Лабораторная работа №9 “Работа с функциями. Создание модулей”

### **Задача 1. Условие.**

Список A содержит N чисел. Перепишите из списка A в список B только те элементы, значения которых не равны заданному значению znach. Назначение функции: переписывание из одного списка в другой только тех элементов, которые не совпадают с заданным значением.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Входные данные | znach | Заданное значение для удаления | целое число |  |
| Выходные данные | B | Список без элементов znach | список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [1, 2, 3, 4, 5, 2, 3] |
| Входные данные | znach | 2 |
| Выходные данные | B | [1, 3, 4, 5, 3] |

#### **Листинг программы:**

Программа создает список B, исключая элементы из списка A, равные заданному значению znach.

#### **Код программы:**

def filter\_list(A, znach):  
B = [x for x in A if x != znach]  
return B  
A = [int(x) for x in input("Введите элементы списка A через пробел: ").split()]  
znach = int(input("Введите значение для удаления: "))  
B = filter\_list(A, znach)  
print(f"Список B без элементов, равных {znach}: {B}")

### **Задача 12. Условие.**

Список A содержит N чисел. Разработайте программу, с помощью которой можно определить количество наибольших элементов в нем. Назначение 1 функции: нахождение максимального элемента. Назначение 2 функции: подсчет количества максимальных элементов.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Выходные данные | max\_element | Максимальный элемент | целое число |  |
| Выходные данные | count\_max | Количество максимальных элементов | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [4, 7, 2, 7, 5, 7] |
| Выходные данные | max\_element | 7 |
| Выходные данные | count\_max | 3 |

Листинг программы:  
A = [int(x) for x in input("Введите элементы списка через пробел: ").split()]  
max\_element = max(A)  
count\_max = A.count(max\_element)  
print(f"Максимальный элемент: {max\_element}")  
print(f"Количество максимальных элементов: {count\_max}")

### **Задача 13. Условие.**

Список A содержит N чисел. Заданное число вставьте на место с порядковым номером K. Назначение функции: вставка в список заданного значения на место с заданным порядковым номером.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Входные данные | znach | Число для вставки | целое число |  |
| Входные данные | K | Порядковый номер для вставки | целое число |  |
| Выходные данные | A | Обновленный список | список |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [10, 20, 30, 40, 50] |
| Входные данные | znach | 25 |
| Входные данные | K | 2 |
| Выходные данные | A | [10, 20, 25, 30, 40, 50] |

Листинг программы:  
A = [int(x) for x in input("Введите элементы списка через пробел: ").split()]  
znach = int(input("Введите число для вставки: "))  
K = int(input("Введите порядковый номер: "))  
A.insert(K, znach)  
print(f"Обновленный список: {A}")

### **Задача 14. Условие.**

Список A содержит N чисел. Найдите в нем элемент с наибольшим значением. Назначение функции: поиск в списке элемента с наибольшим значением.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Выходные данные | max\_element | Максимальный элемент | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [10, 20, 30, 5, 50] |
| Выходные данные | max\_element | 50 |

Листинг программы:  
A = [int(x) for x in input("Введите элементы списка через пробел: ").split()]  
max\_element = max(A)  
print(f"Максимальный элемент: {max\_element}")

### **Задача 15. Условие.**

Разработайте программу для определения числа сочетаний из n по m. Число определяется по формуле C(n,m) = n! / (m! \* (n - m)!). Вычисление факториала оформите в виде функции.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | n | Общее количество элементов | целое число | n ≥ m |
| Входные данные | m | Количество выбранных элементов | целое число | m ≥ 0 |
| Выходные данные | C | Число сочетаний | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | n | 5 |
| Входные данные | m | 3 |
| Выходные данные | C | 10 |

Листинг программы:  
import math  
n = int(input("Введите n: "))  
m = int(input("Введите m: "))  
C = math.factorial(n) // (math.factorial(m) \* math.factorial(n - m))  
print(f"Число сочетаний C({n},{m}) = {C}")

### **Задача 16. Условие.**

Из заданной вложенной последовательности A[i,j] удалите строку с порядковым номером N\_st. Назначение функции: удаление строки по заданному порядковому номеру.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Входные данные | N\_st | Порядковый номер строки для удаления | целое число |  |
| Выходные данные | A | Обновленная последовательность | двумерный список |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, 2], [3, 4], [5, 6]] |
| Входные данные | N\_st | 1 |
| Выходные данные | A | [[1, 2], [5, 6]] |

Листинг программы:  
A = [[int(x) for x in input().split()] for \_ in range(3)]  
N\_st = int(input("Введите порядковый номер строки для удаления: "))  
del A[N\_st]  
print("Обновленная последовательность:")  
for row in A:  
print(row)

### **Задача 17. Условие.**

Разработайте программу, которая заполняет список A[10] случайными целыми числами от 1 до 99. Определите, сколько в нем имеется простых чисел. (Число, которое делится только на единицу или само на себя, называется простым).

