# 1. ПРАКТИЧЕСКАЯ РАБОТА №1.

1.1 Задание 1

Задание 1 представлено на рисунке 1.1.

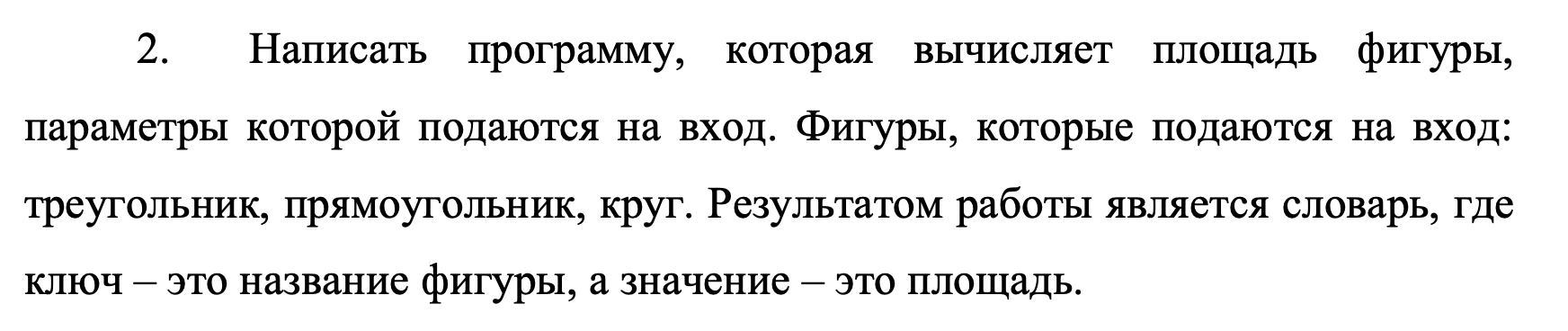


Рисунок 1.1 ー Задание 1

Код задания представлен в листинге 1.1.

Листинг 1.1 ー Код Задания 1

tri\_a,h = input("Введите сторону и высоту треугольника: ").split()

sq\_a = input("Введите сторону квадрата: ")

R = input("Введите радиус окружности: ")

squares = {"Треугольник": 1/2\*int(tri\_a)\*int(h), "Квадрат": int(sq\_a)\*\*2, "Окружность": 3.14\*int(R)\*\*2}

print(squares)

Результат выполнения Задания 1 представлен на рисунке 1.2

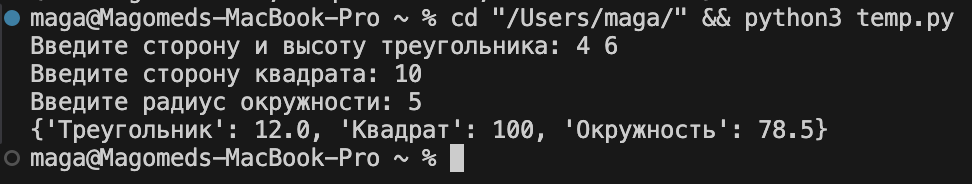


Рисунок 1.2 ー Результат выполнения Задания 1

1.2 Задание 2

Задание 2 представлено на рисунке 1.3.

