def main ():

print()

print(". . 0 . . 0 . . . . .")

print(". . 0 . . 0 . . 0 . .")

print(". 0 0 0 0 0 0 0 0 0 .")

print(". . 0 . . 0 . . 0 . .")

print(". . 0 . . 0 . . 0 . .")

print("0 0 0 0 . 0 . 0 . . .")

print(". . 0 . . 0 . 0 . . .")

print(". . 0 . 0 0 0 0 0 0 0")

print("0 0 0 . . 0 . 0 . . .")

print(". . 0 . . 0 . 0 . . .")

import random

word10 = ["avengement", "beekeeping", "blethering", "cheapening", "breakevens", "fledgeling", "greaseband", "overseeing", "previewing", "sweetening"]

word101 = ["checkpoint", "cherrylike", "creativity", "skepticism", "theologian", "energizing", "identities", "inequality", "beekeeping", "pressurise"]

word9 = ["beekeeper", "deflected", "henpecked", "requested", "sequences", "weekender"]

word7 = ["kittens", "liftoff", "mittens", "dietary", "dieting", "vintage"]

word5 = ["actor", "after", "batch", "eaten", "gates", "hatch", "match", "mitre", "metro", "octet", "paths", "pitch", "watch", "water"]

word4 = ["bead", "beam", "beer", "bees", "best", "beta", "fear", "fees", "felt", "fern", "feta", "geek", "gels", "gene", "heal", "herd", "heed", "hens", "hero", "lead", "leap", "less", "meal", "melt", "memo", "tech"]

word41 = ["meet", "teem", "heed", "geek", "fees", "feel", "bees", "beer", "need", "reed", "seep", "step", "tees", "week", "weed"]

word3 = ["ban", "bun", "can", "fan", "fun", "run", "sun", "ten", "tin", "ton", "van"]

print()

g = (random.choice(word3))

b = (random.choice(word4))

f = (random.choice(word41))

a = (random.choice(word5))

h = (random.choice(word7))

e = (random.choice(word9))

c = (random.choice(word10))

d = (random.choice(word101))

import random

#print ( list ("\*"\*10) + c + list("\*"\*4) + list("\*"\*30) + g + list("\*"\*20) + list("\*"\*10) + a + ("\*"\*26) )

from itertools import zip\_longest

#lets = list("\*"\*20) + list(g) + list("\*"\*24) + list(c) + list("\*"\*19) + list(a) + list("\*"\*24)

lets = list("."\*108)

def table(it, rows):

return zip\_longest(\*[it[i:i+rows] for i in range(0, len(it), rows)], fillvalue=" ")

import re

import string

alphabet = string.ascii\_letters + string.punctuation

for q in range (15):

import re

lets[74:79:1] = a

lets[80:83:1] = b

lets[50:59:1] = d

lets[20:30:1] = c

lets[12:102:10] = e

lets[47:117:10] = h

lets[8:38:10] = g

lets[5:45:10] = f

#lets[81:85:1] = j

import random

import string

#print (list (filter(lambda x:x in e,b)))

#t = print (re.search ("[\d\w\s]", "---"+ e[3] +"---"))

#u = print (re.search ("[\d\w\s]", "--"+ b[2] +"--"))

#print (list(filter(lambda x:x in e[3], b[2])))

values = word4

for x in range (0, 11):

letter = random.choice(values)

# print(letter)

#rand = ["a", "e", "f", "g", "i", "l", "n", "o", "r", "s", "u"]

#rand = random.choice(rand)

#lets [82] == rand

#lets [rand == f[7] ]

#f[7] == [82]

#rand = random.choice(rand)

#lets [rand == v]

x += 1

#try

import re

j1 = 0

#v == random.choice(rand)

#lets[82] == rand

#j1 == rand

f = 12+22+32+42+52+62+72+j1+92

f

#lets[v == [81]+[82]+[83]+[84]]

#the correct answer

#word4 == random.choice(word4)

j = 0

j == word4

lets[word4 == j]

j == random.choice(word4)

if j == [82]:

if f == [82]:

print(lets[81:85:1]) == j

for q in range(10):

v = random.choice(word4)