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список случайных чисел | список | целые числа от 1 до 99 |
| Выходные данные | count\_primes | Количество простых чисел | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [5, 7, 8, 9, 11, 14, 15, 19, 21, 23] |
| Выходные данные | count\_primes | 5 |

Листинг программы:  
import random

def is\_prime(n):  
if n < 2:  
return False  
for i in range(2, int(n\*\*0.5) + 1):  
if n % i == 0:  
return False  
return True

A = [random.randint(1, 99) for \_ in range(10)]  
count\_primes = sum(1 for num in A if is\_prime(num))

print(f"Список: {A}")  
print(f"Количество простых чисел: {count\_primes}")

### **Задача 18. Условие.**

Разработайте программу, в которой осуществляется подсчет количества встречающегося в заданной целочисленной вложенной последовательности A[5,5] максимального по величине числа. Оформите в виде функций: 1. Нахождение максимального элемента. 2. Подсчет числа вхождений максимального элемента.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Выходные данные | max\_element | Максимальный элемент | целое число |  |
| Выходные данные | count\_max | Количество вхождений максимального элемента | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, 5, 3], [4, 5, 2], [5, 1, 3]] |
| Выходные данные | max\_element | 5 |
| Выходные данные | count\_max | 3 |

Листинг программы:  
A = [[int(x) for x in input().split()] for \_ in range(3)]

def find\_max\_element(matrix):  
return max(max(row) for row in matrix)

def count\_max\_element(matrix, max\_element):  
return sum(row.count(max\_element) for row in matrix)

max\_element = find\_max\_element(A)  
count\_max = count\_max\_element(A, max\_element)

print(f"Максимальный элемент: {max\_element}")  
print(f"Количество вхождений максимального элемента: {count\_max}")

### **Задача 19. Условие.**

Список A содержит N чисел. Значения всех элементов списка, отличающихся от заданного значения znach, замените другим заданным значением znach1.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Входные данные | znach | Заданное значение | целое число |  |
| Входные данные | znach1 | Новое значение для замены | целое число |  |
| Выходные данные | A | Обновленный список | список |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [5, 6, 7, 6, 5] |
| Входные данные | znach | 6 |
| Входные данные | znach1 | 9 |
| Выходные данные | A | [5, 9, 7, 9, 5] |

Листинг программы:  
A = [int(x) for x in input("Введите элементы списка через пробел: ").split()]  
znach = int(input("Введите значение для поиска: "))  
znach1 = int(input("Введите значение для замены: "))

A = [znach1 if x != znach else x for x in A]  
print(f"Обновленный список: {A}")

### **Задача 20. Условие.**

Заданы вложенные последовательности A и B согласованного размера. Найдите произведение обеих последовательностей.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A, B | Вложенные последовательности | двумерные списки | целые числа |
| Выходные данные | C | Произведение последовательностей | двумерный список |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, 2], [3, 4]] |
| Входные данные | B | [[5, 6], [7, 8]] |
| Выходные данные | C | [[5, 12], [21, 32]] |

Листинг программы:  
A = [[int(x) for x in input().split()] for \_ in range(2)]  
B = [[int(x) for x in input().split()] for \_ in range(2)]

C = [[A[i][j] \* B[i][j] for j in range(2)] for i in range(2)]

print("Произведение последовательностей:")  
for row in C:  
print(row)

### **Задача 21. Условие.**

Найдите сумму значений элементов заданной вложенной последовательности A[i,j], меньших заданного значения Znach.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Входные данные | Znach | Пороговое значение | целое число |  |
| Выходные данные | sum\_elements | Сумма элементов, меньших порогового значения | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[5, 7], [8, 10]] |
| Входные данные | Znach | 8 |
| Выходные данные | sum\_elements | 20 |

