puzzles

What is the output of this code?

print('hello world')

The correct solution:

What is the output of this code?

x = 55 / 11

print(x)

The correct solution:

What is the output of this code?

x = 50 \* 2 + (60 - 20) / 4

print(x)

The correct solution:

What is the output of this code?

text = "# Is this a comment?"

print(text)

The correct solution:

What is the output of this code?

x = 'silent'

print(x[2] + x[1] + x[0] + x[5] + x[3] + x[4])

The correct solution:

What is the output of this code?

squares = [1, 4, 9, 16, 25]

print(squares[0])

The correct solution:

What is the output of this code?

x = 50 // 11

print(x)

The correct solution:

What is the output of this code?

print(3 \* 'un' + 'ium')

The correct solution:

What is the output of this code?

x = 'py' 'thon'

print(x)

The correct solution:

What is the output of this code?

print(sum(range(0, 7)))

The correct solution:

What is the output of this code?

cubes = [1, 8, 27]

cubes.append(4 \*\* 3)

print(cubes)

The correct solution:

What is the output of this code?

word = "galaxy"

print(word[4:50])

The correct solution:

What is the output of this code?

x = 51 % 3

print(x)

The correct solution:

What is the output of this code?

def if\_confusion(x, y):

if x > y:

if x - 5 > 0:

x = y

return "A" if y == y + y else "B"

elif x + y > 0:

while x > y: x -= 1

while y > x: y -= 1

if x == y:

return "E"

else:

if x - 2 > y - 4:

x\_old = x

x = y \* y

y = 2 \* x\_old

if (x - 4) \*\* 2 > (y - 7) \*\* 2:

return "C"

return "D"

return "H"

print(if\_confusion(3, 7))

The correct solution:

What is the output of this code?

x = 'cool'

print(x[-1] + x[-2] + x[-4] + x[-3])

The correct solution:

What is the output of this code?

words = ['cat', 'mouse']

for word in words:

print(len(word))

The correct solution:

What is the output of this code?

def func(x):

return x + 1

f = func

print(f(2) + func(2))

The correct solution:

What is the output of this code?

word = "galaxy"

print(word[:-2] + word[-2:])

The correct solution:

What is the output of this code?

def func(a, \*args):

print(a)

for arg in args:

print(arg)

func("A", "B", "C")

The correct solution:

What is the output of this code?

def ping(i):

if i > 0:

return pong(i - 1)

return "0"

def pong(i):

if i > 0:

return ping(i - 1)

return "1"

print(ping(29))

The correct solution:

What is the output of this code?

word = "bender"

print(word[1:4])

The correct solution:

What is the output of this code?

customers = ['Marie', 'Anne', 'Donald']

customers[2:4] = ['Barack', 'Olivia', 'Sophia']

print(customers)

The correct solution:

What is the output of this code?

def ask(prompt, retries=4, output='Error'):

for \_ in range(retries):

response = input(prompt).lower()

if response in ['y', 'yes']:

return True

if response in ['n', 'no']:

return False

print(output)

print(ask('Want to know the answer?', 5))

The correct solution:

What is the output of this code?

letters = ['a', 'b', 'c', 'd']

print(len(letters[1:-1]))

The correct solution:

What is the output of this code?

a = ['a', 'b']

n = [1, 2]

x = [a, n]

print(x[1])

The correct solution:

What is the output of this code?

letters = ['a', 'b', 'c', 'd', 'e', 'f', 'g']

letters[1:] = []

print(letters)

The correct solution:

What is the output of this code?

# Fibonacci series:

a, b = 0, 1

while b < 5:

print(b)

a, b = b, a + b

The correct solution:

What is the output of this code?

for num in range(2, 8):

if not num % 2:

continue

print(num)

The correct solution:

What is the output of this code?

print(range(5, 10)[-1])

print(range(0, 10, 3)[2])

print(range(-10, -100, -30)[1])

The correct solution:

What is the output of this code?

def matrix\_find(matrix, value):

if not matrix or not matrix[0]:

return False

j = len(matrix) - 1

for row in matrix:

while row[j] > value:

j = j - 1

if j == -1:

return False

if row[j] == value:

return True

return False

matrix = [[3, 4, 4, 6], [6, 8, 11, 12], [6, 8, 11, 15], [9, 11, 12, 17]]

print(matrix\_find(matrix=matrix, value=11))

The correct solution:

What is the output of this code?

def maximum\_profit(prices):

'''Maximum profit of a single buying low and selling high'''

profit = 0

for i, buy\_price in enumerate(prices):

sell\_price = max(prices[i:])

profit = max(profit, sell\_price - buy\_price)

return profit

# Ethereum daily prices in Dec 2017 ($)

eth\_prices = [455, 460, 465, 451, 414, 415, 441]

print(maximum\_profit(prices=eth\_prices)

The correct solution:

What is the output of this code?

def bubble\_sort(lst):

'''Implementation of bubble sort algorithm'''

for border in range(len(lst)-1, 0, -1):

for i in range(border):

if lst[i] > lst[i + 1]:

lst[i], lst[i + 1] = lst[i + 1], lst[i]

return lst

list\_to\_sort = [27, 0, 71, 70, 27, 63, 90]

print(bubble\_sort(lst=list\_to\_sort))