# # # print(v)

v == j

else:

q += 1

#result = pd.concat( [f[7], j[1]] )

import re

import string

letters = []

for v in word4:

v, v[1] == word4

#j == [ b[1],b[82],b[3],b[4] ]

#print(b"".join(j))

#the formula

def table (it,rows):

return zip\_longest(\*[it[i:i+rows] for i in range(0, len(it), rows)], fillvalue=" ")

import re

import string

alphabet = string.ascii\_letters + string.punctuation

for i in range (10):

import re

i += 1

".".join(lets)

lets

".".join(lets)

for t in table(lets, 10):

print(\*t)

print()

print()

M = input("the clues (press any key)")

print ()

print ()

print ("down 1D")

print ()

print ("avengement: revenge")

print ("beekeeping: looking after wasps")

print ("blethering: speaking excessevely")

print ("cheapening: reducing in price")

print ("breakevens: meet halfway")

print ("fledgeling: a young bird")

print ("greaseband: a greasy mark")

print ("overseeing: group discussions, taking charge")

print ("previewing: sampling before")

print ("sweetening: improving the taste")

print ()

print ("down 2D")

print ()

print ("checkpoint: point of reference")

print ("cherrylike: like a cherry")

print ("creativity: artistic ability")

print ("skepticism: pessimistic approach")

print ("theologian: an expert in religious studies")

print ("energizing: invigorating")

print ("identities: a frame of reference")

print ("inequality: uneven")

print ("beekeeping: looking after wasps")

print ("pressurise: put pressure on")

print ()

print ("across 1A")

print ()

print ("beekeeper: the keeper of bees")

print ("deflected: bouncing off route")

print ("henpecked: looked after excessively")

print ("requested: a demand for")

print ("sequences: an array")

print ("weekender: an all-week person")

print ()

print ("across 3A")

print ()

print ("kittens: baby cats")

print ("liftoff: taking off for the skies")

print ("mittens: gloves")

print ("dietary: of food")

print ("dieting: less food")

print ("vintage: quality wine")

print ()

print ("down 3D")

print ()

print ("actor: a thespian creature")

print ("after: the opposite of before")

print ("batch: category")

print ("eaten: comsumed")

print ("gates: barred doors")

print ("hatch: a porthole")

print ("match: found equality")

print ("mitre: for a bishop")

print ("metro: train")

print ("octet: eight players")

print ("paths: of a road or roads")

print ("pitch: party piece")

print ("watch: keep looking")

print ("water: a drink")

print ()

print ("down 4D")

print ()

print ("bead: pearl or similar")

print ("beam: wooden ceiling piece")

print ("beer: a foamimg drink")

print ("bees: wasps")

print ("best: par excellence")

print ("beta: second best")

print ("fear: terror")

print ("fees: due cost")

print ("felt: past tense of feel")

print ("fern: green leaf with fronds")

print ("feta: type of cheese")

print ("geek: a brainbox")

print ("gels: colourful inks")

print ("gene: a strand of DNA")

print ("heal: mend naturally")

print ("herd: a group of cows")

print ("heed: hark")

print ("hens: kept chickens")

print ("hero: homecoming soldier")

print ("lead: in the first position")

print ("leap: for joy")

print ("less: least")

print ("meal: a good repast")

print ("melt: dissove slowly")

print ("memo: note to oneself")

print ("tech: high modern")

print ()

print ("across 2D")

print ()

print ("meet: to re\_encounter")

print ("teem: to rain heavily")

print ("heed: hark")

print ("geek: a brainbox")

print ("fees: due cost")

print ("feel: experience")

print ("bees: wasps")

print ("beer: a foaming drink")

print ("need: want desperately")

print ("reed: a wooden flute")

print ("seep: slow leak")

print ("step: up a notch")

print ("tees: golfers'")

print ("week: seven days")

print ("weed: a useless plant")

print ()

print ("across 4A")

print ()

print ("ban: a forbidden item")

print ("bun: a cake")

print ("can: a metal tin of food")

print ("fan: a rotating blade")

print ("fun: enjoyment for a time")

print ("run: put on speed")

print ("sun: sunbathing under")

print ("ten: number of fingers")

print ("tin: a can")

print ("ton: weigh a lot")

print ("van: driverless")