|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего образования  **«МИРЭА** - **Российский технологический университет»**  **РТУ МИРЭА** |

Институт информационных технологий (ИТ)

Кафедра прикладной математики (ПМ)

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №1** | | |
| **по дисциплине** | | |
| «Технологии и инструментарий анализа больших данных» | | |
| Выполнил: студент группы ИКБО-33-22 | Шило Ю. С. |  |
| Проверил: преподаватель | Юрченков И.А. |  |
|  |  |  |
|  |  |  |

Практическая работа выполнена «     »                           2025 г.

(подпись студента)

Зачтено «     »                           2025 г.

(подпись преподавателя)

Москва 2025

**Задание 1**

Текст задания:

Написать программу, которая вычисляет площадь фигуры, параметры которой подаются на вход. Фигуры, которые подаются на вход: треугольник, прямоугольник, круг. Результатом работы является словарь, где ключ – это название фигуры, а значение – это площадь.

Выполнение задания:

|  |
| --- |
| import math  def circle(radius):  return radius \* radius \* math.pi  def triangle(side, heigh):  return side \* heigh  def square(side, s\_side=None):  if s\_side is None:  s\_side = side  return side \* s\_side  if \_\_name\_\_ == '\_\_main\_\_':  dict = {'circle': circle(10), 'triangle': triangle(10, 2), 'square': square(10, 12)}  print(dict) |

**Задание 2**

Текст задания:

Написать программу, которая на вход получает два числа и операцию, которую к ним нужно применить. Должны быть реализованы следующие операции: +, -, /, //, abs – модуль, pow или \*\* – возведение в степень. Результатом работы программы является одно число.

Выполнение задания:

|  |
| --- |
| def calculate(string):  string = string.split(' ')  operation = string[2]  a = float(string[0])  b = float(string[1])  operations = {  '+': lambda a, b: a + b,  '-': lambda a, b: a - b,  '\*': lambda a, b: a \* b,  '/': lambda a, b: a / b,  '\*\*': lambda a, b: a \*\* b,  '//': lambda a, b: a // b,  }  if operation in operations:  return operations[operation](a, b)  return None  if \_\_name\_\_ == '\_\_main\_\_':  print(calculate('12 12 +')) |

**Задание 3**

Текст задания:

Написать программу, вычисляющую площадь треугольника по переданным длинам трёх его сторон по формуле Герона.

Выполнение задания:

|  |
| --- |
| import math  def formula(a, b, c):  p = a + b + c / 2  return math.sqrt(p \* (p - a) \* (p - b) \* (p - c))  if \_\_name\_\_ == '\_\_main\_\_':  print(formula(2, 3, 4)) |

**Задание 4**

Текст задания:

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и после этого выводит сумму квадратов всех считанных чисел.

Выполнение задания:

|  |
| --- |
| if \_\_name\_\_ == '\_\_main\_\_':  arr = list()  while (num := input()) != '0':  arr.append(int(num) \*\* 2)  print(sum(arr)) |

**Задание 5**

Текст задания:

Напишите программу, которая выводит последовательность чисел, длинною N, где каждое число повторяется столько раз, чему оно равно. На вход программе передаётся неотрицательное целое число N. Например, если N = 7, то программа должна вывести 1 2 2 3 3 3 4. Вывод элементов списка через пробел – print(\*list).

Выполнение задания:

|  |
| --- |
| n = int(input())  res = []  current\_number = 1  while len(res) < n:  for i in range(current\_number):  if len(res) < n:  res.append(current\_number)  else:  break  current\_number += 1  print(\*res) |

**Задание 6**

Текст задания:

Даны два списка:

А = [1, 2, 3, 4, 2, 1, 3, 4, 5, 6, 5, 4, 3, 2]

В = [‘a’, ’b’, ’c’, ’c’, ’c’, ’b’, ’a’, ’c’, ’a’, ’a’, ’b’, ’c’, ’b’, ’a’]

Создать словарь, в котором ключи – это содержимое списка В, а значения для ключей словаря – это сумма всех элементов списка А в соответствии с буквой, содержащийся на той же позиции в списке В.

Пример результата программы: {‘a’ : 10, ‘b’ : 15, ‘c’ : 6}.

Выполнение задания:

|  |
| --- |
| A = [2, 3, 4, 1]  B = ['a', 'b', 'a', 'a']  res = {}  for i in range(len(A)):  num = A[i]  ltr = B[i]  if ltr in res:  res[ltr] += num  else:  res[ltr] = num  print(res) |

**Задание 7**

Текст задания:

Скачать и загрузить данные о стоимости домов в калифорнии, используя библиотеку sklearn.

Выполнение задания:

|  |
| --- |
| from sklearn.datasets import fetch\_california\_housing  data = fetch\_california\_housing(as\_frame=True) |

**Задание 8**

Текст задания:

Использовать метод info().

