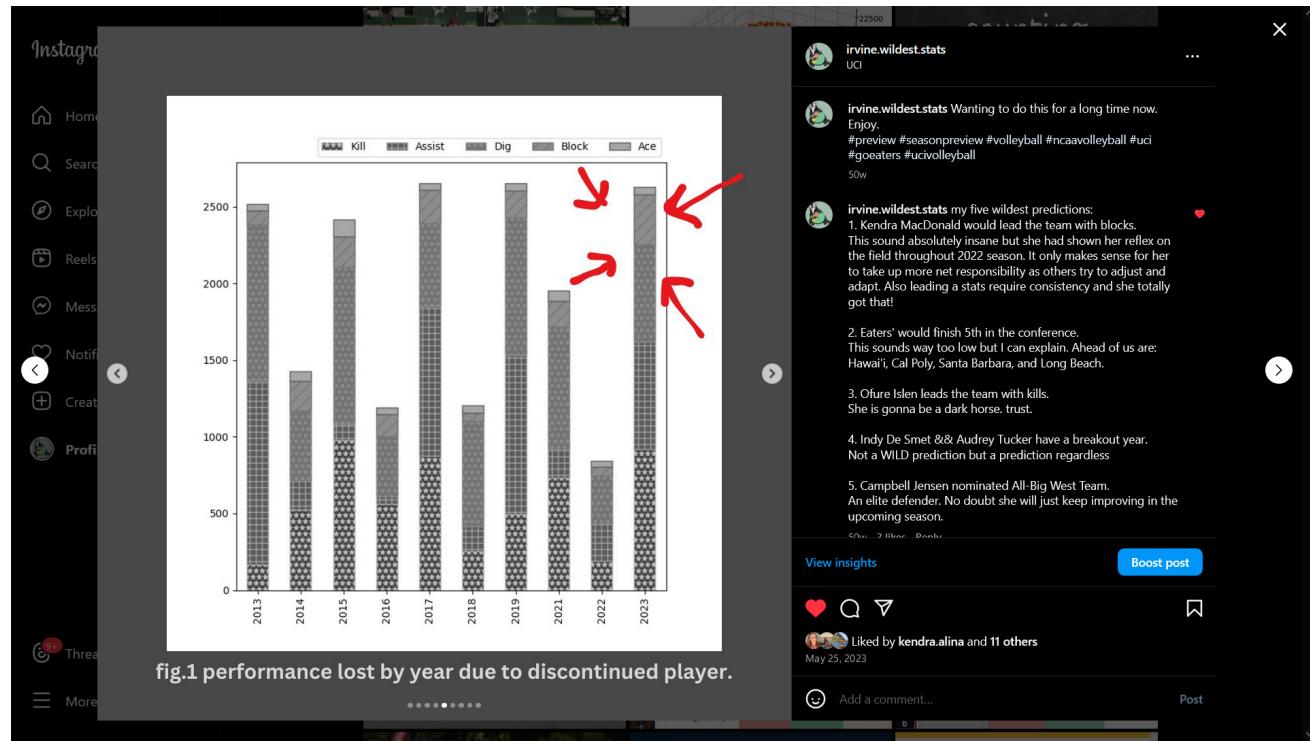
# sport analytical analysis



volleyball probabilistic events analysis in tree diagram

BW head to head match-up  
advantage/disadvantage analysis





## pre-season analysis and breakdown

### Top Players

Search by Name: Stats: Sort by: Results per page:

Enter player name... Kills Name 5

Conferences/Schools: Positions(color coded): Years:

