

## ОГЛАВЛЕНИЕ

ОГЛАВЛЕНИЕ .....	1
ЛАБОРАТОРНАЯ РАБОТА №1 .....	3
Содержание работы .....	<b>Ошибка! Закладка не определена.</b>
Варианты заданий .....	4
Пример выполнения .....	6
Условие задачи .....	6
Код программы.....	8
Блок-схема .....	6
ЛАБОРАТОРНАЯ РАБОТА №2 .....	9
Варианты заданий .....	9
Пример выполнения .....	11
Условие задачи .....	11
Код программы.....	11
Блок-схема .....	<b>Ошибка! Закладка не определена.</b>
ЛАБОРАТОРНАЯ РАБОТА №3 .....	14
Варианты заданий .....	14
Пример выполнения .....	18
Условие задачи .....	18
Код программы.....	18
Блок-схема .....	18
ЛАБОРАТОРНАЯ РАБОТА №4 .....	20
Функции.....	20
Варианты заданий .....	20
Пример выполнения .....	20
Условие задачи .....	22
Код программы.....	22
Блок-схема .....	23
ЛАБОРАТОРНАЯ РАБОТА №5 .....	29
Варианты заданий .....	31
Пример выполнения .....	33
Условие задачи .....	33
Код программы.....	33
Блок-схема .....	33
ЛАБОРАТОРНАЯ РАБОТА №6 .....	36
Варианты заданий .....	36
Пример выполнения .....	37

Условие задачи .....	37
Код программы.....	37
Блок-схема .....	37
ЛАБОРАТОРНАЯ РАБОТА №7 .....	41
Варианты заданий .....	41
Пример выполнения .....	41
Условие задачи .....	42
Код программы.....	42
Блок-схема .....	42

## ЛАБОРАТОРНАЯ РАБОТА №1

### Введение в среду разработки PyCharm. Типы данных.

Цель работы: получить навыки разработки алгоритмов линейной структуры.

#### Содержание работы

1. Разработать алгоритм, описать его в виде блок-схемы и составить программу для решения задачи соответствующего варианта.
2. Набрать текст программы и отладить её в среде разработки PyCharm.
3. Оформить отчет о проделанной работе. Отчет должен включать в себя номер варианта, условие задачи, блок-схему, код программы, результаты работы программы, краткие ответы на контрольные вопросы.

#### Алгоритм выполнения работы

1. Запустить оболочку PyCharm, создать новый проект (Файл / Новый проект).
2. В появившемся окне выбрать пункт “Чистый Python” в контекстном меню, далее отметить радиокнопку “Текущий интерпретатор” и нажать “Создать”.

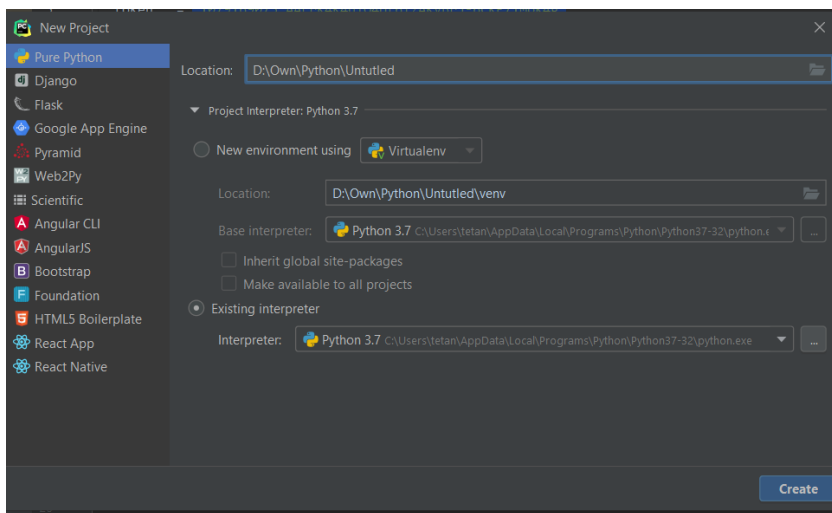


Рисунок 1. Окно создания проекта.

3. Создаём .py файл (Файл / Создать / Файл Python)
4. После написания кода программы, запускаем её (Запуск / Запуск).
5. Для отладки программы необходимо создать точку останова, нажав на пустое поле справа от номера строки. Результатом создания станет красный маркер, появившийся на месте нажатия.

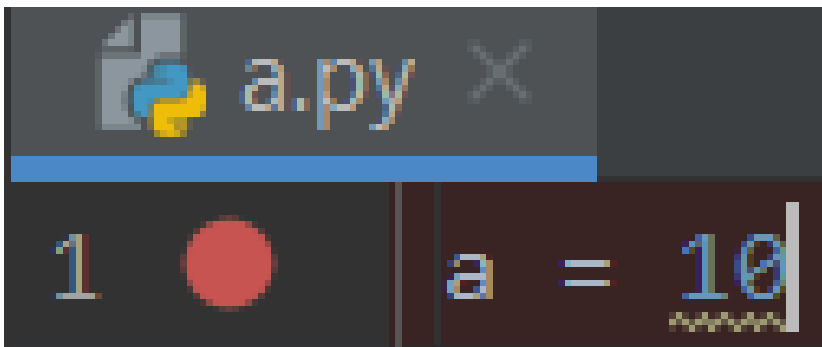


Рисунок 2. Точка останова.

6. Далее необходимо начать отладку (Запуск / Отладка). Код будет выполняться до точки останова.

#### Варианты заданий

№ варианта	Расчетная формула	X	y	z
1	2	3	4	5
1	$a = \frac{2lgx * cos^2 y}{ \frac{secx + ctgzy^3}{z}  + cos^2 y}$	3,017	$\frac{\pi}{7}$	-2,69
2	$a = z - \frac{y^7}{x^5} - \frac{x^5}{y^7} ctg(z^4 - x^2)$	2,7	1,83	-0,789
3	$a = \frac{1 + sin^2(x + y)}{ 2x / sin^2(x + y) } + z$	$70^0$	$\frac{\pi}{3}$	10
4	$a = y + \frac{x}{y^2 + \left  \frac{sec^2 y}{z + sec^2 y / z} \right }$	15,3	$81^0$	8,91
5	$a = \frac{\sqrt{\left  \frac{cos^4 yz}{sin^3 xz} * x \right } - \sqrt[9]{\left( \left  \frac{cos^4 yz}{sin^3 xz} * x \right  \right)^5}}{yz * \sqrt[29]{\left( \left  \frac{cos^4 yz}{sin^3 xz} * x \right  \right)^{14}}}$	-1	0,59	5,17
6	$a = x^{-9} + \frac{y + e^{x^2 y-tgz }}{z + e^{x^2 y-tgz }}$	$35^0$	0,95	-1,05
7	$a = \frac{1 + cos(y - 2) + x^2 y - z}{x^4 / 2 + cos(y - 2)}$	2,5	$33^0$	5,8

№ варианта	Расчетная формула	X	y	z
8	$a = 1 - \frac{cos  tgx^2 - sin z }{\sqrt{y + cos  tgx^2 - sin z }}$	$15^0$	0,5	1,95
9	$a = \frac{\sqrt[5]{e^{3x}}}{\sqrt[5]{e^{3x}} + 1,5tg3z * \sqrt[5]{e^{3x}}}$	-0,5	1,3	$5^0$
10	$b = e^{cos \frac{1}{xz}} + lg(lncos \frac{1}{xz} + sin x)$	$41^0$	10,18	3,72
11	$a = \frac{lg(\ln(sin(2z + 7)) + e^x) + \sqrt[3]{xy + \ln(sin(2z + 7))}}{y + \ln(sin(2z + 7))}$	2	2,1	$41^0$
12	$a = \frac{\ln 7x - 15  * cos^4(\ln 5y + tg2z)}{\ln y + 5 + tg(cos^4(\ln 5y + tg2z))}$	2,8	5,2	$8^0$
13	$a = \frac{cos(\ln \left  \frac{y}{2x + z} \right )}{(xyz)^8 * cos(\ln \left  \frac{y}{2x + z} \right )}$	2,5	5,8	$33^0$

14	$a = \frac{\sin(tg(x + 12^\circ) + 0,105y) - \lg(xyz)}{3z^y - \sin(tg(x + 12^\circ) + 0,105y)}$	$18^0$	$4^0$	3,5
15	$a = \frac{\sqrt[5]{\cos^5(\lg  \sin((xyz)^5) ) - 10^{3z}}}{\cos^5(\lg  \sin((xyz)^5) ) * 10^{-3z}}$	$\frac{\pi}{3}$	4,021	-5,72
16	$a = \frac{\ln \sqrt[3]{x - 2y * \ln(5y^{3z})}  + \lg(\cos(xy))}{xy - 109z * \ln \sqrt[3]{x - 2y * \ln(5y^{3z})} }$	4,6	2,67	-1,123
17	$a = \frac{tg^3(0,5y +  \cos z )}{10x + ctg^2(tg^3(0,5 +  \cos z ))}$	0,1	$\frac{\pi}{5}$	$63^0$
№ варианта	Расчетная формула	<b>X</b>	<b>y</b>	<b>z</b>
18	$a = \sqrt[5]{\cos(3z - xy)} - \frac{\sqrt[5]{\cos(3z - xy)}}{\sqrt[5]{\sin^3(3z - xy)}}$	21,5	$75^0$	7,56
19	$b = \frac{y \cos(\ln(x tg \frac{y}{2}))/7 + 81z + \sqrt[9]{\cos(\ln(x tg \frac{y}{2}))}}{z + \sqrt[3]{\cos(\ln(x tg \frac{y}{2}))}}$	-2	0,61	4,12
20	$a = \frac{0,08 \log_3^2 \sec^2 3x - e^{2x}}{\sqrt[5]{\log_3^2 \sec^2 3x - y} - 7z}$	$40^0$	2,35	-3,48
21	$a = \frac{\ln(1 + \sin z - 0,5 ) + 0,07y}{y^{2x} + \sin \ln(1 + \sin z - 0,5 ) }$	4,05	3,1	$40^0$
22	$a = \frac{\sin^3(\sqrt[5]{(5 + xy)})}{10 \ln \sin^3(\sqrt[5]{(5 + xy)})} + z$	2,6	$11^0$	41
23	$a = \frac{2 \arcsin(\sqrt{tzy^3})}{7x - \log_8^3 \arcsin(\sqrt{zy^3})}$	1,5	$20^0$	3,06
24	$a = \frac{\ln \arctg(y - 2x + \sin z ) }{\arctg(y - 2x +  z )^{-9}}$	$15^0$	2,1	-0,78
25	$a = \frac{\log_3(\ln \sqrt[3]{5 - z} ) + \ln \sqrt[3]{5 - z} }{22 - 4z - \sin xy}$	$31^0$	21,67	5,18
26	$a = \frac{\sqrt{\cos^3(\ln(tg^2 0,2x))} \sqrt{\cos^3(\ln(tg^2 0,2x))}}{\sqrt[5]{\cos^3(\ln(tg^2 0,2x))} + tg(y - 4)} - \cos z$	3	4,51	$51^0$
27	$a = \frac{15 \sin^5(\sqrt[4]{e^{2y+7z}}) - \cos x}{8yz + 0,128 \sin^5(\sqrt[4]{e^{2y+7z}})}$	$18^0$	7,3	3,5

1. № варианта	Расчетная формула	X	y	z
28	$a = \frac{\sin(\frac{x}{tgyz} + 0,06yx^2) - \lg(8 + z)}{6z^3 + \lg(\sin(\frac{x}{tgyz} + 0,06yx^2))}$	41 <sup>0</sup>	5,8	3,2
29	$a = y + \frac{\sin^{tg(\ln(xy+z*3))}(\frac{x}{z})}{\sin^{tg(\ln(xy+z*3))}(\frac{x}{z}) - 3,00102 + z^2}$	23 <sup>0</sup>	70 <sup>0</sup>	8,3
30	$a = \frac{2lgx * \sin^2 y}{ \frac{\sec x + ctgzy^3}{z}  + \sin^2 y}$	30 <sup>0</sup>	60 <sup>0</sup>	4

### Пример выполнения

#### I. Постановка задачи

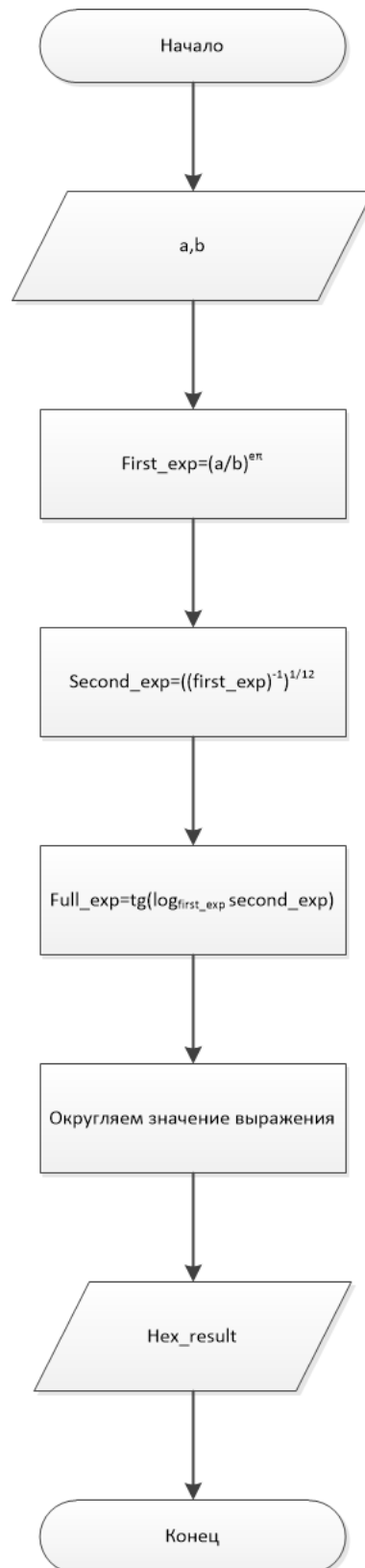
В консоль вводится два рациональных положительных числа a, b. Вывести значение следующего выражения, округлённого вниз, в шестнадцатеричной системе счисления:

$$\tan \log_{\left(\frac{a}{b}\right)^{e\pi}} \sqrt[12]{\left(\frac{b}{a}\right)^e}$$

#### II. Этапы решения задачи

1. Определить исходные значения – a, b
2. Выполнить анализ ОДЗ исходных значений для заданного выражения
3. Используя вспомогательные переменные - first\_exp, second\_exp, full\_exp, hex\_result, выполнить детализацию заданного выражения.
4. Вывести на экран результат выражения

#### III. Описание алгоритма решения задачи в виде блок-схемы



#### IV. Код программы

```
# Импортируем библиотеку math для использования сложных арифметических операций
import math

# Вводим из консоли числа a, b. С помощью функции int() переводим их в вещественный тип
a = float(input())
b = float(input())

# Разбиваем выражение на составные части. Вычисляем их значения одного из них.
first_exp = math.pow(a / b, math.e)

# Вычисляем значение корня. Представляем b/a в степени e, как a/b в степени e в степени -1
second_exp = math.pow(math.pow(first_exp, -1), 1/12)

# Вычисляем полное значение выражения.
full_exp = math.tan(math.log(math.pow(first_exp, math.pi), second_exp))

# Округляем значение выражения вверх и переводим его в шестнадцатеричную систему с помощью стандартной функции hex
hex_result = hex(math.ceil(full_exp))

# Выводим получившийся результат в шестнадцатеричной системе
print(hex_result)
# Выводим получившийся результат

print(full_exp)
```

#### Контрольные вопросы

1. Что такое литералы?
2. Приведите основные принципы PEP8.
3. Какие типы данных относятся к скалярным, а какие к структурированным?
4. Какие типы данных вы знаете?
5. Что такое Nonetype, и для чего он нужен?
6. Что делает функция print()?
7. Как задать f-строку?
8. Какие способы форматирования строк вы знаете?
9. Приведите пример каждого из видов форматирования строк.
10. Приведите пример расширенного форматирования чисел.
11. Сформулируйте определение алгоритма.
12. Перечислите и дайте определения свойствам алгоритма.
13. Какие существуют способы записи алгоритмов? Какие из них, по вашему мнению, чаще применяются на практике? Почему?
14. Сформулируйте определение оператора присваивания?
15. Почему желательно выводить на экран подсказку перед вводом данных?
16. Дайте определение форматного вывода? Когда необходимо применение?
17. Выберите правильные имена переменных в Python:

1 Vasya @mail\_ru m11 Petya123 CY\_27 lenta.ru 1m



Митин брат    \_27    "Pes barbos"    m 1 Quo vadis

18. Что будет выведено в результате работы фрагмента программы:

а) a = 5; b = 3

```
print ( a, "Z(", b, ")", sep = "" )
```

б) a = 5; b = 3

```
print ( "Z(a)=", "(b)", sep = "" )
```

в) a = 5; b = 3

```
print ( "Z(", a, ")=( ", a+b, ")", sep = "" )
```

19. Какие данные записываются в логические переменные?

20. Расскажите об особенностях переменных в языке Python. Почему может получиться, что изменение одной переменной автоматически приводит к изменению другой?

21. Что такое приоритет операций? Зачем он нужен?

22. В каком порядке выполняются операции, если они имеют одинаковый приоритет?

23. Зачем используются скобки?

24. Что происходит, если в выражения входят переменные разных числовых типов? Какого типа будет результат?

25. Опишите операции // и %.

26. Какие стандартные математические функции вы знаете? В каких единицах задается аргумент тригонометрических функций?

27. Как выполнить округление вещественного числа к ближайшему целому?

28. Какие числа называют случайными? Зачем они нужны?

## ЛАБОРАТОРНАЯ РАБОТА №2

### Условные конструкции. Циклы.

Цель работы: приобрести навыки работы с условными конструкциями, циклами.

#### Содержание работы

1. Разработать алгоритм, описать его в виде блок-схемы и составить программу для решения задачи соответствующего варианта.
2. Набрать текст программы и отладить её в среде разработки PyCharm.
3. Оформить отчет о проделанной работе. Отчет должен включать в себя номер варианта, условие задачи, блок-схему, код программы, результаты работы программы, краткие ответы на контрольные вопросы.

#### Варианты заданий

1. С клавиатуры вводятся последовательность натуральных чисел. Признак конца ввода - 0. Определить первое число, идущее после первого отрицательного. Если их нет, вывести сообщение.

2. С клавиатуры вводятся вещественные числа. Признаки конца ввода – 0. Определить сумму всех чисел в интервале между 0 и 3.
3. С клавиатуры вводятся вещественные числа. Признак конца ввода – 0. Определить количество отрицательных и положительных чисел.
4. Вводится последовательность символов. Признак конца ввода – точка. Посчитать количество пробелов.
5. С клавиатуры вводятся целые числа. Признак конца ввода – отрицательное число. Посчитать сумму максимального и минимального элемента.
6. Вводится последовательность символов. Признак конца ввода – точка. Посчитать количество символов, не являющихся пробелами.
7. Вводится последовательность вещественных чисел. Признак конца ввода – 0. Определить является ли последовательность арифметической прогрессией.
8. С клавиатуры вводятся целые числа. Признак конца ввода – 0. Определить все числа, идущие перед отрицательными числами.
9. Вводится последовательность вещественных чисел. Признак конца ввода – 0. Определить является ли последовательность упорядоченной по убыванию.
10. Вводится последовательность целых чисел. Признак конца ввода отрицательное число. Определить является ли последовательность упорядоченной по возрастанию.
11. С клавиатуры вводится последовательность натуральных чисел. Признак конца ввода - 0. Вывести на экран максимальный элемент и его номер.
12. С клавиатуры вводится последовательность латинских символов. Признак конца ввода – точка. Посчитать количество гласных букв.
13. Вводится последовательность латинских символов. Признак конца ввода – точка. Определить каких букв больше: гласных или согласных.
14. Даны два числа. Определить наименьшее общее кратное.
15. С клавиатуры вводятся символы. Признак конца ввода – '0' Определить, сколько было введено пар следующих друг за другом символов, которые и в таблице ASCII расположены в таком же порядке.
16. Вводится последовательность натуральных чисел. Признак конца ввода – 0. определить является ли последовательность геометрической прогрессией.
17. Даны два натуральных числа. Определить наибольший общий делитель.
18. Дано натуральное число. Определить являются ли цифры данного числа упорядоченными по неубыванию или по невозрастанию. Если цифры не упорядочены, то вывести среднее значение суммы данных цифр.
19. С клавиатуры вводится последовательность вещественных чисел. Признак конца ввода – 0. Определить полусумму дробной части всех чисел в интервале от 0 до 5.
20. С клавиатуры вводятся числа. Признак конца ввода – 0. Найти минимальное число и вывести на экран число, предшествующее минимальному.
21. Вводится последовательность символов. Признак конца ввода – точка. Вывести на экран символы, не являющиеся символами цифр.
22. Вводится последовательность символов. Признак конца ввода – точка. Определить количество двухбуквенных слов в данной последовательности. Слова отделяются друг от друга пробелами.
23. Дано целое неотрицательное число. Определить является ли оно палиндромом.

24. Вводится последовательность символов. Признак конца ввода – точка. Посчитать сколько раз встречается символ «+», и сколько раз символ «\*». Если они не встречаются, то вывести сообщение.

25. Найти наименьшее натуральное число  $M$ , кратное 6, для которого  $\frac{\sqrt{|x|}}{M} < \varepsilon$ , где  $\varepsilon = 0,01$ ,  $x$  – заданное число и вычислить сумму  $S = \sum_{n=1}^M \frac{M}{n+|x|}$ .

26. Дано число  $a$ . С клавиатуры вводится последовательность натуральных чисел. Признак конца ввода – 0. Если есть хотя бы один член равный  $a$ , то получить сумму всех членов, следующих за первым таким членом. В противном случае ответом должно быть число -10.

27. Вводится последовательность натуральных чисел. Признак конца ввода – 0. Определить количество соседств двух чисел разного знака.

28. Дана последовательность, состоящая из  $k$  вещественных чисел. Найти порядковый номер того из них, которое наиболее близко к целому числу  $n$ , значение которого вводится с клавиатуры.

29. Дана непустая последовательность натуральных чисел, за которой следует 0. Вычислить сумму тех из них, порядковые номера которых – числа Фибоначчи.

30. Дано 10 вещественных чисел. Вычислить разность между максимальным и минимальным из них.

### Пример выполнения

#### I. Постановка задачи

С клавиатуры в консоль вводится последовательность натуральных чисел  $a_1, a_2, a_3 \dots a_n$ , разделённая знаком конца строки. Признак конца ввода – 0. Вывести максимальное простое число. Гарантируется, что в последовательности встречается хотя бы одно простое число.

#### II. Разработка этапов решения задачи

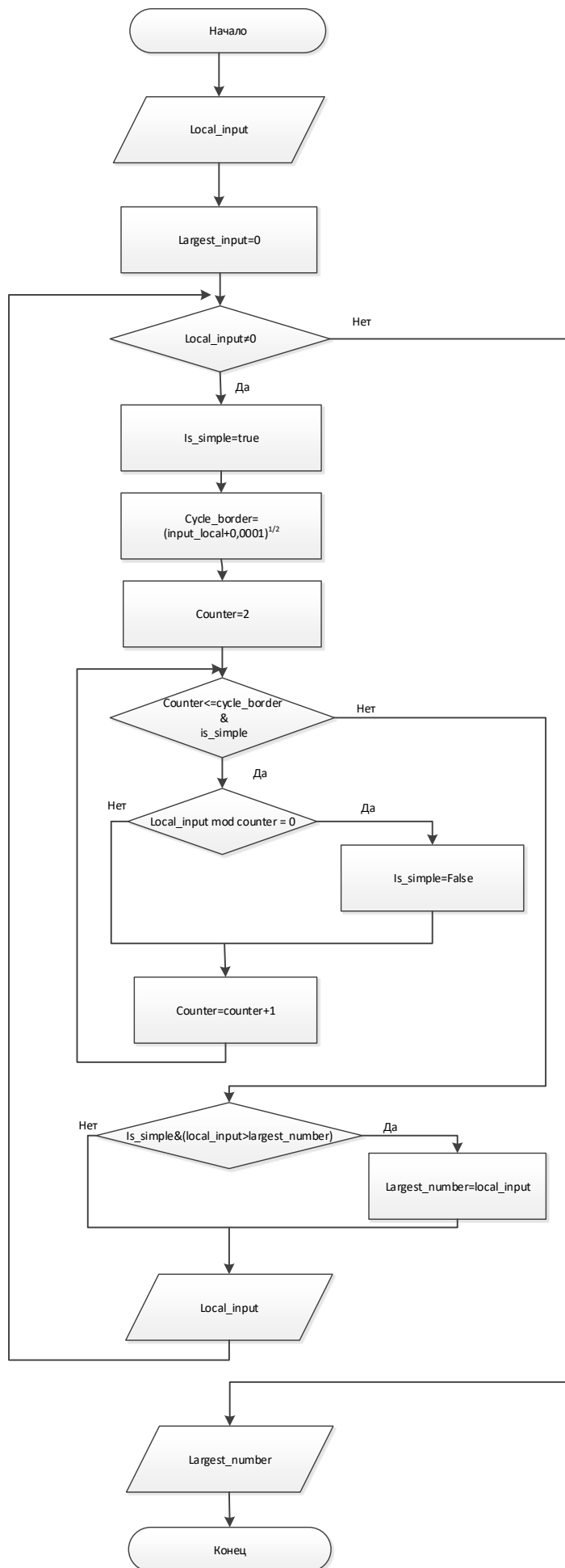
1. Исходные данные: последовательность натуральных чисел  $a_1, a_2, a_3 \dots a_n$  (для хранения очередного значения элемента из последовательности в программе используем - `local_input`)

2. Результирующее значение: максимальное простое число  $a_n$  (для хранения значения в программе используем `largest_number`)