The correct solution:

What is the output of this code?

def concatenation(\*args, sep="/"):

return sep.join(args)

print(concatenation("A", "B", "C", sep=","))

The correct solution:

What is the output of this code?

x = 5 \* 3.8 - 1

print(x)

The correct solution:

What is the output of this code?

def bsearch(l, value):

lo, hi = 0, len(l)-1

while lo <= hi:

mid = (lo + hi) // 2

if l[mid] < value:

lo = mid + 1

elif value < l[mid]:

hi = mid - 1

else:

return mid

return -1

l = [0, 1, 2, 3, 4, 5, 6]

x = 6

print(bsearch(l,x))

The correct solution:

What is the output of this code?

words = ['cat', 'mouse', 'dog']

for word in words[:]:

if len(word) > 3:

words.insert(0, word)

print(words[0])

The correct solution:

What is the output of this code?

def make\_incrementor(n):

return lambda x: x + n

f = make\_incrementor(42)

print(f(0))

print(f(1))

The correct solution:

What is the output of this code?

print("""

A

B

C

""" == "\nA\nB\nC\n")

The correct solution:

What is the output of this code?

print('P"yt\'h"on')

The correct solution:

What is the output of this code?

def fibo(n):

"""Return list containing Fibonacci series up to n."""

result = []

a, b = 0, 1

while a < n:

result.append(a)

a, b = b, a + b

return result

fib100 = fibo(100)

print(fib100[-1] == fib100[-2] + fib100[-3])

The correct solution:

What is the output of this code?

def qsort1(L):

if L:

return qsort1([x for x in L[1:] if x < L[0]]) + L[:1] \

+ qsort1([x for x in L[1:] if x >= L[0]])

return []

def qsort2(L):

if L:

return L[:1] + qsort2([x for x in L[1:] if x < L[0]]) \

+ qsort2([x for x in L[1:] if x >= L[0]])

return []

print(qsort1([0, 33, 22]))

print(qsort2([0, 33, 22]))

The correct solution:

What is the output of this code?

def func(val1=3, val2=4, val3=6):

return val1 + val2 + val3

values = {"val1":9, "val3":-1}

print(func(\*\*values))

The correct solution:

What is the output of this code?

print("Answer")

while True:

pass

print("42")

The correct solution:

What is the output of this code?

def has\_path(graph, v\_start, v\_end, path\_len=0):

'''Graph has path from v\_start to v\_end'''

# Traverse each vertex only once

if path\_len >= len(graph):

return False

# Direct path from v\_start to v\_end?

if graph[v\_start][v\_end]:

return True

# Indirect path via neighbor v\_nbor?

for v\_nbor, edge in enumerate(graph[v\_start]):

if edge: # between v\_start and v\_nbor

if has\_path(graph, v\_nbor, v\_end, path\_len + 1):

return True

return False

# The graph represented as adjancy matrix

G = [[1, 1, 0, 0, 0], [0, 1, 0, 0, 0], [0, 0, 1, 0, 0], [0, 1, 1, 1, 0], [1, 0, 0, 1, 1]]

print(has\_path(graph=G, v\_start=3, v\_end=0))

The correct solution:

What is the output of this code?

pairs = [ (1, 'one'),

(2, 'two'),

(3, 'three'),

(4, 'four')]

# lexicographical sorting (ascending)

pairs.sort(key=lambda pair: pair[1])

print(pairs[0][1])

The correct solution:

What is the output of this code?

# popular instagram accounts

# (millions followers)

inst = { "@instagram":232,

"@selenagomez":133,

"@victoriassecret":59,

"@cristiano":120,

"@beyonce":111,

"@nike":76}

# popular twitter accounts

# (millions followers)

twit = { "@cristiano":69,

"@barackobama":100,

"@ladygaga":77,

"@selenagomez":56,

"@realdonaldtrump":48}

inst\_names = set(filter(lambda key: inst[key]>60, inst.keys()))

twit\_names = set(filter(lambda key: twit[key]>60, twit.keys()))

superstars = inst\_names.intersection(twit\_names)

print(list(superstars)[0])

The correct solution:

What is the output of this code?

words\_list = ["bitcoin", "cryptocurrency", "wallet"]

crawled\_text = ''' Research produced by the University of Cambridge estimates that in 2017, there are 2.9 to 5.8 million unique users using a cryptocurrency wallet, most of them using bitcoin.'''

split\_text = crawled\_text.split()

res1 = True in map(lambda word: word in split\_text, words\_list)

res2 = any(word in words\_list for word in split\_text)

print(res1 == res2)

The correct solution:

What is the output of this code?

def encrypt(text):

encrypted = map(lambda c: chr(ord(c) + 2), text)

return ''.join(encrypted)

def decrypt(text):

decrypted = map(lambda c: chr(ord(c) - 2), text)

return ''.join(decrypted)

s = "xtherussiansarecomingx"

print(decrypt(encrypt(encrypt(s))) == encrypt(s))

The correct solution:

What is the output of this code?

import random

def guess(a, b):

return random.randint(a, b)

def check(x, y):

return y \*\* 2 == x

x = 100

left, right = 0, x

y = guess(left, right)

while not check(x, y):

y = guess(left, right)

print(y)

The correct solution: