Activity Questions

Problem-1

Execute the following code and observe how the type changes dynamically:

```
a = 1
print(type(a))
a = 1.
print(type(a))
a = '1'
print(type(a))
```

Answer

int → float → str

Problem-2

What will be the output when the following code-snippet is executed? Try to predict the answer before executing.

```
x, y, z = 3, 4, 5
x, y, z = x ** 2, y ** 2, z ** 2
print(x + y == z)
```

Answer

`True`

Problem-3

What will happen when the following code is executed? Try to predict the answer before executing.

```
good_variable = 10
a_ = 1
_a = 2
1_a = 10
is = 10
bad variable = 100
```

Answer

Rules for variable names:

1. It can only have alphanumeric characters and underscores.
2. It cannot begin with a number.

Problem-4

Execute the following code and observe the output. What do you think is happening here?

```
x, y = 1, 2
print(x, y)
x, y = y, x
print(x, y)
```

Answer

Swap

Problem-5

Will this code ever print `second`?

```
if True:
    print('first')
else:
    print('second')
```

Answer

Never

Problem-6

What is the difference between the two snippets shown here? Are they functionally equivalent?

```
# Left                                  | # Right
x = True                                | x = True
if x == True:                           | if x:
    print('works')                      |     print('works')
```

Answer

Yes

Problem-7

What is wrong with the following piece of code? If there are errors, modify the code so that it becomes error-free. What is this code trying to do?

```
x = input()
if x % 10 == 0:
    print('multiple of 10')
else:
    print('neither a multiple of 5 nor a multiple of 10')
elif (x % 5 == 0) and (x % 10 != 0):
    print('multiple of 5 and not a multiple of 10')
```

Answer

The code has multiple errors. Correct code is as follows:

```
x = int(input())
if x % 10 == 0:
    print('multiple of 10')
elif (x % 5 == 0) and (x % 10 != 0):
    print('multiple of 5 and not a multiple of 10')
else:
    print('neither a multiple of 5 nor a multiple of 10')
```

Problem-8

What is the following code trying to do? The idea is to take the help of the variable names and the rest of the code to figure out what is happening.

```
year = int(input())
if year % 4 == 0:
    print('leap')
else:
    print('non-leap')
```

Answer

Check for leap year.

Problem-9

The following code accepts three sides of a triangle as input from the user and determines if the triangle is equilateral, isosceles or scalene. Assume that all three sides of the triangle are integers. Is something wrong with the code?

```
x, y, z = int(input()), int(input()), int(input())
if x == y and y == z:
    print('equilateral')
elif x == y and x != z:
    print('isosceles')
elif x == z and x != y:
    print('isosceles')
else:
    print('scalene')
```

Answer

One condition is missing. The complete code is as follows:

```
x, y, z = int(input()), int(input()), int(input())
if x == y and y == z:
    print('equilateral')
elif x == y and x != z:
    print('isosceles')
elif x == z and x != y:
    print('isosceles')
elif y == z and y != x:
    print('isosceles')
else:
    print('scalene')
```

A more optimized version:

```
x, y, z = int(input()), int(input()), int(input())
if x == y == z:
    print('equilateral')
elif (x == y) or (y == z) or (x == z):
    print('isosceles')
else:
    print('scalene')
```

Problem-10

Accept an integer `x` as input from the user. If the number is even, print `even`. If the number is odd, print `odd`.

Answer

```
x = int(input())
if x % 2 == 0:
    print('even')
else:
    print('odd')
```

Problem-11

Accept an integer `x` as input from the user and print the last digit of the number in English. Use the following table for reference.

Last digit of x	Print
0	zero
1	one
...	...
9	nine

Answer

```
x = int(input())
digit = x % 10
if digit == 0:
    print('zero')
elif digit == 1:
    print('one')
elif digit == 2:
    print('two')
elif digit == 3:
    print('three')
elif digit == 4:
    print('four')
elif digit == 5:
    print('five')
elif digit == 6:
    print('six')
elif digit == 7:
    print('seven')
elif digit == 8:
    print('eight')
elif digit == 9:
    print('nine')
```

Problem-12

Accept three integers as input from the user. Print `good triplet` if one of the three numbers is the sum of the other two, and `bad triplet` otherwise.

