

I. Functions. Arguments.

1. (56) Назвіть основні 4 смислові частини, з яких складається функція.

назва, аргументи, тіло і те, що функція повертає

1. (36) Визначте функцію, що приймає 2 аргументи та повертає результат ділення 1го аргумента на інший.

In [33]:

```
def div(a,b):  
    return a/b  
  
print(div(10,2))
```

5.0

1. (26) Викличте вище визначену функцію, де другий аргумент дорівнює 0.

In [34]:

```
print(div(10,0))
```

```
-----  
ZeroDivisionError                                Traceback (most recent call last)  
Cell In[34], line 1  
----> 1 print(div(10,0))  
  
Cell In[33], line 2, in div(a, b)  
      1 def div(a,b):  
----> 2     return a/b
```

ZeroDivisionError: division by zero

1. (46) Перевизначте функцію так, щоби у випадку ділення на 0 функція повертала **None**, а в решті випадків - результат розрахунків.

In [30]:

```
def div(a,b):  
    if b != 0:  
        return a/b  
    else:  
        return None
```

1. (56) Визначте функцію, що віднімає одне число від іншого (підказка: це буде 2 аргументи функції). При чому, якщо не визначити, скільки саме має бути віднято від числа, за замовчуванням має відніматися 1. Викличте функцію, передаючи тільки один аргумент.

In [31]:

```
def sub(a,b=1):  
    return a-b  
  
print(sub(10,5))  
print(sub(10))
```

5
9

1. (76) Визначте функцію, яка приймає **iterable** (ітерабельний об'єкт, напр. список), що складається з 7ми рядкових значень - назв погодних умов (напр. **"sunny"**) за кожен день у тижні. Повертає функція словник, де ключами є рядки з погодними умовами, а значеннями - кількість разів, скільки дана погода була протягом тижня (напр. **{'sunny': 1, ...}**). Викличте цю функцію та виведіть результат її роботи на екран.

In [8]:

```
def dictCount(s):
    d = {}
    for i in s:
        if i in d:
            d[i] += 1
        else:
            d[i] = 1
    return d

numList = [1,2,2,3,3,3,4,4,4,4,5,5,5,5,5,6,7,7]
print(dictCount(numList))
```

```
{1: 1, 2: 2, 3: 3, 4: 4, 5: 5, 6: 1, 7: 2}
```

1. (156) Напишіть програму, яка б шукала необхідне число у впорядкованому списку цілочислених значень, де число має вводитись з консолі. власне алгоритм пошуку винесений в окрему функцію, яка повертає булеве значення (знайдено число - **True**, не знайдено - **False**). в залежності від результату функції, виведіть на екран повідомлення про успішний або ж неуспішний пошук. Зауваження: використовувати оператор **in** заборонено.

In [27]:

```
def isIn(elem, list):
    while 1 < len(list):
        mid = len(list)//2
        if list[mid] == elem:
            return True
        elif list[mid] > elem:
            list = list[:mid]
        else:
            list = list[mid:]
    return list[0] == elem

i = input()
print(isIn(int(i), numList))
```

```
True
```

II. Lambda functions.

1. (46) В яких випадках варто використовувати лямбда-функції?

Їх варто використовувати для коротких (однострокових) функцій, особливо якщо вони використовуються лише один раз, наприклад, для селекторів сортування, фільтрації та мепінгу.

1. (96) Визначте список з рядками. Після цього визначте анонімну функцію для отримання першого символу у верхньому регістрі для кожного рядка списку. Підказка: використовуйте анонімну функцію в якості аргумента до вбудованої ф-ції, яка створена для застосування функцій над **iterable** об'єктами. Результат виведіть на екран у вигляді кортежу.

In [17]:

```
strList = "Після цього визначте анонімну функцію для отримання першого символу у верхньому регістрі для кожного рядка списку.".split()
firstUpper = lambda s: s[0].upper()
```

```
print(tuple(map(firstUpper, strList)))
```

```
('П', 'Ц', 'В', 'А', 'Ф', 'Д', 'О', 'П', 'С', 'У', 'В', 'Р', 'Д', 'К', 'Р', 'С')
```

1. (106) Визначте список з числовими значеннями (і цілими, і з рухомою комою). а) Відфільтруйте всі числа, які діляться націло на **3** за допомогою вбудованої функції та анонімної функції в якості аргумента). Збережіть результат у змінній, не приводячи явно до будь-якого типу. б) Використовуючи результат на кроці "а" у якості аргументу, та за допомогою лямбда-функції в якості іншого аргументу до вбудованої функції, додайте до кожного елементу число **1** та опісля зведіть у квадрат. Остаточний результат приведіть явно до списку та виведіть на екран.

In [15]:

```
numbers = [1,2.2,3,4.4,5,6.6,7.2,8.1,9,10,12]
div3 = lambda x: x%3 == 0
nums3 = filter(div3, numbers)
nums12 = map(lambda x: (x+1)**2, nums3)
print(list(nums12))
```

```
[16, 100, 169]
```

III. Decorators.

1. (116) Визначте декоратор, який би виводив у консоль повідомлення про початок роботи і назву функції (напр. **myfunction. name _**), а після відпрацювання функції - повідомлення про закінчення роботи і назву функції, що задекорована.

In [12]:

```
def nameDecorator(func):
    def wrapper(*args, **kwargs):
        print("Started: ", func.__name__)
        res = func(*args, **kwargs)
        print("Finished: ", func.__name__)
        return res
    return wrapper
```

1. (46) Задекоруйте двома різними способами функції з завдання 4 та 5 декоратором із завдання 11. (УВАГА: не декоруйте функції двічі!).

In [13]:

```
div = nameDecorator(div)

@nameDecorator
def sub(a,b=1):
    return a-b
```

1. (26) Викличте одну з задекорованих функцій і виведіть її результат на екран.

In [14]:

```
print(div(10,2))
```

```
Started:  div
Finished:  div
5.0
```

IV. Recursion.

1. (46) Назвіть дві частини функції, які є обов'язковими для того, щоб називати її рекурсивною.

Функція викликає сама себе з іншими аргументами, вказана умова за якої рекурсія припиняється

1. **(15б)** Визначте функцію, яка рекурсивно реалізовує пошук числа у послідовності Фібоначчі за її індексом (нумерація починається з нуля). Наприклад, потрібно знайти перше число у послідовності, тоді функція прийматиме **0** у якості аргумента і повертатиме **1**, якщо ж знаходити третє число (індекс **2**) послідовності, то функція повертатиме **2**, якщо ж знаходити десяте число (індекс **9**), то функція повертатиме **55**.

In []:

```
def fib(n):  
    if n < 2:  
        return 1  
    else:  
        return fib(n-1) + fib(n-2)  
  
print(fib(0))  
print(fib(9))
```

1
55

Вітаю! Ви велика(ий) молодець, що впоралась(вся). Похваліть себе та побалуйте чимось приємним. Я Вами пишаюся.