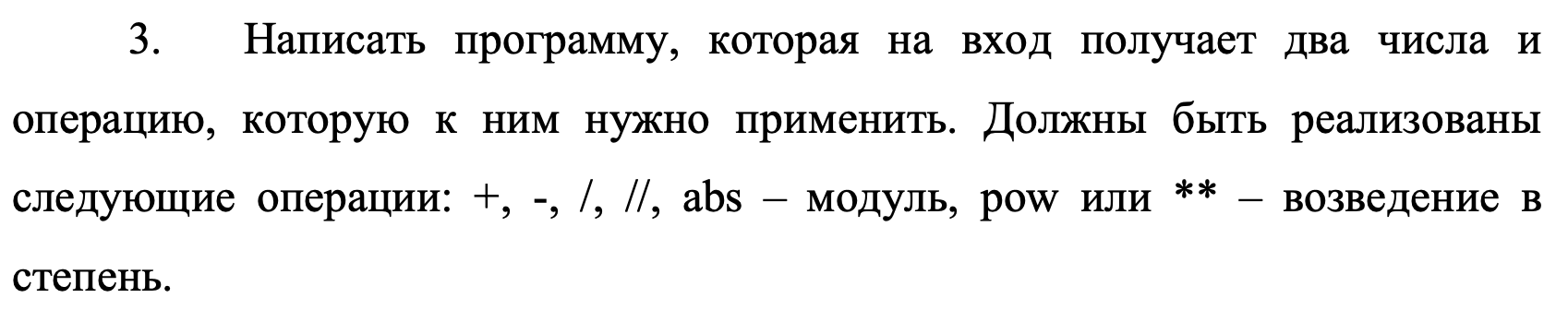


Рисунок 1.3 ー Задание 2

Код задания представлен в листинге 1.2.

Листинг 1.2 ー Код Задания 2

a, b, operator = input("Введите a, b и оператор: ").split()

a = int(a)

b = int(b)

if operator == "+":

print(a+b)

elif operator == "-":

print(a-b)

elif operator == "\*":

print(a\*b)

elif operator == "/":

print(a/b)

elif operator == "//":

print(a//b)

elif operator == "\*\*" or operator == "pow":

print(a\*\*b)

elif operator == "abs":

print(abs(a))

Результат выполнения Задания 2 представлен на рисунке 1.4

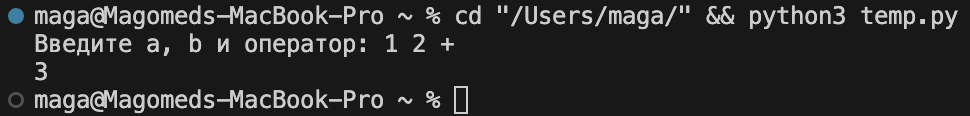


Рисунок 1.4 ー Результат выполнения Задания 2

1.3 Задание 3

Задание 3 представлено на рисунке 1.5.

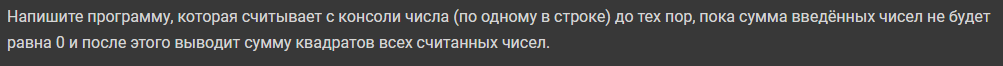


Рисунок 1.5 ー Задание 3

Код задания представлен в листинге 1.3.

Листинг 1.3 ー Код Задания 3

sum = int(input())

square\_sum = sum\*\*2

while sum != 0:

sum += int(input())

square\_sum += sum\*\*2

print(square\_sum)

Результат выполнения Задания 3 представлен на рисунке 1.6

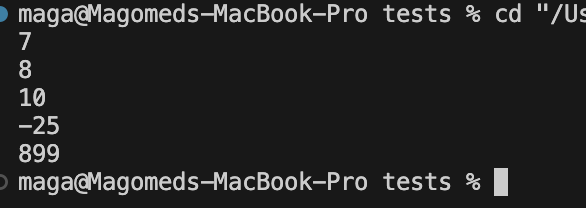


Рисунок 1.6 ー Результат выполнения Задания 3

1.4 Задание 4

Задание 4 представлено на рисунке 1.7.

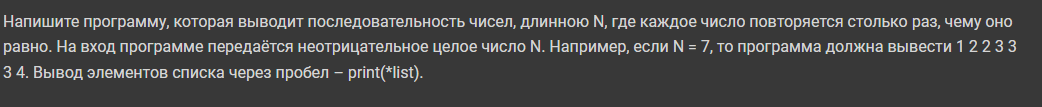


Рисунок 1.7 ー Задание 4

Код задания представлен в листинге 1.4.

