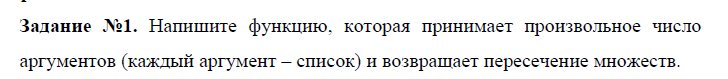
Лабораторная работа №3

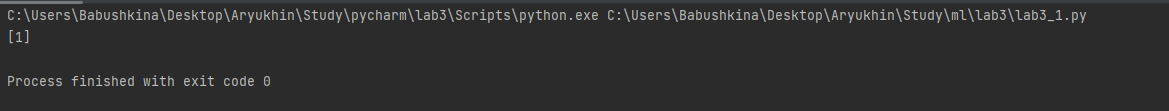
Тема: «Работа с функциями. Хранений функций. Пространство имен. Создание, поиск и использование модулей. Описание основных встроенных модулей».

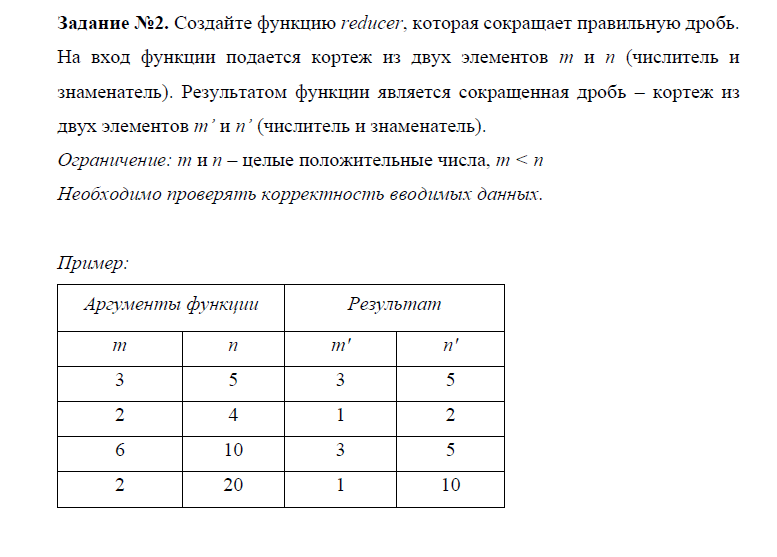
Вариант 1.

Выполнил: Арюхин Александр



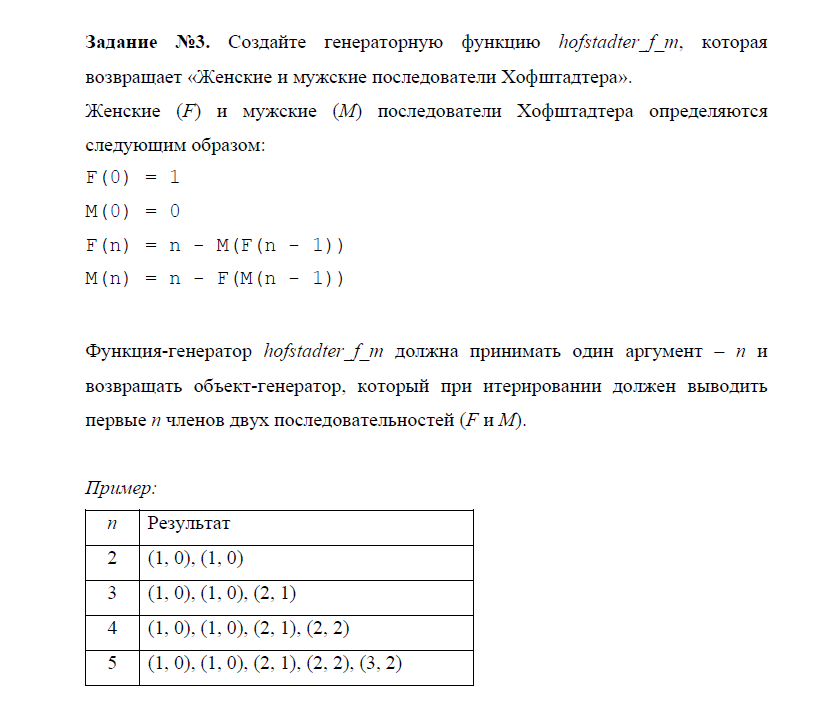
from functools import reduce  
def interesec(\*param: list) ->list:  
 set\_list = list(map(set, param)) # делаем из всех входящих списков множества  
 ans = reduce(lambda x, y: x & y, set\_list) # ищем все пересечения(можно было использовать метод set.intersection)  
 return list(ans)  
  
l1 = [1, 2, 3]  
l2 = [1, 4, 5]  
l3 = [1, 8, 35]  
  
print(interesec(l1, l2, l3))