3. Вспомогательные переменные

4. Ограничение на число итераций цикла

#### III. Описание алгоритма решения задачи в виде блок-схемы



#### IV. Код программы

```
# Импортируем библиотеку math для использования сложных арифметических операций
import math

# Вводим из консоли первое число последовательности. В дальнейшем эта переменная
будет хранить все введенные числа.
local_input = int(input())

# Объявляем переменную, хранящую значение наименьшего числа
largest_number = 0

# Запускаем цикл while, выполняющийся до тех пор, пока очередное значение
переменной local_input не станет нулём
while local_input != 0:

    # Объявляем логическую переменную, хранящую значение простоты числа
    local_input (простое -> true, сложное -> false)
    is_simple = True

    # Объявляем целочисленную переменную, в которую передаём значение корня из
    local_input.
    # + 0.0001 необходимо для корректной работы метода вычисления корня
    cycle_border = int(pow(local_input + 0.0001, 0.5))

    # Запускаем цикл while, который выполняется пока логическое выражение истинно
    counter <= cycle_border & is_simple
    # Внутри цикла число local_input проверяется на простоту
    while counter <= cycle_border & is_simple:

        # Проверяем число local_input на делимость на i
        if local_input % i == 0:

            # Присваиваем переменной isSimple значение false (число inputNumber не
            простое)
            is_simple = False

        # Проверяем переменную local_input на простоту и на то, что она больше, чем
        значение largest_number
        if is_simple and (local_input > largest_number):
            largest_number = local_input

    # Считываем из консоли новое число
    local_input = int(input())

# Выводим значение переменной largest_number
print(largest_number)
```

### Контрольные вопросы

1. Какие виды алгоритмов называются разветвляющимися?
2. Опишите механизм работы условного оператора.
3. Приведите пример описания условного оператора с помощью блок-схемы.
4. Какие алгоритмы называются циклическими?
5. Какие виды структур повторения вам известны?
6. Приведите пример каждого вида с помощью блок-схемы.
7. Как реализовать цикл со счетчиком?
8. Что такое итератор?
9. Какие функции, используемые для итерируемых объектов, вы знаете?
10. Что они возвращают?
11. Чем отличаются разветвляющиеся алгоритмы от линейных?
12. Почему нельзя выполнить обмен значений двух переменных в два шага:  $a=b$ ;  $b=a$ ?

## ЛАБОРАТОРНАЯ РАБОТА №3

### Работа со строками. Работа со списками.

Цель работы: приобрести практический опыт работы при использовании строк, а также одномерных и двумерных списков.

#### Содержание работы

1. Разработать алгоритм, описать его в виде блок-схемы и составить программу для решения задачи соответствующего варианта из каждой части (I – задание со строкой; II – задание с списком).
2. Набрать текст программы и отладить её в среде разработки PyCharm.
3. Оформить отчет о проделанной работе. Отчет должен включать в себя номер варианта, условие задачи, блок-схему и код программы.

#### Варианты заданий

##### Часть I

1. Преобразовать строку, изменив порядок следования слов в строке на обратный.
2. Преобразовать строку таким образом, чтобы цифры каждого слова были перенесены в конец слова, изменив порядок следования их в слове на обратный.
3. Дана строка, состоящая из слов. Преобразовать строку, перенося слова с повторяющимися символами в конец этой строки, оставляя только один символ в каждой последовательности подряд идущих одинаковых символов.
4. Дана строка, состоящая из слов. Преобразовать строку таким образом, чтобы слова были упорядочены по возрастанию своей длины.
5. Дана строка, состоящая из слов. Удалить из строки слова, содержащие повторяющиеся символы.

6. Преобразовать строку таким образом, чтобы цифры каждого слова были перенесены в конец слова без изменения порядка следования их в слове.
7. Преобразовать строку, записав символы каждого слова этой строки в обратном порядке.
8. Дана строка, состоящая из символов. Удалить из этой строки все цифры, записав в другую строку символы, соответствующие коды которых являются этими цифрами в таблице ASCII.
9. Дана строка, состоящая из букв и цифр. Преобразовать строку из слов таким образом, чтобы каждое *i*-слово состояло из количества букв, которое соответствует значению *i* - цифры. При этом, все числовые значения, использованные для формирования строки должны быть удалены. Если символов в строке меньше, чем сумма цифр, содержащихся в ней, и не все цифры использованы для её формирования, то неиспользованные цифры перенести в конец строки. Например: исходная строка - «Мамамыл4а4рам4у3.», полученная строка - «Мама мыла раму3».
10. Дана строка, состоящая из символов и цифр. Повторить символ, предшествующий цифре *n* - раз, где *n* - значение цифры.
11. Составить программу, которая преобразует строку к строке указанного размера, путём добавления пробелов между словами.
12. Дана строка, состоящая из слов, разделённая в некоторых местах запятыми. Поменять местами фрагменты строки, разделённые запятыми.
13. Дана строка. Определить, есть ли в данной строке одинаковые слова, и заменить каждое повторяющееся слово, словом «повтор», начиная со второго.
14. Даны две строки, состоящие из слов. Получить третью строку, состоящую из слов, которые содержатся в одной строке и не содержатся в другой.
15. Дана строка, состоящая из слов. Преобразовать строку так, чтобы каждое слово было отражено зеркально.
16. Определить, является ли все слова строки идентификаторами, т.е. начинается ли оно с английской буквы в любом регистре или знака подчеркивания и не содержит других символов, кроме букв английского алфавита (в любом регистре), цифр и знака подчеркивания.
17. Вводится строка. Удалить из нее все пробелы. После этого определить, является ли она палиндромом (перевертышем), т.е. одинаково пишется как с начала, так и с конца.
18. Найти в строке указанную подстроку и заменить ее на новую. Строку, ее подстроку для замены и новую подстроку вводит пользователь.
19. Вводится строка, содержащая буквы, целые неотрицательные числа и иные символы. Требуется все числа, которые встречаются в строке, поместить в отдельный целочисленный массив. Например, если дана строка "data 48 call 9 read13 blank0a", то в массиве должны оказаться числа 48, 9, 13 и 0.
20. Вводится строка. Требуется удалить из нее повторяющиеся символы и все пробелы. Например, если было введено "abc cde def", то должно быть выведено "abcdef".
21. Вводится ненормированная строка, у которой могут быть пробелы в начале, в конце и между словами более одного пробела. Привести ее к нормированному виду, т.е. удалить все пробелы в начале и конце, а между словами оставить только один пробел.
22. Вводится строка слов, разделенных пробелами. Найти самое длинное слово и вывести его на экран. Случай, когда самых длинных слов может быть несколько, не обрабатывать.
23. Дана строка. Перевернуть в строке каждое слово. Все пробелы должны быть сохранены.
24. Дана строка. Перевернуть порядок слов. Первое слово должно стать последним. Все пробелы должны быть сохранены.

25. Даны две строки, состоящие из слов. Получить третью строку, состоящую из слов, которые содержатся в одной строке и не содержатся в другой.
26. Дана строка, состоящая из слов. Преобразовать строку таким образом, чтобы слова были упорядочены по убыванию своей длины.
27. Дана строка, состоящая из слов. Преобразовать строку таким образом, чтобы слова были упорядочены по возрастанию своей длины.
28. Преобразовать данную строку, удалив из неё три слова, наименьшей длины.
29. Дана строка, состоящая из слов. Преобразовать строку так, чтобы каждое слово было отражено зеркально.
30. Дана строка, состоящая из слов и содержащая одну точку в конце одного из слов. Если в конце строки точка отсутствует, то найти слово, заканчивающееся точкой и перенести его в конец строки.

## Часть II

1. Дан массив целых чисел. Получить упорядоченный по невозрастанию массив, состоящий из членов данного массива, заключенных между первым и последним положительным элементом данного массива. Указание: проверить массив на наличие элементов между ними.
2. Дан массив целых чисел размера  $n(n=10)$ . Получить упорядоченный по возрастанию массив, содержащий все различные числа данного массива.
3. Дан массив  $x$  целых чисел. Получить упорядоченную по возрастанию последовательность из чисел, которые встречаются в данном массиве более двух раз(если такие имеются).
4. Переписать отрицательные элементы массива  $x$ , заключённые между минимальным и максимальным элементами, в массив  $y$ , упорядоченными по убыванию и подсчитать их количество. Указание: проверить массив  $x$  на наличие в нём отрицательных элементов.
5. Дан массив  $x$  целых чисел. Вывести в порядке возрастания все числа, стоящие на четных местах и встречающиеся более двух раз.
6. Дан действительный массив  $x$  размера  $n(n=10)$ . Расположить элементы массива по убыванию. Использовать сортировку обменом.
7. Дан целочисленный массив  $x$ , упорядоченный по неубыванию и некоторое целое число  $a$ , для которого нужно найти такое место среди элементов массива, чтобы после вставки числа  $a$  на это место упорядоченность не нарушилась.
8. Дан вещественный массив  $x$  размера  $n(n=10)$ . Переставить элементы массива таким образом, чтобы вначале в массиве шла группа элементов больших первого элемента в исходном массиве, потом - группа элементов меньших или равных ему.
9. Дан целочисленный массив  $x$  размера  $n(n=10)$ , упорядоченный по неубыванию. Найти среднее арифметическое элементов массива и вставить его в массив так, чтобы не нарушилась упорядоченность.
10. Дан массив  $x$  целых чисел. Исключить из него все числа, встречающиеся более двух раз(если такие имеются). Затем числа, стоящие на четных местах, упорядочить по невозрастанию.
11. Дан массив  $x$ , состоящий из латинских букв. Вывести его так, чтобы его элементы следовали в алфавитном порядке в массиве  $a$ , а в массив  $b$  все повторяющиеся элементы.
12. Дан массив натуральных чисел  $x$  размера  $n(n=10)$  и некоторое натуральное число  $k$ . Удалить из массива элемент с номером  $k$  и вставить элемент, равный  $p$ , так, чтобы не нарушилась упорядоченность.