Листинг 1.4 ー Код Задания 4

N = int(input())

a=[]

for i in range (N):

for k in range(i):

a.append(i)

if len(a) >= N:

break

print(\*a[:N])

Результат выполнения Задания 4 представлен на рисунке 1.8

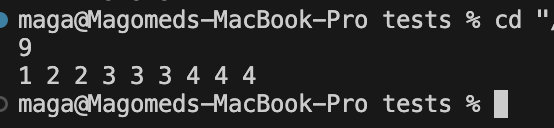


Рисунок 1.8 ー Результат выполнения Задания 4

1.5 Задание 5

Задание 5 представлено на рисунке 1.9.

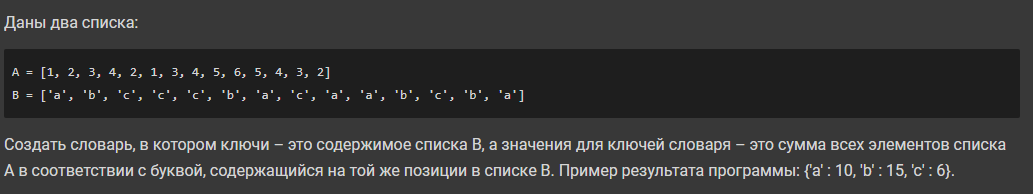


Рисунок 1.9 ー Задание 5

Код задания представлен в листинге 1.5.

Листинг 1.5 ー Код Задания 5

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

B = ['a', 'b', 'c', 'c', 'c', 'b', 'a', 'c', 'a', 'a', 'b', 'c', 'b', 'a']

result = {}

for i in range(len(A)):

updated = {B[i]: result[B[i]] + A[i] if B[i] in result else A[i]}

result.update(updated)

print(result)

Результат выполнения Задания 5 представлен на рисунке 1.10

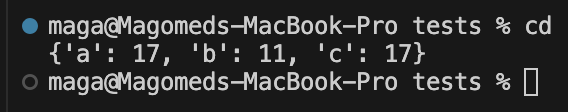


Рисунок 1.10 ー Результат выполнения Задания 5

1.6 Задание 6

Задание 6 представлено на рисунке 1.11.



Рисунок 1.11 ー Задание 6

Так как загруженный датасет является sklearn.utils.\_bunch.Bunch, а дальнейшие задания для pandas.DataFrame, необходимо перевести скачанный датасет в нужный формат.

Код задания представлен в листинге 1.6.

Листинг 1.6 ー Код Задания 6

from sklearn.datasets import fetch\_california\_housing

import pandas as pd

import numpy as np

data=fetch\_california\_housing(as\_frame=True)

data1 = pd.DataFrame(data= np.c\_[data['data'], data['target']], columns= data['feature\_names'] + ['target'])

1.7 Задание 7

Задание 7 представлено на рисунке 1.12.



Рисунок 1.12 ー Задание 7

Код задания представлен в листинге 1.7.

Листинг 1.7 ー Код Задания 7

data1.info()

Результат выполнения Задания 7 представлен на рисунке 1.13

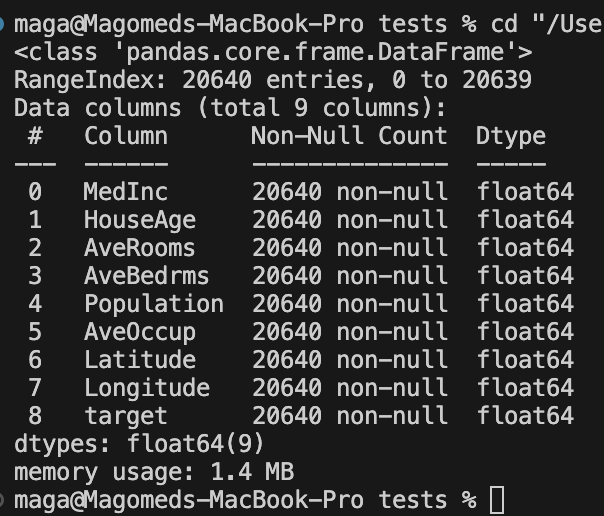


Рисунок 1.13 ー Результат выполнения Задания 7

1.8 Задание 8

Задание 8 представлено на рисунке 1.14.