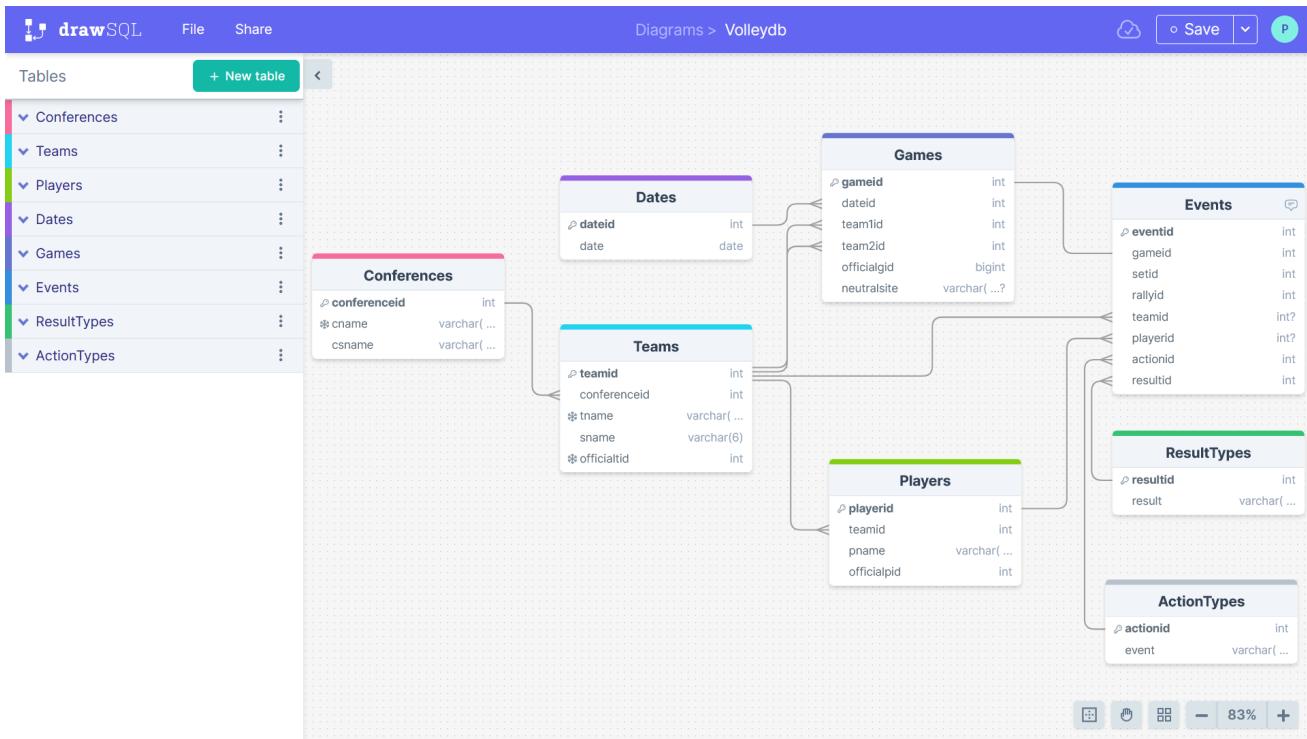
filter school filter position filter year

Collapse All

- A'Nylah Cobb #11
- A'siya Magazine #11
- Aaliyah Byers #11
- Aaliyah Fasavalu #18
- Aaliyah Mitchell #8