Листинг программы:  
A = [[int(x) for x in input().split()] for \_ in range(2)]  
Znach = int(input("Введите пороговое значение: "))

sum\_elements = sum(A[i][j] for i in range(2) for j in range(2) if A[i][j] < Znach)  
print(f"Сумма элементов, меньших {Znach}: {sum\_elements}")

### **Задача 22. Условие.**

Разработайте программу решения группы квадратных уравнений aix2+bix+ci=0a\_i x^2 + b\_i x + c\_i = 0ai​x2+bi​x+ci​=0, где aaa, bbb, ccc – списки вещественных чисел, состоящие из k элементов. Решение одного уравнения оформите в виде функции.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | a, b, c | Списки коэффициентов уравнения | списки вещественных чисел | целые или вещественные числа |
| Выходные данные | solutions | Решения уравнений | список строк |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | a | [1, 1] |
| Входные данные | b | [-3, -2] |
| Входные данные | c | [2, 1] |
| Выходные данные | solutions | ['x1 = 2.0, x2 = 1.0', 'x1 = 1.0'] |

Листинг программы:  
import math

def solve\_quadratic(a, b, c):  
discriminant = b\*\*2 - 4 \* a \* c  
if discriminant > 0:  
x1 = (-b + math.sqrt(discriminant)) / (2 \* a)  
x2 = (-b - math.sqrt(discriminant)) / (2 \* a)  
return f"x1 = {x1}, x2 = {x2}"  
elif discriminant == 0:  
x = -b / (2 \* a)  
return f"x1 = {x}"  
else:  
return "Нет вещественных решений"

a = [float(x) for x in input("Введите коэффициенты a: ").split()]  
b = [float(x) for x in input("Введите коэффициенты b: ").split()]  
c = [float(x) for x in input("Введите коэффициенты c: ").split()]

solutions = [solve\_quadratic(a[i], b[i], c[i]) for i in range(len(a))]  
print("Решения уравнений:")  
for solution in solutions:  
print(solution)

### **Задача 23. Условие.**

Дано n целых чисел. Найдите среди них число, у которого сумма цифр имеет максимальное значение.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | numbers | Список целых чисел | список | целые числа |
| Промежуточные данные | sum\_digits | Сумма цифр числа | целое число |  |
| Выходные данные | max\_number | Число с максимальной суммой цифр | целое число |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | numbers | [123, 456, 789] |
| Выходные данные | max\_number | 789 |

Листинг программы:  
def sum\_of\_digits(n):  
return sum(int(digit) for digit in str(n))

numbers = [int(x) for x in input("Введите числа через пробел: ").split()]  
max\_number = max(numbers, key=sum\_of\_digits)

print(f"Число с максимальной суммой цифр: {max\_number}")

### **Задача 24. Условие.**

Дана вложенная последовательность n×mn \times mn×m. Найдите индексы всех максимальных элементов вложенной последовательности.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Выходные данные | indices | Индексы максимальных элементов | список кортежей | индексы |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, 2, 3], [4, 5, 3], [5, 1, 3]] |
| Выходные данные | indices | [(2, 0), (2, 1)] |

Листинг программы:  
A = [[int(x) for x in input().split()] for \_ in range(3)]  
max\_element = max(max(row) for row in A)

indices = [(i, j) for i in range(len(A)) for j in range(len(A[i])) if A[i][j] == max\_element]

print(f"Максимальный элемент: {max\_element}")  
print(f"Индексы максимальных элементов: {indices}")

### **Задача 25. Условие.**

Сформируйте список, каждый элемент которого равен сумме отрицательных элементов соответствующей строки заданной вложенной последовательности.

Таблица данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Выходные данные | sum\_negatives | Список сумм отрицательных элементов | список целых чисел |  |

Тесты.

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, -2, -3], [4, -1, -5], [-2, -3, 0]] |
| Выходные данные | sum\_negatives | [-5, -6, -5] |

Листинг программы:  
A = [[int(x) for x in input().split()] for \_ in range(3)]  
sum\_negatives = [sum(x for x in row if x < 0) for row in A]

print(f"Суммы отрицательных элементов по строкам: {sum\_negatives}")

### **Задача 2. Условие.**

Список A содержит N элементов, значения которых не определены. Организуйте запрос: «Сколько элементов списка следует заполнить?» Заполните список заданным числовым значением znach.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список из N элементов | список | не заполнен |
| Входные данные | znach | Числовое значение для заполнения списка | целое число |  |
| Выходные данные | A | Заполненный список | список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | N | 5 |
| Входные данные | znach | 7 |
| Выходные данные | A | [7, 7, 7, 7, 7] |

#### **Листинг программы:**

Запрашивается количество элементов для заполнения, и все элементы заполняются значением znach.

