

Python in 4 sittings

Problem Set 1

15th June, 2021

© pi4py.netlify.app

feel free to contact us;

arabindo@protonmail.com

kaustavbasu97@gmail.com

A few problems are challenging. Take your time. Think again and again. Don't lose hope. Happy Coding!

DIY-1: Ordinarily, '+' used for arithmetic addition. But run the following code block and observe the output.

```
In [ ]: lang = 'Python'
         remarks = 'is a high level programming language'
         new = lang + remarks
         print(new)
```

This kind of operation is known as concatenation. Now run the below code. Figure out why there is an error! Try to resolve the error.

HINT: Concatenation always work with same kind of datatypes. So you may need function like `str()`

```
In [ ]: #math is a library in Python, which contains a number of mathematical func
         import math
         str1 = "The value of pi is"
         p = math.pi # put a comment on dot operator
         new = str1 + p
         print(new)
```

You can hvae concatenation operation with list and tuple . Even you can replicate data with * operation.

```
In [ ]: lang = 'python'
l1 = [1,2,3]
l2 = ['a','b', 'c']
t1 = ('x','y','z')
t2 = ('m','n','o')
l3 = l1 + l2
prnt('concatenation of string: ',l3) # you may assign the result to a new
print('replication of string: ', lang*3) # Or you can directly print the r
```

```
In [ ]: print('concatenation of tuple: ' t1+t2) #Is there is an error? Try to find
```

Problem 1:

Suppose, this semester you have 5 courses.

paper code	credit
PHY101	5
PHY102	5
MTM103	5
CS104	5
ENVS001	4

Part A

Write a program that takes the score input from the user, print the grade and grade points(GP). Suppose GP is calculated in 10 scale, i.e. score*0.10

Score(S)	Grade(G)
S >= 90%	O
80 <= S < 90	A+
60 <= S < 80	A
40 <= S < 60	B
S < 40	F

```
In [ ]: # I'm leaving this row. You may write your code here or anywhere else you
```

Part B

Modify the above code in such a way, that if a student score below 40% it'll assign 0 GP and then use that GP to calculate SGPA using the following formula.

Formula to calculate SGPA:

- number of courses: N (here 5)

- credit for the i^{th} course: C_i
- Grade Point of the i^{th} course: GP_i

$$\text{SGPA for } n\text{th sem}(S_n) : \frac{\sum_{i=1}^N C_i * GP_i}{\sum_{i=1}^N C_i}$$

Now, print the score, grade, grade points and also calculate and print the sgpa to the corresponding semester

```
In [ ]: # I'm leaving this row. You may write your code here or anywhere else you
```

DIY-2

Can you run a code for multiple number of time until user hit a particular input. Here is a little help from our side. consider the following block and try to understand what is going on.

In the below code, whenever, the user hit 0 the loop will end.

Say we want to calculate factorial of different number multiple times.

```
In [ ]: def fact(n):
    factorial = 1
    for i in range(1,n+1): # do you understandd why n+1?
        factorial *= i # equivalent to factorial = factorial * i
    return factorial
flag = '123.'
while True:
    num = int(input("Enter a number to calculate the factorial: "))
    print("Factorial of the number is ", fact(num))
    flag = input("To continue, press an key. To stop press 0: ")
    if(flag == '0'): break
    print("****")
```

Now modify the problem 1-B to run that program multiple number of times.

```
In [ ]: # I'm leaving this row. You may write your code here or anywhere else you
```

DIY-3 Importing Random

Consider the following code

```
In [ ]: import random
for i in range(5):
    print(random.randint(1,20))
    # random.randint(i,j) will return a random integer in between i and j
```

Problem 2

Now, consider to create a guessing game

- store a random number in a variable(say, num).
- Now ask the user to guess a number.
- check if the given input is large or small compared to num .
- Accordingly prompt a message, eg- 'your guess is high'
- Break the loop if the guess is right or ask the user to make another choice, considering the prompted message, if the guess is wrong.
- If you like, you may print the number of iteration

Output may look like:

```
Enter a number in between 1 and 15=5
Your guess is too low
loop no: 1
Enter a number in between 1 and 15=10
Your guess is too high
loop no: 2
Enter a number in between 1 and 15=6
Your guess is too low
loop no: 3
Enter a number in between 1 and 15=7
Congrats! You got it!
```

In []: `# I'm leaving this row. You may write your code here or anywhere else you`

DIY-4 Append method

How you'll create a list taking an input from the user? okay, you need to define a blank list at the first place. Like, `new = []` . Now, you can use append method to add a new data in the list. Consider the following program

```
In [ ]: # say we want to make a list of float numbers
a = []
n = int(input("Enter the number of elements: "))
for i in range(n):
    elements = float(input("Enter the number: "))
    a.append(elements)

# you can check the new list
print(a)

# you can sort the array easily
a.sort()
print('sorted list: ', a)
```

In []:

```
# You can remove element from the list in the following manner:
# Consider a lame example :p
new = ['rat', 'cat', 'bat', 'ant']
new.remove('bat')
print(new)
```

Goto this link, learn more about data structures. You don't have to understand everything. But at least try to understand the first code block in the web page. That is enough for now.