Рисунок 1.14 ー Задание 8

Код задания представлен в листинге 1.8.

Листинг 1.8 ー Код Задания 8

print(data1.isna().sum())

Результат выполнения Задания 8 представлен на рисунке 1.15

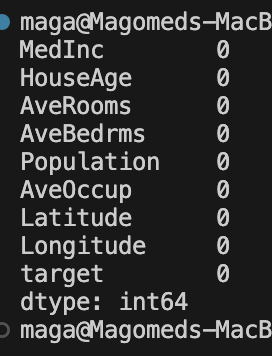


Рисунок 1.15 ー Результат выполнения Задания 8

1.9 Задание 9

Задание 9 представлено на рисунке 1.16.



Рисунок 1.16 ー Задание 9

Код задания представлен в листинге 1.9.

Листинг 1.9ー Код Задания 9

print(data1.loc[(data1['HouseAge']>50) & (data1['Population']>2500)])

Результат выполнения Задания 9 представлен на рисунке 1.17

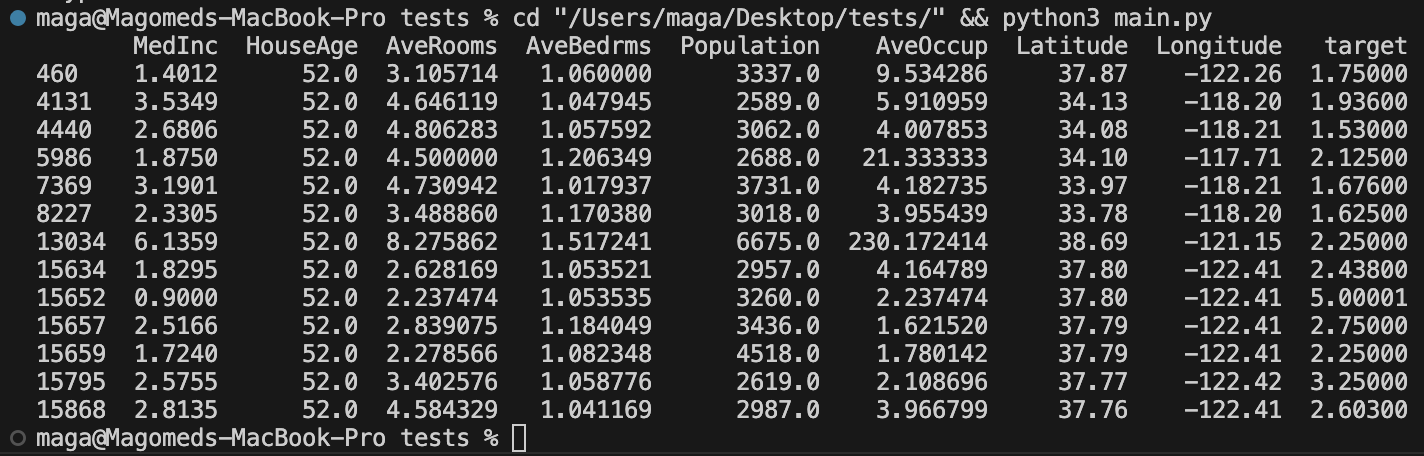


Рисунок 1.17 ー Результат выполнения Задания 9

1.10 Задание 10

Задание 10 представлено на рисунке 1.18.



Рисунок 1.18 ー Задание 10

Код задания представлен в листинге 1.10.

Листинг 1.10ー Код Задания 10

print(data1.MedInc.max())

print(data1.MedInc.min())

Результат выполнения Задания 10 представлен на рисунке 1.19

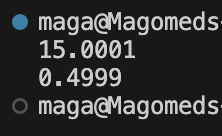


Рисунок 1.19 ー Результат выполнения Задания 10

1.11 Задание 11

Задание 11 представлено на рисунке 1.20.