First 1 2 3 4 5 Last

volleyball database (under-construction) for querying and data analysis



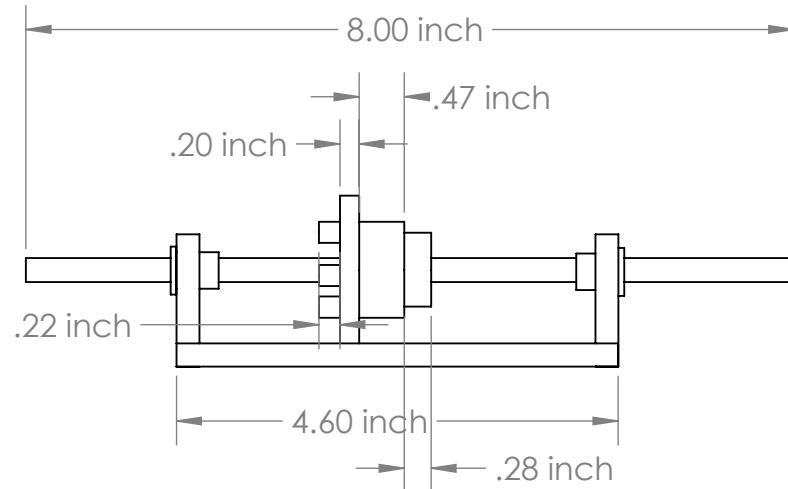
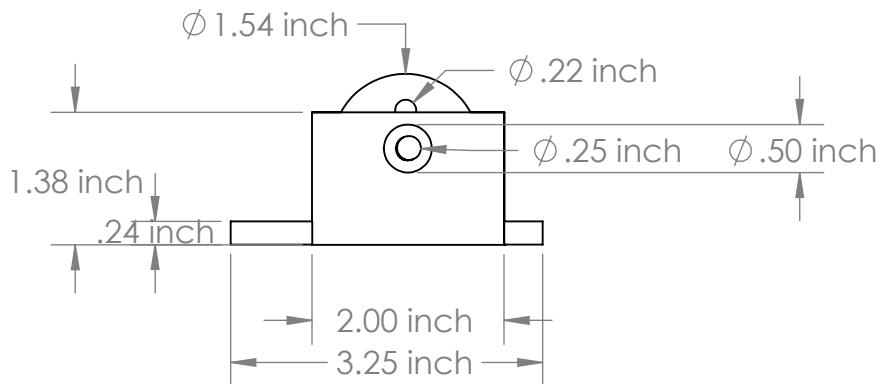
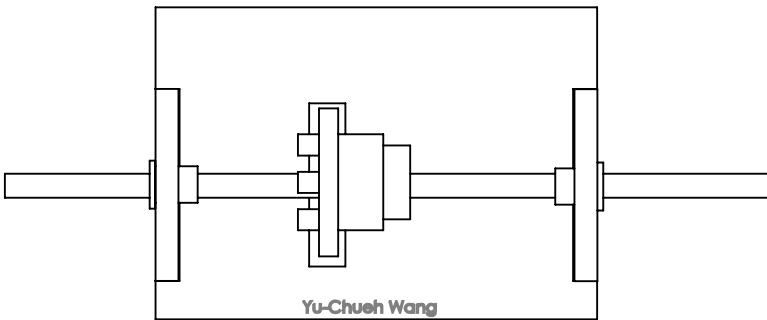
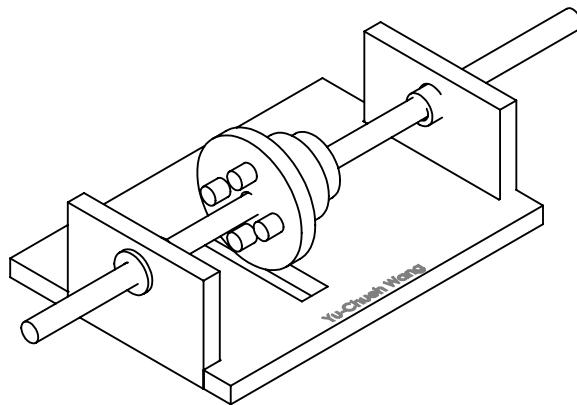
data base structure for the volleyball database

2

1

B

B



A

A

PROPRIETARY AND CONFIDENTIAL  
THE INFORMATION CONTAINED IN THIS  
DRAWING IS THE SOLE PROPERTY OF  
<INSERT COMPANY NAME HERE>. ANY  
REPRODUCTION IN PART OR AS A WHOLE  
WITHOUT THE WRITTEN PERMISSION OF  
<INSERT COMPANY NAME HERE> IS  
PROHIBITED.

		UNLESS OTHERWISE SPECIFIED:				
		DIMENSIONS ARE IN INCHES TOLERANCES: FRACTIONAL $\pm$ ANGULAR: MACH $\pm$ BEND $\pm$ TWO PLACE DECIMAL $\pm$ THREE PLACE DECIMAL $\pm$		DRAWN	NAME	DATE
		INTERPRET GEOMETRIC TOLERANCING PER: MATERIAL		CHECKED		
NEXT ASSY		FINISH		ENG APPR.		
APPLICATION		DO NOT SCALE DRAWING		MFG APPR.		