Answer

```
x, y, z = int(input()), int(input()), int(input())
if (x + y - z == 0) or (y + z - x == 0) or (z + x - y == 0):
    print('good triplet')
else:
    print('bad triplet')
```

Problem-13

Accept three non-negative real numbers as input from the user. If the three numbers form the sides of a triangle, print `True`. If not, print `False`.

Answer

Triangle inequality: Three numbers form the sides of a triangle if and only if the sum of any two numbers is greater than the third number.

```
x, y, z = float(input()), float(input()), float(input())
if (x + y <= z) or (y + z <= x) or (z + x <= y):
    print(False)
else:
    print(True)
```

Problem-14

Accept four distinct integers as input from the user. Print `in ascending order` if the four numbers have been entered in ascending order, and print `not in ascending order` otherwise.

Answer

```
m, n, p, q = int(input()), int(input()), int(input()), int(input())
if m < n < p < q:
    print('in ascending order')
else:
    print('not in ascending order')
```

Problem-15

On what day of the week were you born? If you don't know the answer to this, use the `calendar` library to get the answer.

Answer

```
import calendar
# Entire calendar for the month of August, 1947
calendar.prmonth(1947, 8)
# Gives the exact answer
print(calendar.weekday(1947, 8, 15))
```

For more details, refer https://pypod.github.io/chapter-2/lesson-2.4.html#calendar

Problem-16

The following string is encoded using the Caesar cipher with a shift of 5: `udymts`. Decode the string!

Answer

```
alpha = 'abcdefghijklmnopqrstuvwxyz'
encoded = 'udymts'
decoded = ''

i = 0
decoded += alpha[(alpha.index(encoded[i]) - 5) % 26]
decoded += alpha[(alpha.index(encoded[i + 1]) - 5) % 26]
decoded += alpha[(alpha.index(encoded[i + 2]) - 5) % 26]
decoded += alpha[(alpha.index(encoded[i + 3]) - 5) % 26]
decoded += alpha[(alpha.index(encoded[i + 4]) - 5) % 26]
decoded += alpha[(alpha.index(encoded[i + 5]) - 5) % 26]

print(decoded)  # answer is python
```

Using loops, which you will learn in next week:

```
alpha = 'abcdefghijklmnopqrstuvwxyz'
encoded = 'udymts'
decoded = ''

for i in range(6):
    decoded += alpha[(alpha.index(encoded[i]) - 5) % 26]
print(decoded)  # answer is python
```

Problem-17

Consider the following snippet of code. What do you expect the output to be?

```
empty = ''
if empty:
    print('if')
else:
    print('else')
```

Compare the previous snippet with the following one:

```
zero = 0
if zero:
    print('if')
else:
    print('else')
```

Answer

`else` in both cases

Problem-18

What is the difference in the output between the second and third line?

```
import math
print(2 ** 5)
print(math.pow(2, 5))
```

Answer

1. The `**` operator returns an int while the `math.pow` function returns a float.
2. `**` operator is generally faster than the `math.pow` function.

Problem-19

How many lines do you expect in the output between the strings `before` and `after`?

```
print('before')
print('\n')
print('after')
```

Answer

2

Problem-20

A three digit number is called a `sandwich` number if the difference between its first and last digit is equal to its middle digit. Accept a three digit number as input and print `sandwich` if the number is a sandwich number. Print `plain` if the number is not a sandwich number. For example, `123` and `853` are sandwich numbers.

Answer

```
num = input()
first, middle, last = int(num[0]), int(num[1]), int(num[2])
if abs(first - last) == middle:
    print('sandwich')
else:
    print('plain')
```

`abs` gives the absolute value.