13. Дан массив  $x$ , состоящий из латинских букв. Исключить из него элементы, стоящие на чётных местах, затем элементы стоящие на нечётных местах упорядочить по неубыванию.
14. Даны два действительных массива  $x$  и  $y$ . Объединить их так, чтобы в массиве-результате элементы были упорядочены по возрастанию.
15. Дан целочисленный массив  $x$ . Определить, сколько в нем пар соседних одинаковых элементов. Вывести пары соседних одинаковых элементов в массив  $y$ , упорядоченными по неубыванию.
16. Дан массив целых чисел размера  $n(n=10)$ . Получить упорядоченный по возрастанию массив, содержащий все различные числа данного массива.
17. Дан массив  $x$  целых чисел. Получить упорядоченную по возрастанию последовательность из чисел, которые встречаются в данном массиве более двух раз(если такие имеются).
18. Даны два целочисленных массива. Выяснить, является ли один из них подмножеством другого. Если является, то упорядочить это подмножество по убыванию.
19. Переписать отрицательные элементы массива  $x$ , заключённые между минимальным и максимальным элементами, в массив  $y$ , упорядоченными по убыванию и подсчитать их количество. Указание: проверить массив  $x$  на наличие в нём отрицательных элементов.
20. Даны два упорядоченных по убыванию массива, получить упорядоченную по возрастанию массив, состоящий из элементов первого массива, которых нет во втором.
21. Дан массив  $x$  целых чисел. Вывести в порядке возрастания все числа, стоящие на четных местах и встречающиеся более двух раз.
22. Дан действительный массив  $x$  размера  $n(n=10)$ . Расположить элементы массива по убыванию. Использовать сортировку обменом.
23. Дан целочисленный массив  $x$ , упорядоченный по неубыванию и некоторое целое число  $a$ , для которого нужно найти такое место среди элементов массива, чтобы после вставки числа  $a$  на это место упорядоченность не нарушилась.
24. Дан вещественный массив  $x$  размера  $n(n=10)$ . Переставить элементы массива таким образом, чтобы вначале в массиве шла группа элементов больших первого элемента в исходном массиве, потом - группа элементов меньших или равных ему.
25. Дан целочисленный массив  $x$  размера  $n(n=10)$ , упорядоченный по неубыванию. Найти среднее арифметическое элементов массива и вставить его в массив так, чтобы не нарушилась упорядоченность.
26. Дан массив  $x$  целых чисел. Исключить из него все числа, встречающиеся более двух раз(если такие имеются). Затем числа, стоящие на четных местах, упорядочить по невозрастанию.
27. Дан массив  $x$ , состоящий из латинских букв. Вывести его так, чтобы его элементы следовали в алфавитном порядке в массиве  $a$ , а в массив  $b$  все повторяющиеся элементы.
28. Дан массив натуральных чисел  $x$  размера  $n(n=10)$  и некоторое натуральное число  $k$ . Удалить из массива элемент с номером  $k$  и вставить элемент, равный  $p$ , так, чтобы не нарушилась упорядоченность.
29. Дан массив  $x$ , состоящий из латинских букв. Исключить из него элементы, стоящие на чётных местах, затем элементы стоящие на нечётных местах упорядочить по неубыванию.
30. Даны два действительных массива  $x$  и  $y$ . Объединить их так, чтобы в массиве-результате элементы были упорядочены по возрастанию.

## Пример выполнения

### I. Постановка задачи

В первой строке вводится три числа n, m, p. В последующих n строках вводятся случайные символы. Вывести первые m процентов строк, содержащих первые p процентов символов от среднего количества символов в строках.

### II. Разработка этапов решения задачи

1. Исходные данные - lines\_quantity, lines\_percentage, symbols\_percentage
2. Результат -

### IV. Код программы

```
# Вводим первую строку, разбиваем её на массив строк через пробел и переводим всё в числа
lines_quantity, lines_percentage, symbols_percentage = list(map(int,
input().split()))

# Инициализируем список строк и переменную, хранящую общее количество введенных символов
lines = []
symbols_quantity = 0

# Вводим n строк в список с клавиатуры
for i in range(lines_quantity):
    lines.append(input())

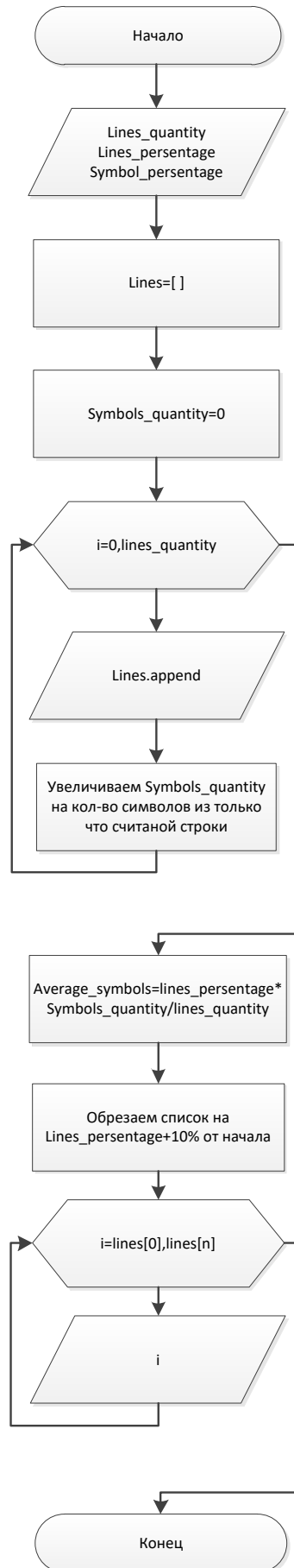
    # Добавляем количество символов из только что считанной строки
    symbols_quantity += len(lines[i])

# Записываем в переменную 70% от среднего значения символов в каждой строке
average_symbols = int(0.7 * symbols_quantity / lines_quantity)

# Обрезаем список lines на 80% от начала
lines = lines[:int(0.8 * lines_quantity)]

# Выводим перевёрнутые 70% символов от среднего числа в каждой строке
for i in lines:
    print(i[:average_symbols][::-1])
```

### III. Блок-схема



### Контрольные вопросы

1. Что такое строка?
2. Что нужно сделать для использования спец. символов?
3. Какие операции над строковым типом вы знаете?
4. Приведите примеры операций над строками.
5. Какие виды форматирования строк вы знаете?
6. Приведите пример для каждого вида форматирования.
7. Что такое список?
8. Назовите особенности списка.
9. Приведите примеры операций со списком.

## ЛАБОРАТОРНАЯ РАБОТА №4

### Функции.

Цель работы: приобрести практический опыт работы при работе с строками и списками.  
Приобрести навыки при разработке функций.

### Содержание работы

1. Разработать алгоритм, описать его в виде блок-схемы и составить программу для решения задачи соответствующего. Использовать функцию для решения поставленной задачи
2. Набрать текст программы и отладить её в среде разработки PyCharm.
3. Оформить отчет о проделанной работе. Отчет должен включать в себя номер варианта, условие задачи, блок-схему и код программы.

### Варианты заданий

1. Дан целочисленный массив  $x$  размера  $n$  ( $n=10$ ). Упорядочить по убыванию элементы массива, заключённые между минимальным и максимальным. Указание: проверить массив на наличие элементов между ними.
2. Дана целочисленная последовательность. Получить упорядоченную по возрастанию последовательность из чисел, которые входят в данную последовательность по одному разу.
3. Дан вещественный массив  $x$  размера  $n$  ( $n=10$ ). Удалить из него элементы, кратные двум (если такие имеются) и расположить элементы массива по возрастанию.
4. Переписать положительные элементы массива  $x$ , до последнего вхождения максимального элемента, в массив  $y$ , упорядоченными по неубыванию и подсчитать их количество. Указание: проверить массив  $x$  на наличие в нём положительных элементов.
5. Из двух отсортированных в убывающем порядке массивов получить новый массив, отсортированный в том же порядке.
6. Дан массив  $x$  вещественных чисел. Получить упорядоченный по невозрастанию массив, состоящий из чисел данного массива, находящихся между предпоследним и последним отрицательным членом. Указание: проверить массив на наличие элементов между ними.