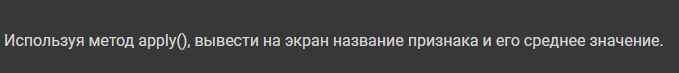


Рисунок 1.20 ー Задание 11

Код задания представлен в листинге 1.11.

Листинг 1.11ー Код Задания 11

def print\_mean(column):

mean\_value = column.mean()

print(f'Name: {column.name}, AvgValue: {mean\_value}')

data1.apply(print\_mean, axis=0)

Результат выполнения Задания 11 представлен на рисунке 1.21

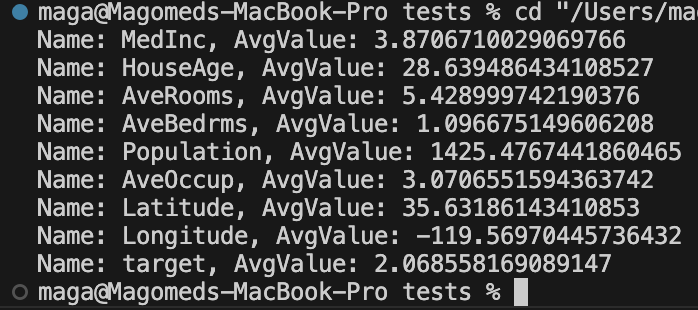


Рисунок 1.21 ー Результат выполнения Задания 11

1.12 Задание 12

Задание 12 представлено на рисунке 1.22.

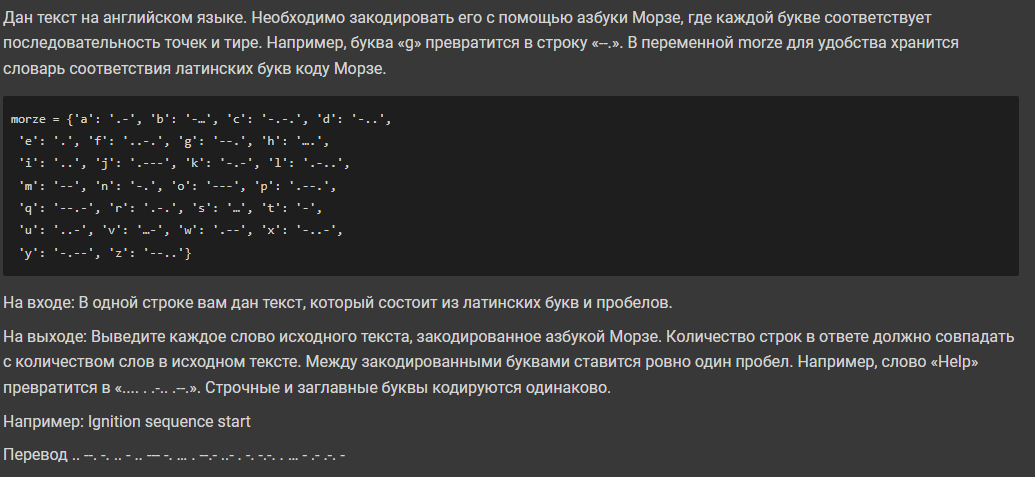


Рисунок 1.22 ー Задание 12

Код задания представлен в листинге 1.12.

Листинг 1.12ー Код Задания 12

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': '--..'}

text=input("Enter text: ")

text=text.lower()

for i in text:

if i==" ":

print("\n", end=" ")

else:

print(morze[i], end=" ")

Результат выполнения Задания 12 представлен на рисунке 1.23

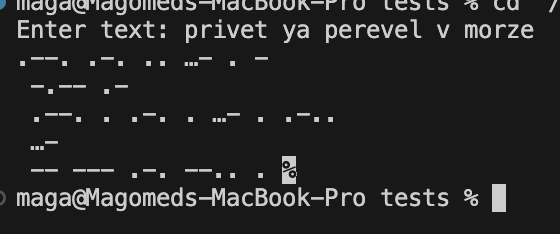


Рисунок 1.23 ー Результат выполнения Задания 12