Problem-21

Consider the following message in SMS-speak:

```
message = 'Txt me when u receive this msg.'
```

Using string methods, convert the above message into the following form:

```
Text me when you receive this message.
```

Answer

```
message = 'Txt me when u receive this msg.'
message = message.replace('Txt', 'Text')
message = message.replace('u', 'you')
message = message.replace('msg', 'message')
print(message)
```

Problem-22

Accept a sentence as input and find the number of words in it. Assume that the sentence has no punctuation marks. For example:

Sentence	# words
what a shot	3
this is a chair	4
this sentence is not true	5

Answer

```
sent = input()
num_words = sent.count(' ') + 1
print(num_words)
```

Problem-23

Accept a paragraph as input and find the number of sentences in it. Assume that full stops are the only sentence breaks. All other punctuation marks can be ignored. For example:

Paragraph	# sentences
this is a chair.	1
this is a chair. it is red in color.	2
this is a chair. what a shot. this paragraph has no coherence.	3

Answer

```
para = input()
num_sents = para.count('.')
print(num_sents)
```

Problem-24

A data entry operator at a telephone exchange is using a faulty keyboard — the zero key is missing — to enter phone numbers into an application. As a workaround, he uses the capital letter "O" wherever there is a zero in the phone number. Write a program to accept one of these phone numbers and convert all the "O"s to zeros. In the next week, we will see how to fix all the phone numbers with few more lines of code.

Answer

```
faulty = input()
correct = faulty.replace('O', '0')
print(correct)
```

Problem-25

Consider the decimal expansion of the number 3^{100}. How many times does the digit 3 appear? Is there any digit that never appears in the expansion? Strings may come in handy here.

Answer

```
num = str(3 ** 100)
print(num.count('0'))
print(num.count('1'))
print(num.count('2'))
print(num.count('3'))
print(num.count('4'))
print(num.count('5'))
print(num.count('6'))
print(num.count('7'))
print(num.count('8'))
print(num.count('9'))
```

8 never makes an appearance. This is quite remarkable considering that the number has 48 digits in it.

Problem-26

Consider the following code-snippet:

```
lines = '''one
two
three'''
print('\n' in lines)
print(lines.count('\n'))
```

What do you think is the answer? Execute the code and observe the output. Do you see any `\n` in the string? Explain.

Answer

```
True
2
```

It is a multi-line string.

Problem-27

Use the math library to compute the following function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Here, e is the mathematical constant called the Euler number. $\sigma(x)$ is an example of a "sigmoid function". You will learn more about sigmoid functions in the ML courses in later semesters.

Answer

```
import math
x = float(input())
sigmoid = 1.0 / (1 + math.exp(-x))
```

Problem-28

What is the difference between the string methods `find` and `index`?

Answer

`string.index(x)` and `string.find(x)` return the index of the leftmost occurence of the substring `x` if it is present in the string. If the substring is not present in the string, then `index` throws a `ValueError` while `find` returns the value `-1`.

Problem-29

If `a` and `b` are two strings, what can you say about the value of the expressions in lines 1 and 2?

```
1 (a in b) or (b in a)
2 (a in b) or (a not in b)
```

Answer

First one could be `True` or `False`. The second one is always `True`.

Problem-30

Consider the following code:

```
import random
alpha = 'HT'
toss = random.choice(alpha)
print(toss)
```

What do you think is happening here?