7. Даны две последовательности. Получить упорядоченную по невозрастанию последовательность, состоящую из тех членов первой последовательности, которых нет во второй.
8. Дан целочисленный массив  $x$  размера  $n$ . Упорядочить по невозрастанию те элементы массива, которые не находятся между минимальным и максимальным (если такие имеются).
9. Если число  $a$  встречается в массиве вещественных чисел  $x$  размера  $n$  ( $n=10$ ), то упорядочить по неубыванию часть массива до последнего вхождения  $a$ , в противном случае упорядочить по невозрастанию часть массива между первым и пятым элементами.
10. Дан массив  $x$  размера  $n$  ( $n=10$ ) целых чисел, содержащий как положительные, так и отрицательные элементы. Упорядочить массив следующим образом: сначала идут отрицательные числа, упорядоченные по невозрастанию, потом положительные, упорядоченные по неубыванию.
11. Дан вещественный массив  $x$  размера  $n$  ( $n=10$ ). Удалить из него элементы, кратные двум (если такие имеются) и расположить элементы массива по возрастанию.
12. Дан массив  $x$  целых чисел. Упорядочить элементы, стоящие на четных местах по невозрастанию, а на нечетных – по неубыванию.
13. Дан массив  $x$  целых чисел, содержащий как положительные, так и отрицательные элементы. Исключить все положительные элементы, встречающиеся в массиве более одного раза, затем отрицательные упорядочить по убыванию.
14. Даны два целочисленных массива. Выяснить, является ли один из них подмножеством другого. Если является, то упорядочить это подмножество по убыванию.
15. Даны два упорядоченных по убыванию массива, получить упорядоченную по возрастанию массив, состоящий из элементов первого массива, которых нет во втором.
16. Дана строка, состоящая из слов, разделённых пробелами. Подсчитать среднюю длину слов этой строки и вывести все слова, длина которых отличается не более чем на два символа от средней.
17. Даны две строки, состоящие из слов. Получить строку, в которой чередуются слова первой и второй строки. Если в одной из строк слов больше, чем в другой, то оставшиеся слова этой строки дописать подряд в строку-результат.
18. Даны две строки. Пусть  $p1$  - число слов в первой строке, а  $p2$  - во второй ( $p1 < p2$ ). Добавить в конец первой строки, последние  $p2 - p1$  слов второй строки.
19. Дана строка, состоящая из слов, разделённых пробелами. Определить количество слов-палиндромов. Замечание: словом-палиндромом называется слово, состоящее из  $p$  букв, каждый символ  $i$  от 1 до  $p/2$ , которого является соответственно символом  $p-i+1$ . Например, слово «ДЕД» или «КАЗАК».
20. Дана строка, состоящая из слов, разделённых пробелами. Вывести слова этой строки, которые отличны от последнего слова.
21. Преобразовать строку таким образом, чтобы цифры каждого слова были перенесены в начало слова, изменив порядок следования цифр в слове на обратный.
22. Определить, есть ли в данной строке одинаковые слова.
23. Даны две строки, состоящие из слов. Получить третью строку, состоящую из слов, которые содержатся в обеих строках.
24. Дана строка, состоящая из слов. Вывести слова, которые повторяются хотя бы один раз.
25. Преобразовать строку, изменив порядок следования слов в строке на обратный.
26. Преобразовать строку таким образом, чтобы цифры каждого слова были перенесены в конец слова, изменив порядок следования их в слове на обратный. *Замечание: наличие цифр можно проверить с*

помощью проверки вхождения элемента во множество. Например, *Symbol in ['0'.. '9']*. Результат принимает истинное значение, если *Symbol* – цифра

27. Дана строка, состоящая из слов. Преобразовать строку, перенося слова с повторяющимися символами в конец этой строки, оставляя только один символ в каждой последовательности подряд идущих одинаковых символов.

28. Дана строка, состоящая из слов. Преобразовать строку таким образом, чтобы слова были упорядочены по возрастанию своей длины.

29. Дана строка, состоящая из слов. Удалить из строки слова, содержащие повторяющиеся символы.

30. Дана строка, состоящая из букв и цифр. Преобразовать строку из слов таким образом, чтобы каждое *i*-слово состояло из количества букв, которое соответствует значению *i* - цифры. При этом, все числовые значения, использованные для формирования строки должны быть удалены. Если символов в строке меньше, чем сумма цифр, содержащихся в ней, и не все цифры использованы для её формирования, то неиспользованные цифры перенести в конец строки. Например: исходная строка - «Мамамыл4а4рам4у3.», полученная строка - «Мама мыла раму3».

Пример выполнения

#### I. Постановка задачи

В первой строки вводятся два целых числа  $a$ ,  $b$ . В следующих  $a$  строках вводятся промежутки от  $k_{i1}$  до  $k_{i2}$ . Гарантируется, что  $k_{i2} < k_{i+1 1}$ . После вводится  $b$  целых чисел  $p_{1...n}$ . Необходимо для каждого числа  $p$  определить, какому промежутку он относится.

#### II. Разработка этапов решения задачи

1. Решение задачи выполнено с использованием бинарного поиска, реализованного в подпрограмме `binary_search`

2. Разработаны спецификации к каждой подпрограмме:

##### Процедура `binary_search`

Назначение: бинарный поиск

Входные данные: начальная и конечная точки, результирующее значение, значение

Возвращаемые данные: начальное значение

##### Процедура `read_arguments`

Назначение: считывание данных

Входные данные: нет

Возвращаемые данные: количество, промежуток

##### Процедура `handle_arguments`

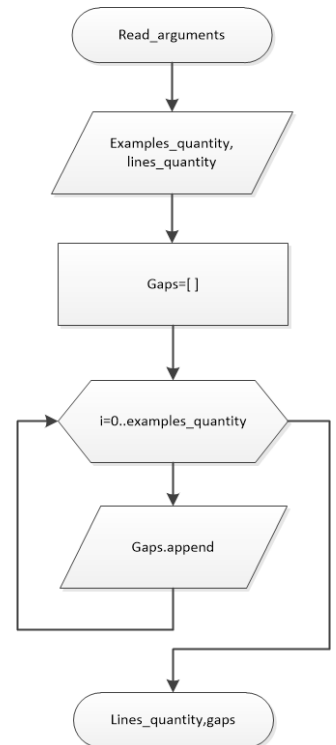
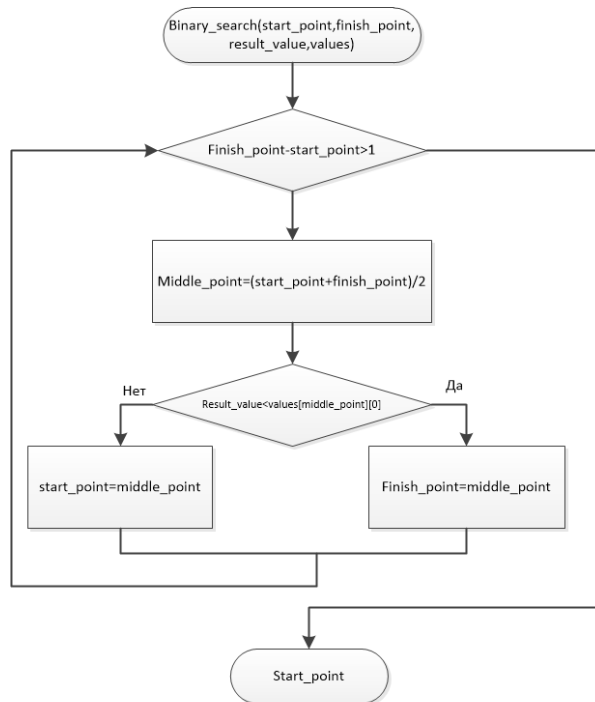
Назначение: вывод данных

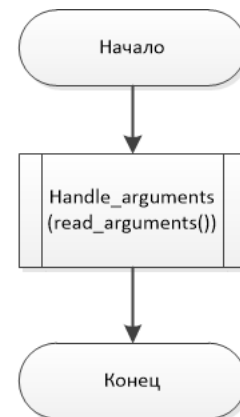
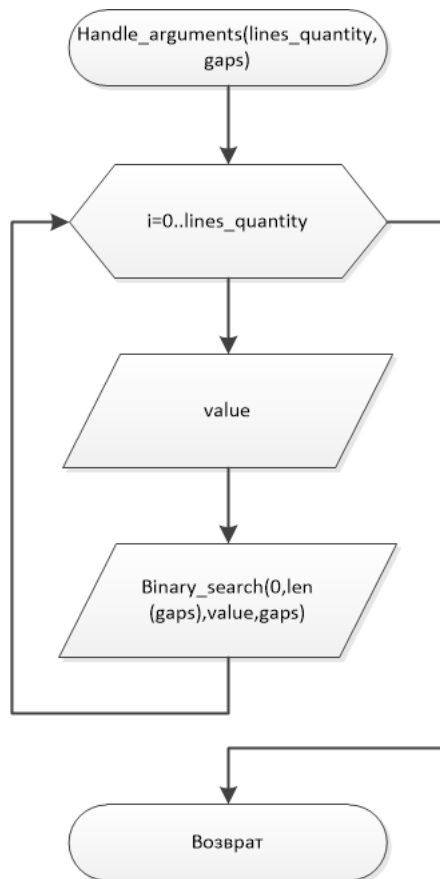
Входные данные: количество, промежуток