#### **Код программы:**

N = int(input("Сколько элементов списка следует заполнить? "))  
znach = int(input("Введите значение для заполнения: "))  
A = [znach] \* N  
print(f"Заполненный список A: {A}")

### **Задача 3. Условие.**

Список A содержит N чисел. Найдите количество элементов списка, значения которых превышают заданное значение znach.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Входные данные | znach | Заданное значение для сравнения | целое число |  |
| Выходные данные | count | Количество элементов больше znach | целое число |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [1, 3, 5, 7, 9] |
| Входные данные | znach | 4 |
| Выходные данные | count | 3 |

#### **Листинг программы:**

Программа подсчитывает количество элементов в списке A, которые больше значения znach.

#### **Код программы:**

A = [int(x) for x in input("Введите элементы списка A через пробел: ").split()]  
znach = int(input("Введите значение для сравнения: "))  
count = len([x for x in A if x > znach])  
print(f"Количество элементов больше {znach}: {count}")

### **Задача 4. Условие.**

Список A содержит N чисел. Найдите сумму значений элементов списка, меньших заданного значения znach.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Входные данные | znach | Заданное значение для сравнения | целое число |  |
| Выходные данные | sum\_values | Сумма элементов меньше znach | целое число |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [1, 3, 5, 7, 9] |
| Входные данные | znach | 6 |
| Выходные данные | sum\_values | 9 |

#### **Листинг программы:**

Программа вычисляет сумму элементов, которые меньше значения znach.

#### **Код программы:**

A = [int(x) for x in input("Введите элементы списка A через пробел: ").split()]  
znach = int(input("Введите значение для сравнения: "))  
sum\_values = sum([x for x in A if x < znach])  
print(f"Сумма элементов меньше {znach}: {sum\_values}")

### **Задача 5. Условие.**

Список A содержит N чисел. Найдите произведение значений элементов списка.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Выходные данные | product | Произведение элементов | целое число |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [2, 3, 5] |
| Выходные данные | product | 30 |

#### **Листинг программы:**

Программа находит произведение всех элементов списка A.

#### **Код программы:**

import math  
A = [int(x) for x in input("Введите элементы списка A через пробел: ").split()]  
product = math.prod(A)  
print(f"Произведение элементов списка: {product}")

### **Задача 6. Условие.**

Подсчитайте количество элементов заданной вложенной последовательности A[i,j], значения которых превышают заданное значение znach.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Входные данные | znach | Заданное значение для сравнения | целое число |  |
| Выходные данные | count | Количество элементов больше znach | целое число |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, 3, 5], [7, 9, 11], [13, 15, 17]] |
| Входные данные | znach | 8 |
| Выходные данные | count | 5 |

#### **Листинг программы:**

Программа подсчитывает количество элементов во вложенной последовательности, которые больше значения znach.

#### **Код программы:**

A = [[int(x) for x in input().split()] for \_ in range(3)]  
znach = int(input("Введите значение для сравнения: "))  
count = len([A[i][j] for i in range(3) for j in range(3) if A[i][j] > znach])  
print(f"Количество элементов больше {znach}: {count}")

### **Задача 7. Условие.**

Список A содержит N чисел. Удалите из списка значение с порядковым номером K. Назначение функции: удаление из списка значения с заданным порядковым номером.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список чисел | список | целые числа |
| Входные данные | K | Порядковый номер элемента для удаления | целое число |  |
| Выходные данные | A | Список после удаления | список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [10, 20, 30, 40, 50] |
| Входные данные | K | 2 |
| Выходные данные | A | [10, 30, 40, 50] |

#### **Листинг программы:**

Программа удаляет элемент с порядковым номером K из списка A.