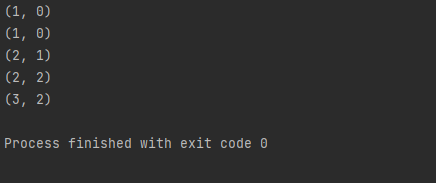


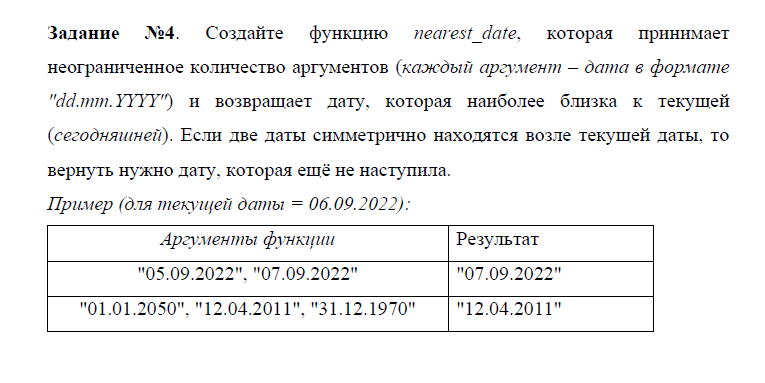
def reducer(x: tuple) -> tuple:  
 if type(x[0]) != int or type(x[1]) != int or x[0] > x[1]:  
 return 'Err'  
 else:  
 if x[1] % x[0] == 0:  
 return (1, int(x[1] / x[0]))  
 for i in range(int(x[0]/2), 0, -1):  
 if x[0] % i == 0:  
 if x[1] % i == 0:  
  
 return tuple(int(element / i) for element in x)  
  
  
  
a = (6,10)  
print(reducer(a))



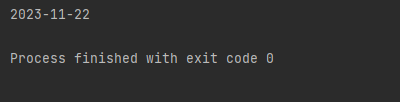


def hofstadter\_f\_m(n: int) -> tuple:  
 F = [1]  
 M = [0]  
 yield (F[0], M[0])  
 for i in range(1, n):  
 M.append(i - F[M[i-1]])  
 F.append(i - M[F[i-1]])  
 yield (F[i], M[i])  
  
  
  
a = hofstadter\_f\_m(5)  
for i in a:  
 print(i)





from datetime import datetime  
  
def nearest\_date(\*dates: str) -> str:  
 wr\_dates = []  
 for i in dates:  
 wr\_dates.append(datetime.strptime(i, "%d.%m.%Y"))  
 today = datetime.now().date()  
 wr\_dates.sort(reverse=True)  
 closest\_date = min(wr\_dates, key=lambda date: abs((today - date.date())))  
 return closest\_date.date()  
  
  
result = nearest\_date("18.12.2020", "20.11.2023", "22.11.2023")  
print(result)



1. В какой момент создается новый объект функции?

Определение функции начинается с выражения ***def***, которое состоит из имени функции, набора параметров в скобках и двоеточия. Параметры в скобках необязательны. Со следующей строки идет блок инструкций, которые выполняет функция. Все инструкции функции имеют отступы от начала строки.

1. Почему следует избегать модификации изменяемых аргументов?

Это может привести к тому, что состояние объекта изменится за пределами функции

1. Что такое интроспекция?

Интроспекция – это способность программы исследовать тип или свойства объекта во время работы программы. Поскольку функции являются объектами, мы можем работать с ними посредством обычных инструментов для объектов.

1. Что такое аннотация функций?

Произвольные определяемые пользователем данные об аргументах и результате функции.

1. Что такое анонимная функция?

Анонимные функции могут содержать лишь одно выражение, но и выполняются они быстрее. Анонимные функции создаются с помощью инструкции lambda. Кроме этого, их не обязательно присваивать переменной, как делали мы инструкцией def func():

1. Правило LEGB.

Область видимости, или scope, определяет контекст переменной, в рамках которого ее можно использовать. Другими словами, область видимости определяет, когда и где вы можете использовать свои переменные, функции, и т. д. Если вы попытаетесь использовать что-либо, что не является в вашей области видимости, вы получите ошибку ***NameError***.

Python содержит три разных типа области видимости:

1) локальная область видимости [local];

2) глобальная область видимости [global];

3) нелокальная область видимости [*еnclosing*] (*была добавлена в Python 3*).

1. Генераторные функции.

Специальный вид функции, которая использует yield для возврата значений во время итерации

1. Генераторные выражения.

Представляют синтаксис для создания генераторов на основе выражения, аналогично списковому включению. Создает объект-генератор, который генерирует значение по требованию, не загружая память.

1. Что такое модуль?

Модуль в языке Python представляет отдельный файл с кодом, который можно повторно использовать в других программах.

Для создания модуля необходимо создать собственно файл с расширением \*.py, который будет представлять модуль. Название файла будет представлять название модуля. Затем в этом файле надо определить одну или несколько функций.

1. Как работает поиск модуля при импортировании?

Поиск модуля происходит в соответствии с LEGB.

Для использования модуля его надо импортировать с помощью оператора import, после которого указывается имя модуля: import account.

Чтобы обращаться к функциональности модуля, нам нужно получить его пространство имен. По умолчанию оно будет совпадать с именем модуля, то есть в нашем случае также будет называться account.

Получив пространство имен модуля, мы сможем обратиться к его функциям по схеме пространство\_имен.функция:

1. Для чего нужны файлы \_\_init\_\_.py?

Для того, чтобы пайтон рассматривал каталог как пакет, а не как директорию