<https://docs.python.org/3/tutorial/datastructures.html>

Problem 3 & 4 are taken from Automate the boring stuff with Python by A. I. Sweigart

Problem 3

Character Picture Grid

Say you have a list of lists where each value in the inner lists is a one-character string, like this:

```
grid = [['.', '.', '.', '.', '.', '.'],
        ['.', '0', '0', '.', '.', '.'],
        ['0', '0', '0', '0', '.', '.'],
        ['0', '0', '0', '0', '0', '.'],
        ['.', '0', '0', '0', '0', '0'],
        ['0', '0', '0', '0', '0', '.'],
        ['0', '0', '0', '0', '.', '.'],
        ['.', '0', '0', '.', '.', '.'],
        ['.', '.', '.', '.', '.', '.']]
```

You can think of `grid[x][y]` as being the character at the x- and y-coordinates of a “picture” drawn with text characters. The (0, 0) origin will be in the upper-left corner, the x-coordinates increase going right, and the y-coordinates increase going down.

Copy the previous grid value, and write code that uses it to print the image.

```
..00.00..
.0000000.
.0000000.
..00000..
...000...
....0....
```

Hint: You will need to use a loop in a loop in order to print `grid[0][0]`, then `grid[1][0]`, then `grid[2][0]`, and so on, up to `grid[8][0]`. This will finish the first row, so then print a newline. Then your program should print `grid[0][1]`, then `grid[1][1]`, then `grid[2][1]`, and so on. The last thing your program will print is `grid[8][5]`. Also, remember to pass the end keyword argument

to print() if you don't want a newline printed automatically after each print() call

```
In [ ]: # I'm leaving this row. You may write your code here or anywhere else you
```

Problem 4

You are creating a fantasy video game. The data structure to model the player's inventory will be a dictionary where the keys are string values describing the item in the inventory and the value is an integer value detailing how many of that item the player has. For example, the dictionary value {'rope': 1, 'torch': 6, 'gold coin': 42, 'dagger': 1, 'arrow': 12} means the player has 1 rope, 6 torches, 42 gold coins, and so on. Write a function named displayInventory() that would take any possible

“inventory” and display it like the following:

```
Inventory:
12 arrow
42 gold coin
1 rope
6 torch
1 dagger
Total number of items: 62
```

HINT: Loop through the key of the dictionary. Remember how to do that? We've discussed in the class. But we forgot to put it in the notebook. Here it is

```
game = {'gems':20, 'life':3, 'user_name':'abc123'}

for keys in game.keys():
    print('keys:', keys)

for elements in game.items():
    print(elements)

for values in game.values():
    print(values)
```

```
In [ ]: # I'm leaving this row. You may write your code here or anywhere else you
```

Problem 5

If you've remember, we worked with Identity matrix in our last session. We had explicitly defined the matrix there. But this time, your job is to generate the matrix. Okay, it's not that hard.

Suggestion: If you are having difficulties, start with 2x2 matrix then generalise it to nxn matrix. Where n is an user input.

Possible HINTS:

- Use nested for loop and check whenever $i=j$. If so append an integer 1 otherwise 0.
- Remember that list supports heterogeneous data types. So may want to append a list within a list.
- Although you can carry out the whole job from a single block of code, We recommend you to use a function to generate the matrix elements: You can define a delta function. That'll take two argument say m and n . While m is equal to n function will return 1

```
In [ ]: # I'm leaving this row. You may write your code here or anywhere else you
```

Problem 6

If you've done the previous problem using function, then this problem is just a slight modification of the previous problem. Otherwise, go back and try to do the problem 5 with the function.

Now import the `math` module in your program. There is method called `sqrt` in the `math` module. So if you say `math.sqrt(2)`, it will return the square root of 2. Design a program to generate the matrix whose elements are:

$$A_{mn} = \sqrt{p(p+1) - n(n+1)} \delta_{m,n+1}$$

where, $\delta_{a,b} = 1$ if $a = b$, otherwise 0. and assume that $p = 1$

$m = 1, 0, -1$ and $n = 1, 0, -1$

You must realise, you have to manipulate the for loop. It is easy in Python with `range` function.

```
In [ ]: # I'm leaving this row. You may write your code here or anywhere else you
```

Problem 7

Create a substitution cipher a.k.a Caesar Cipher.

Part 1 : Encryptor

- Store user input in a variable named `msg`.
- Store user specified key in a variable named `'k'`
- For starting, use `k=1`, as a default input.
- Store the Encrypted message in a variable named `encM`
- For every character `C` in the input string `'msg'`, `C' = C+K` will be stored into `encM`
- clear `msg`

Example: `msg = 'python'`, `k= 1` `encM = 'qzuipo'`

Part 2: Decryptor

- For every character C' in the input string 'encM', $C = C' - K$ will be stored into msg.

Part 3: Extend It

- For any integer value of K , the functions should work.
- Tip, use modular arithmetic for looping purposed.

All the best, folks.