Возвращаемые данные: нет

### III. Блок-схемы

#### 1. Подпрограмм и основной программы





#### IV. Код программы

```

def binary_search(start_point, finish_point, result_value, values):
    while finish_point - start_point > 1:

        # Получаем медианное значение индекса
        middle_point = (start_point + finish_point) // 2

        if result_value < values[middle_point][0]:
            finish_point = middle_point
        else:
            start_point = middle_point

    return start_point

def read_arguments():
    # Вводим первую строку, разбиваем её на массив строк через пробел и переводим всё
    # в числа
    examples_quantity, lines_quantity = list(map(int, input().split()))
    gaps = []

    # Вводим последующие a строк
    for i in range(examples_quantity):
        gaps.append(list(map(int, input().split())))

    return lines_quantity, gaps

def handle_arguments(lines_quantity, gaps):

```



```
for i in range(lines_quantity):  
    value = int(input())  
  
    print(binary_search(0, len(gaps), value, gaps))
```

*# Код основной программы*

*# Сначала выполняется read\_arguments, после чего передаёт возвращаемые значения в handle\_arguments. Звёздочка (\*) нужна для того, чтобы аргументы передавались не в виде списка, а как отдельные аргументы.*

```
handle_arguments(*read_arguments())
```

### Контрольные вопросы

1. Приведите пример описания функции.
2. Что нужно для вызова подпрограммы?
3. Какие способы передачи аргументов вы знаете?
4. В чём различия способов передачи аргументов в подпрограмму?
5. Опишите механизм работы подпрограммы.
6. Какие типы функций вы знаете?
7. Что такое область видимости?
8. Что такое рекурсия?
9. Какие бывают рекурсии?
10. Что такое побочный эффект?

### ЛАБОРАТОРНАЯ РАБОТА №5

**Двумерные массивы. Использование подпрограмм при работе с двумерными массивами**

Цель работы: Приобрести практические навыки работы с двумерными массивами.

#### Содержание работы

1. Разработать алгоритм, описать его в виде блок-схемы и составить программу для решения задачи соответствующего варианта.
2. Набрать текст программы и отладить её в среде разработки PyCharm.
3. Оформить отчет о проделанной работе. Отчет должен включать в себя номер варианта, условие задачи, блок-схему и код программы.

#### Варианты заданий

1. Выполнить обработку элементов прямоугольной матрицы **a**, имеющей **n** строк и **m** столбцов. Найти наибольший элемент столбца матрицы **a**, для которого сумма абсолютных значений элементов максимальна.
2. Выполнить обработку элементов прямоугольной матрицы **a**, имеющей **n** строк и **m** столбцов. Найти наибольшее значение среди средних значений для каждой строки матрицы.
3. Выполнить обработку элементов прямоугольной матрицы **a**, имеющей **n** строк и **m** столбцов. Найти наименьший элемент столбца матрицы **a**, для которого сумма абсолютных значений элементов максимальна.
4. Выполнить обработку элементов прямоугольной матрицы **a**, имеющей **n** строк и **m** столбцов. Найти наименьшее значение среди средних значений для каждой строки матрицы.
5. Выполнить обработку элементов прямоугольной матрицы **a**, имеющей **n** строк и **m** столбцов. Определить средние значения по всем строкам и столбцам матрицы. Результат оформить в виде матрицы из **n+1** строк и **m+1** столбцов.
6. Выполнить обработку элементов прямоугольной матрицы **a**, имеющей **n** строк и **m** столбцов. Найти сумму элементов всей матрицы. Определить, какую долю в этой сумме составляет сумма элементов каждого столбца. Результат оформить в виде матрицы из **n+1** строк и **m** столбцов.

7. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $n$  строк и  $m$  столбцов. Найти сумму элементов всей матрицы. Определить, какую долю в этой сумме составляет сумма элементов каждой строки. Результат оформить в виде матрицы из  $n$  строк и  $m+1$  столбцов.
8. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $n$  строк и  $m$  столбцов. Определить, сколько отрицательных элементов содержится в каждом столбце и в каждой строке матрицы. Результат оформить в виде матрицы из  $n+1$  строк и  $m+1$  столбцов.
9. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $n$  строк и  $m$  столбцов. Определить, сколько нулевых элементов содержится в верхних  $d$  строках матрицы и в левых  $k$  столбцах матрицы.
10. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Перемножить элементы каждого столбца матрицы с соответствующими элементами  $K$ -го столбца.
11. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Просуммировать элементы каждой строки матрицы с соответствующими элементами  $L$ -й строки.
12. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Разделить элементы каждой строки на элемент этой строки с наибольшим значением.
13. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Разделить элементы каждого столбца матрицы на элемент этого столбца с наибольшим значением.
14. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Разделить элементы матрицы на элемент матрицы с наибольшим значением.
15. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Все элементы имеют целый тип. Дано целое число  $H$ . Определить, какие столбцы имеют хотя бы одно такое число, а какие не имеют.
16. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Исключить из матрицы строку с номером  $L$ . Сомкнуть строки матрицы.
17. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Добавить к матрице строку и вставить ее под номером  $L$ .
18. Выполнить обработку элементов квадратной матрицы  $a$ , имеющей  $N$  строк и  $N$  столбцов. Найти сумму элементов, стоящих на главной диагонали, и сумму элементов, стоящих на побочной диагонали (элементы главной диагонали имеют индексы от  $[0, 0]$  до  $[N, N]$ , а элементы побочной диагонали – от  $[N, 0]$  до  $[0, N]$ ).
19. Выполнить обработку элементов квадратной матрицы  $a$ , имеющей  $N$  строк и  $N$  столбцов. Определить сумму элементов, расположенных параллельно главной диагонали (ближайшие к главной). Элементы главной диагонали имеют индексы от  $[0, 0]$  до  $[N, N]$ .

20. Выполнить обработку элементов квадратной матрицы  $a$ , имеющей  $N$  строк и  $N$  столбцов. Определить произведение элементов, расположенных параллельно побочной диагонали (ближайшие к побочной). Элементы побочной диагонали имеют индексы от  $[N, 0]$  до  $[0, N]$ .
21. Выполнить обработку элементов квадратной матрицы  $a$ , имеющей  $N$  строк и  $N$  столбцов. Каждой паре элементов, симметричных относительно главной диагонали (ближайшие к главной), присвоить значения, равные полусумме этих симметричных значений (элементы главной диагонали имеют индексы от  $[0, 0]$  до  $[N, N]$ ).
22. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Исходная матрица состоит из нулей и единиц. Добавить к матрице еще один столбец, каждый элемент которого делает количество единиц в каждой строке чётным.
23. Выполнить обработку элементов квадратной матрицы  $a$ , имеющей  $N$  строк и  $N$  столбцов. Найти сумму элементов, расположенных выше главной диагонали, и произведение элементов, расположенных выше побочной диагонали (элементы главной диагонали имеют индексы от  $[0, 0]$  до  $[N, N]$ , а элементы побочной диагонали — от  $[N, 0]$  до  $[0, N]$ ).
24. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Дан номер строки  $L$  и номер столбца  $K$ , при помощи которых исходная матрица разбивается на четыре части. Найти сумму элементов каждой части.
25. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Определить, сколько нулевых элементов содержится в каждом столбце и в каждой строке матрицы. Результат оформить в виде матрицы из  $N + 1$  строк и  $m + 1$  столбцов.
26. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Дан номер строки  $L$  и номер столбца  $K$ , при помощи которых исходная матрица разбивается на четыре части. Найти среднее арифметическое элементов каждой части.
27. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Все элементы имеют целый тип. Дано целое число  $H$ . Определить, какие строки имеют хотя бы одно такое число, а какие не имеют.
28. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Исключить из матрицы столбец с номером  $K$ . Сомкнуть столбцы матрицы.
29. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Добавить к матрице столбец чисел и вставить его под номером  $K$ .
30. Выполнить обработку элементов прямоугольной матрицы  $a$ , имеющей  $N$  строк и  $m$  столбцов. Добавить к элементам каждого столбца такой новый элемент, чтобы сумма положительных элементов стала бы равна модулю суммы отрицательных элементов. Результат оформить в виде матрицы из  $N + 1$  строк и  $m$  столбцов.