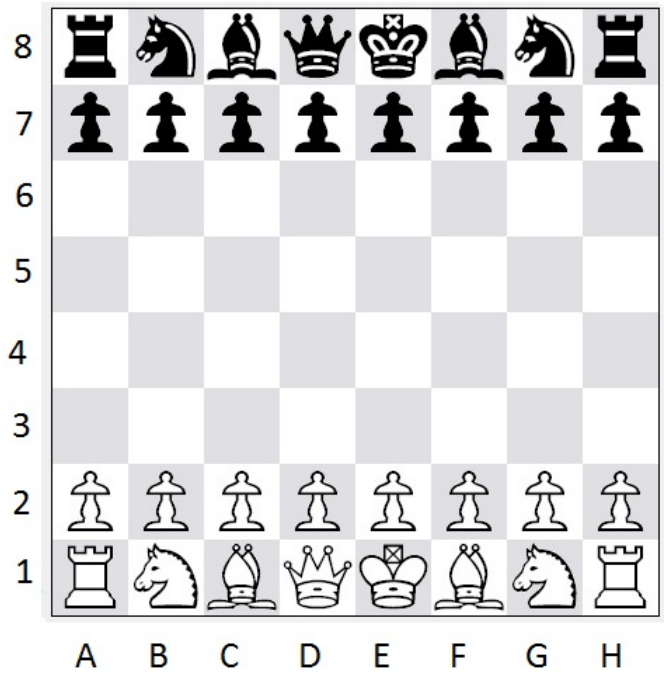
Answer

It simulates the tossing of an unbiased coin.

Problem-31

Bishop's walk

Consider the following image of a chess-board [credits]:



Accept two positions as input: `start` and `end`. Determine if a bishop at `start` can move to `end`. For example:

Input	Output
C3,E5	YES
H8,A1	YES
B2,D3	NO

Use the function `abs` to get the absolute value of an integer. You may need it.

Answer

The bishop travels in a straight line of slope ± 1.

```
start, end = input(), input()
pos = 'ABCDEFGH'
start_horiz, start_vert = pos.index(start[0]), int(start[1])
end_horiz, end_vert = pos.index(end[0]), int(end[1])
if abs(start_horiz - end_horiz) == abs(start_vert - end_vert):
    print('YES')
else:
    print('NO')
```

Problem-32

You have `n` gold coins with you. You wish to divide this among two of your friends under the following conditions:

- Both of them should get a non-zero share.
- Both of them should end up with an even number of coins.

Accept the value of `n` and determine if you can divide it among your two friends. Print `YES` if you can and `NO` if you can't.

Answer

```
n = int(input())
if n % 2 == 0 and n != 0:
    print('YES')
else:
    print('NO')
```

Problem-33

You are given the results of a sequence of ODI matches played by India. A win is represented by `'W'` and a loss is represented by `'L'`. A winning streak is a string of consecutive wins. The outcomes are represented by a single string. If India has played five matches, then the string could look like this — `'WLWWW'`. Answer the following questions:

- Find the number of matches played by India.
- How many matches has it won?
- Does it have a three-match winning streak?
- What is the first match that it has won?

Answer

```
results = input()
# Number of matches
num_matches = len(results)
# Number of matches won
num_wins = results.count('W')
# Three match streak?
if results.find('WWW') != -1:
    print('Yes, it has a three-match winning streak!')
else:
    print('No, it doens\'t have a three-match streak')
# First win
first_win = results.find('W')
if first_win == -1:
    print('India hasn\'t won a single match')
else:
    print('India\'s first win came during match', first_win + 1)
```

Problem-34

In spreadsheets, columns are labeled as follows:

- First 26 columns correspond to the letters A to Z.
- The 27th column is represented by AA, 28th by AB and so on.

You are given a column label as input. Your task is to determine the column number. Example inputs follow:

Input	Output
A	1
Z	26
AC	29
BE	57

Assume that the input will have at most two characters in it.

Answer

```
col = input()
col_parser = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
if len(col) == 1:
    label = col_parser.index(col) + 1
    print(label)
else:
    col_1, col_2 = col
    label_1 = col_parser.index(col_1) + 1
    label_2 = col_parser.index(col_2) + 1
    label = label_1 * 26 + label_2
    print(label)
```

Problem-35

Accept a string as input. Print `valid` if it is a valid name and `invalid` if it is not a valid name. A name is valid if it doesn't contain any numbers or special characters. As an aside, Elon Musk might feel bad if he reads this!

Answer

```
name = input()
if name.isalpha():
    print('valid')
else:
    print('invalid')
```