Выполнение задания:

|  |
| --- |
| from sklearn.datasets import fetch\_california\_housing  data = fetch\_california\_housing(as\_frame=True).frame  data.info() |

**Задание 9**

Текст задания:

Узнать, есть ли пропущенные значения, используя isna().sum().

Выполнение задания:

|  |
| --- |
| from sklearn.datasets import fetch\_california\_housing  data = fetch\_california\_housing(as\_frame=True).frame  print(data.isnull().sum()) |

**Задание 10**

Текст задания:

Вывести записи, где средний возраст домов в районе более 50 лет и население более 2500 человек, используя метод loc().

Выполнение задания:

|  |
| --- |
| from sklearn.datasets import fetch\_california\_housing  data = fetch\_california\_housing(as\_frame=True).frame  print(data.loc[(data['HouseAge'] > 50) & (data['Population'] > 2500)]) |

**Задание 11**

Текст задания:

Узнать максимальное и минимальное значения медианной стоимости дома.

Выполнение задания:

|  |
| --- |
| from sklearn.datasets import fetch\_california\_housing  data = fetch\_california\_housing(as\_frame=True).frame  print(data['MedHouseVal'].max())  print(data['MedHouseVal'].min()) |

**Задание 12**

Текст задания:

Используя метод apply(), вывести на экран название признака и его среднее значение.

Выполнение задания:

|  |
| --- |
| from sklearn.datasets import fetch\_california\_housing  data = fetch\_california\_housing(as\_frame=True).frame  mean\_values = data.apply(lambda col: col.mean())  for column, mean\_value in mean\_values.items():  print(f"{column}: {mean\_value}") |

**Задание 1\***

Текст задания:

Дан текст на английском языке. Необходимо закодировать его с помощью азбуки Морзе, где каждой букве соответствует последовательность точек и тире. Например, буква «g» превратится в строку «--.». В переменной morze для удобства хранится словарь соответствия латинских букв коду Морзе.

Выполнение задания:

|  |
| --- |
| morze = {'a': '.-', 'b': '-…', 'c': '-.-.', 'd': '-..', 'e': '.', 'f': '..-.', 'g': '--.', 'h': '….', 'i': '..',  'j': '.---', 'k': '-.-', 'l': '.-..', 'm': '--', 'n': '-.', 'o': '---', 'p': '.--.', 'q': '--.-', 'r': '.-.',  's': '…', 't': '-', 'u': '..-', 'v': '…-', 'w': '.--', 'x': '-..-', 'y': '-.--', 'z': '--..'}  def coding(string):  string = string.lower()  word = ''  for char in string:  if char in morze:  word += morze[char] + ' '  else:  word += '\n'  return word  if \_\_name\_\_ == '\_\_main\_\_':  print(coding('Ignition sequence start')) |

**Задание 2\***

Текст задания:

В некотором городе открывается новая служба по доставке электронных писем. Необходимо наладить систему регистрации новых пользователей. Регистрация должна работать следующим образом: если новый пользователь хочет зарегистрироваться на сайте, то он должен послать системе запрос name со своим именем. Система должна определить, существует ли уже такое имя в базе данных. Если такого имени не существует, то оно заносится в базу данных системы и пользователю возвращается ответ "ОК", подтверждающий успешную регистрацию. А если пользователь с таким именем уже существует, то система должна сформировать новое имя и выдать его пользователю в качестве подсказки, при этом сама подсказка также добавляется в базу данных. Новое имя формируется следующим образом: к name последовательно приписываются числа, начиная с 1 (name1, name2 и так далее), и среди них находят такое наименьшее i, что namei еще не содержится в системе.

Выполнение задания:

|  |
| --- |
| i = int(input())  data = {}  for \_ in range(n):  name = input()  if name not in data:  data[name] = 1  print('OK')  else:  tmp\_num = data[name]  tmp\_name = name + str(tmp\_num)  while tmp\_name in data:  tmp\_num += 1  tmp\_name = name + str(tmp\_num)  print(tmp\_name)  data[name] = tmp\_num + 1  data[tmp\_name] = 1 |

**Задание 3\***

Текст задания:

Необходимо создать программу обработки запросов пользователей к файловой системе компьютера. Над каждым файлом можно производить следующие действия: запись – w ("write"), чтение – r ("read"), запуск – x ("execute").

Выполнение задания:

|  |
| --- |
| operation\_map = {  "read": "r",  "write": "w",  "execute": "x"  }  n = int(input())  files = {}  for \_ in range(n):  parts = input().split()  filename, permissions = parts[0], set(parts[1:])  files[filename] = permissions  m = int(input())  for \_ in range(m):  op, filename = input().split()  if operation\_map[op] in files[filename]:  print("OK")  else:  print("Access denied") |