#### **Код программы:**

A = [int(x) for x in input("Введите элементы списка A через пробел: ").split()]  
K = int(input("Введите порядковый номер элемента для удаления: "))  
if 0 <= K < len(A):  
del A[K]  
print(f"Список после удаления элемента {K}: {A}")

### **Задача 8. Условие.**

Задана исходная вложенная последовательность A[i,j]. Умножьте ее на заданное число chislo. Назначение функции: умножение заданной вложенной последовательности на число.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Входные данные | chislo | Число для умножения | целое число |  |
| Выходные данные | A | Умноженная последовательность | двумерный список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, 2], [3, 4]] |
| Входные данные | chislo | 2 |
| Выходные данные | A | [[2, 4], [6, 8]] |

#### **Листинг программы:**

Программа умножает все элементы вложенной последовательности на заданное число chislo.

#### **Код программы:**

A = [[int(x) for x in input().split()] for \_ in range(2)]  
chislo = int(input("Введите число для умножения: "))  
A = [[x \* chislo for x in row] for row in A]  
print("Результат умножения последовательности на число:")  
for row in A:  
print(row)

### **Задача 9. Условие.**

В заданной вложенной последовательности A[i,j] значения всех элементов, отличающихся от заданного значения znach1, замените другим заданным значением znach2. Назначение функции: замена значений всех элементов, отличающихся от заданного значения, другим заданным значением.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Вложенная последовательность | двумерный список | целые числа |
| Входные данные | znach1 | Заданное значение для сравнения | целое число |  |
| Входные данные | znach2 | Значение для замены | целое число |  |
| Выходные данные | A | Обновленная последовательность | двумерный список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [[1, 2], [3, 4]] |
| Входные данные | znach1 | 2 |
| Входные данные | znach2 | 5 |
| Выходные данные | A | [[1, 5], [3, 5]] |

#### **Листинг программы:**

Программа заменяет все элементы, отличающиеся от значения znach1, на значение znach2.

#### **Код программы:**

A = [[int(x) for x in input().split()] for \_ in range(2)]  
znach1 = int(input("Введите значение для сравнения: "))  
znach2 = int(input("Введите значение для замены: "))  
A = [[znach2 if x != znach1 else x for x in row] for row in A]  
print("Обновленная последовательность:")  
for row in A:  
print(row)

### **Задача 10. Условие.**

Список A содержит N упорядоченных по возрастанию чисел. Вставьте в список некоторое значение znach так, чтобы упорядоченность списка не нарушилась. Назначение функции: вставка заданного значения в упорядоченный список.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Упорядоченный список | список | целые числа |
| Входные данные | znach | Значение для вставки | целое число |  |
| Выходные данные | A | Обновленный список | список |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [1, 3, 5, 7, 9] |
| Входные данные | znach | 6 |
| Выходные данные | A | [1, 3, 5, 6, 7, 9] |

#### **Листинг программы:**

Программа вставляет значение znach в упорядоченный список A, сохраняя порядок.

#### **Код программы:**

A = [int(x) for x in input("Введите упорядоченный список A через пробел: ").split()]  
znach = int(input("Введите значение для вставки: "))  
A.append(znach)  
A.sort()  
print(f"Обновленный список: {A}")

### **Задача 11. Условие.**

Списки A и B содержат N и M целых чисел, соответственно. Разработайте программу, которая выводит сообщение о том, в каком списке произведение элементов имеет большее значение.

#### **Таблица данных.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Класс** | **Имя** | **Смысл** | **Тип, структура** | **Ограничения на значения** |
| Входные данные | A | Список целых чисел | список | целые числа |
| Входные данные | B | Список целых чисел | список | целые числа |
| Выходные данные | result | Сообщение о большем произведении | строка |  |

#### **Тесты.**

|  |  |  |
| --- | --- | --- |
| **Класс** | **Имя** | **Значение** |
| Входные данные | A | [1, 3, 5] |
| Входные данные | B | [2, 4, 6] |
| Выходные данные | result | "Произведение списка B больше" |

#### **Листинг программы:**

Программа находит произведение элементов двух списков и выводит сообщение о том, у какого списка произведение больше.

#### **Код программы:**

import math  
A = [int(x) for x in input("Введите элементы списка A через пробел: ").split()]  
B = [int(x) for x in input("Введите элементы списка B через пробел: ").split()]  
product\_A = math.prod(A)  
product\_B = math.prod(B)  
if product\_A > product\_B:  
print("Произведение списка A больше")  
else:  
print("Произведение списка B больше")