Q.A.	COMMENTS: Build by Yu-Chueh Wang
------	----------------------------------

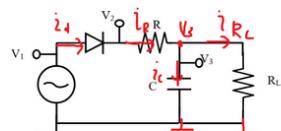
TITLE:

# Gear Box

SIZE	DWG. NO.	REV
A	HW1	
SCALE: 1:2	WEIGHT:	SHEET 1 OF 1



iii



$$i_B = i_C + i_{R_L}$$

$$\frac{V_3 - V_L}{R} = C \frac{dV_3}{dt} + \frac{V_3}{R_L}$$

ON

$$V = V_f + V_2$$

$$i_d > 0$$

$$i_R = 0$$

$$V_2 = V_3$$

OFF

$$V_1 < V_f$$

$$i_d = 0$$

ON

$$\frac{V_3 - V_L}{C_R} = \left( \frac{AV_3}{RL} \right) + \frac{V_3}{RL}$$

$$\frac{V_3 - V_L}{C_R} = \left( \frac{AV_3}{RL} \right) + \frac{V_3}{CRL}$$

$$\frac{V_3}{C_R} - \frac{V_L}{C_R} = \frac{dV_3}{dt} + \frac{V_3}{CRL}$$

$$\frac{dV_3}{dt} - \frac{V_3}{C_R} + \frac{V_3}{CRL} = -\frac{V_L}{C_R}$$

$$V_3 = K_1 e^{-\alpha t} + \frac{\beta}{\alpha} \quad V_3(0) = 0$$

$$= \frac{R}{\alpha} \cdot (1 - e^{-\alpha t}) \quad K_1 = \frac{R}{\alpha}$$

$$= \left( \frac{-\frac{V_L}{C_R}}{\frac{-L}{C_R} + \frac{1}{CRL}} \right) \cdot \left( 1 - e^{-\left( \frac{L}{C_R} + \frac{1}{CRL} \right)t} \right)$$

$$\xrightarrow{\text{Plug in}} V_3(0.5) = 3.14$$



```
"""
explanation:
while loop generate candidate. sort. then check if candidate is a prime.
further explanation:
since candidate must be in ascending order there's only 2**9(512) options. biggest being 123456789 and
each character can either be present or absent and thus why 2**9.
the while loop create candidate by converting i(2~2**9) to binary and use 0 and 1 to represent the absent
or present of the 9 numbers.
as of 01/02/2024 the code rank 35th in bytes and chars.
"""

i,s,d=1,"123456789",[]
while i<512:d+=[int("".join(s[n]for n,i in enumerate(bin(i)[:1:-1])if i=="1"))];i+=1
for i in sorted(d)[1:]:
    if all(i%j for j in range(2,i)):print(i)
```

## using innovating way to find ascending primes with minimal code! (code golf)



```
from collections import namedtuple

Monkey = namedtuple("Monkey","id items op mod success fail")

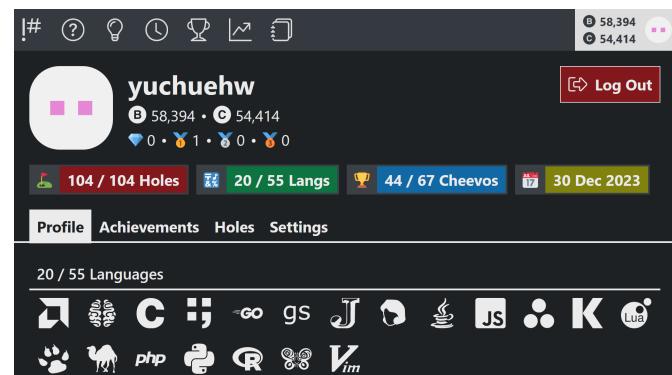
def question_input():
    return [
        Monkey(0,[73, 77],lambda x:x*5,11,6,5),
        Monkey(1,[57, 88, 80],lambda x:x*5,19,6,0),
        Monkey(2,[61, 81, 84, 69, 77, 88],lambda x:x*19,5,3,1),
        Monkey(3,[78, 89, 71, 60, 81, 84, 87, 75],lambda x:x*7,3,1,0),
        Monkey(4,[60, 76, 90, 63, 86, 87, 89],lambda x:x*2,13,2,7),
        Monkey(5,[88],lambda x:x+1,17,4,7),
        Monkey(6,[84, 98, 78, 85],lambda x:x*x,7,5,4),
        Monkey(7,[98, 89, 78, 73, 71],lambda x:x+4,2,3,2),
    ]

    BORED = lambda x:x//3
    POST = lambda x:x%(11*19*5*3*13*17*7*2)
    NONE = lambda x:x

def sim(monkeys,BORED,POST,rounds):
    inspected = [0]*len(monkeys)

    for i in range(rounds):
        for monkey in monkeys:
            while monkey.items:
                inspected[monkey.id] += 1
                item = monkey.items.pop(0)
                item = monkey.op(item)
                item = BORED(item)
                post = POST(item)
                if item % monkey.mod == 0:
                    monkeys[monkey.success].items.append(post)
                else:
                    monkeys[monkey.fail].items.append(post)
    return sorted(inspected)[-1] * sorted(inspected)[-2]

print(sim(question_input(),BORED,NONE,20))
print(sim(question_input(),NONE,POST,10000))
```



code.golf profile:<https://code.golf/golfers/yuchuehw>  
repo:<https://github.com/yuchuehw/CodeGolf>

## simulating monkey games in python (advent of code)

see more at my github page:  
<https://github.com/yuchuehw/Advent-Of-Code-2022/>

[README](#) [Code of conduct](#) [MIT license](#) [Security](#)

# Digits Solver

English • 繁體中文 • 简体中文 • 日本語 • Español • Français • Italiano • Deutsche • Русский

Welcome to Digits Solver, the ultimate Python companion for conquering the mind-bending [Digits](#) puzzle game, developed by The New York Times. Dive into a captivating world of numerical challenges and master the art of strategic manipulation. With Digits Solver, you'll strategically manipulate a set of starting digits using mathematical operations to reach the elusive target digit. Its powerful algorithm and meticulous analysis empower you to swiftly unravel each puzzle, delivering step-by-step solutions with unwavering precision. Elevate your puzzle-solving prowess and unlock the secrets hidden within the digits. Get ready for an exciting journey to become a Digits master!

Python application [passing](#) CodeQI [passing](#) pylint score [9.67](#)

Python Selenium TimeShift.js contributions welcome license MIT code style black hits 431

## Demo

See the algorithm in action by clicking the green run button after being redirected:

repo:<https://github.com/yuchuehw/DigitsSolver?tab=readme-ov-file>  
speedrun video: <https://www.youtube.com/watch?v=se2OdZnEHHA>

The screenshot shows a YouTube video player with a dark theme. The video content is a digital calculator interface from the Digits Solver project. The calculator has a numeric keypad and operators (+, -, ×, ÷). A target number '62' is displayed above the keypad. Below the keypad, there are four circles containing the numbers 6, 2, 3, and 4. A green circle labeled '10' is highlighted. At the bottom of the calculator interface, there is a 'Solve!' button. To the left of the calculator, there is a terminal window showing command-line output. The video title is 'Digits World Record! 9:58 April All Level Speed Run'. The video has 13 views and was posted 11 months ago. The channel has 1 subscriber.

Longest Plank EVER - Guinness World Records

5 Signs of an Inexperienced Self-Taught Developer (and how to fix them)

20 Programming Projects That Will Make You A God At Coding

Snapdragon just obsoleted Intel and AMD

AI Learns How To Play The Chrome Dinosaur Game

Why It's Almost Impossible to Run 100 Meters In 9 Seconds...

The SAT Question Everyone Got Wrong

Almost EVERYONE is Wasting Money on Dash Cams.