Пример выполнения

Условие задачи

Выполнить обработку квадратной матрицы (NxN), удалив из неё все строки, имеющие отрицательное произведение среднего и медианы.

#### Код программы

```
import numpy as np

def read_inf():
    n = int(input())

    # Создаём пустую матрицу NxN
    matrix = np.empty((n, n))

    for i in range(n):
        matrix[i] = np.array(input().split(), float)

    return n, matrix

def handle_matrix(n, matrix):
    for i in range(n - 1, -1, -1):
        # Проверяем произведение медианы и среднего значения строки на
        # отрицательность
        if np.mean(matrix[i]) * matrix[i][(n - 1) // 2] < 0:
            # Удаляем строку
            matrix = np.concatenate([matrix[:i], matrix[i + 1:]])

    return matrix

def print_matrix(n, matrix):
    for i in range(len(matrix)):
        for j in range(n):
            print(matrix[i][j], end=" ")
        print("\n")

n, matrix = read_inf()
matrix = handle_matrix(n, matrix)
print_matrix(n, matrix)
```

#### Блок-схема

##### Процедура read\_inf

Назначение: инициализация матрицы

Входные данные: нет

Возвращаемые данные: размерность матрицы, матрица

##### Процедура handle\_matrix

Назначение: удаление строк матрицы подходящих условию

Входные данные: размерность матрицы, матрица

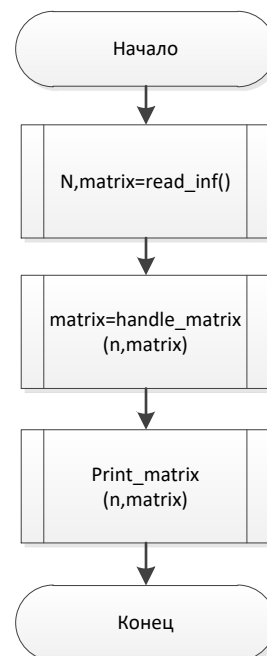
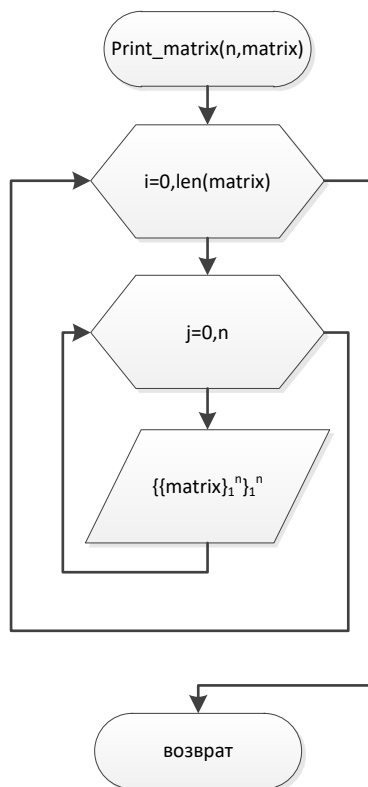
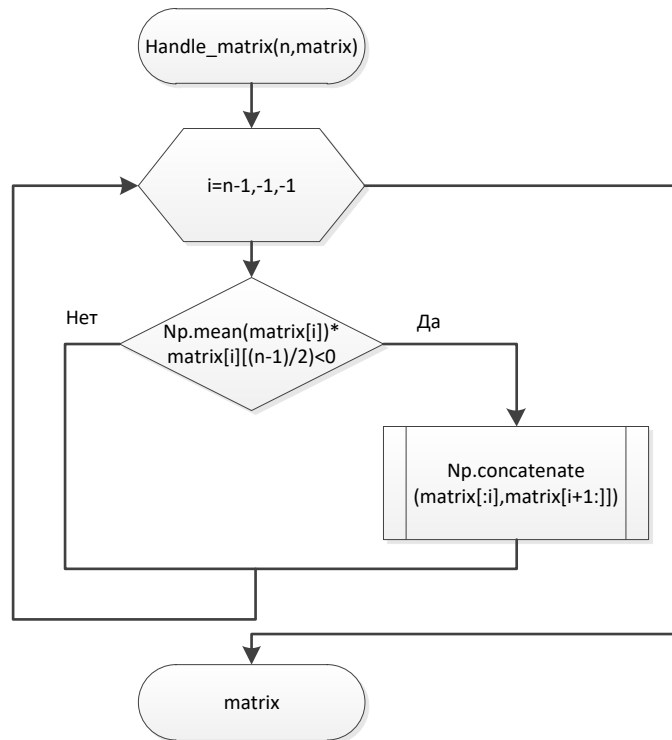
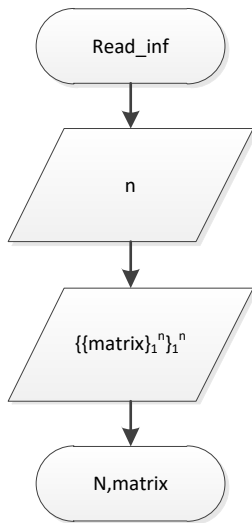
Возвращаемые данные: матрица

##### Процедура print\_matrix

Назначение: вывод матрицы

Входные данные: размерность матрицы, матрица

Возвращаемые данные: нет



## Контрольные вопросы

1. Что такое numPy?
2. Какие способы создания многомерных массивов вы знаете?
3. Как располагаются в памяти ЭВМ элементы многомерных массивов?
4. Приведите пример функции, генерирующей массив размера NxN.
5. Как вывести многомерный массив?
6. Как создать массив случайных чисел?
7. Перечислите операции, которые можно производить над массивами.

## ЛАБОРАТОРНАЯ РАБОТА №6

### Словари. Множества

Цель работы: получить навыки работы со словарями.

#### Варианты заданий

1. Дана строка, содержащая слова, разделённые пробелом. Вывести все слова, повторяющиеся более 3 раз.
2. В первой строке вводится число n. Далее следует n строк с кодом одного из трёх языков программирования: Pascal, C++, Python. Необходимо вывести название языка, код которого был введён.
3. В первых n строчках дан список стран и городов каждой страны. Затем, в следующих m строчках, даны произвольные названия городов. Для каждого города укажите, в какой стране он находится.
4. Дана последовательность строк, оканчивающаяся символом 0. Каждая строка содержит никнейм игрока и его счёт. Необходимо после ввода 0 вывести топ 3 игроков по сумме их счетов за всё время.
5. На вход программы вводится обучающий блок, состоящий из чисел, написанных прописью. Он включает числа 1, 2, ..., 19, 20, 30, 40, ..., 90. Далее вводится n чисел, каждое из которых не более 99. Необходимо вывести каждое введённое число прописью.
6. В консоль вводится n строк. Каждая строка содержит название сообщества и города, входящие в него. Необходимо вывести все города, входящие сразу во все сообщества.
7. Дана строка, содержащая слова, разделённые пробелом. Вывести уникальные слова в данной строке.
8. В первой строке вводится число n. Далее следует n строк с кодом одного из трёх форматов: HTML, XML, JSON. Необходимо вывести название языка, код которого был введён.
9. В первых n строчках дан список стран и озёр каждой страны. Затем, в следующих m строчках, даны произвольные названия озёр. Для каждого озера укажите, в какой стране он находится.
10. Дана последовательность строк, оканчивающаяся пробелом. Каждая строка содержит имя человека и сумму его счета для оплаты. Необходимо после ввода 0 вывести имя человека с самой большой суммой счетов.
11. В консоль вводится n строк. Каждая строка содержит название трофея и никнеймы всех игроков, которые владеют данным трофеем. Необходимо вывести никнейм игрока, владеющего всеми трофеями, кроме первого.
12. Дана строка, содержащая слова, разделённые пробелом. Вывести все слова, ровно 2 раза.

13. В первой строке вводится число  $n$ . Далее следует  $n$  строк с кодом одного из трёх языков программирования: JavaScript, PHP, Lua. Необходимо вывести название языка, код которого был введён.
14. В первых  $n$  строчках дан список отелей и их посетителей. Затем, в следующих  $m$  строчках, произвольно введены имена посетителей. Для каждого посетителя укажите, в каком отеле он проживает.
15. Дана последовательность строк, оканчивающаяся пробелом. Каждая строка содержит название города, его годовую прибыль и годовые расходы, а также номер года. Необходимо после ввода 0 вывести город с самыми большими убытками (расходы – доходы) за всё время, а также вывести самый убыточный год для этого города.
16. В консоль вводится  $n$  строк. Каждая строка содержит имя человека и лицензии, которые у него есть. Необходимо вывести лицензии, которыми владеет только один человек из выборки.
17. Дана не пустая последовательность слов из строчных латинских букв; между соседними словами – запятая, за последним словом – точка. Напечатать в алфавитном порядке все гласные буквы, которые входят в каждое слово.
18. Дано слово  $w$  и строка  $s$ . Вывести все гласные буквы, которые одновременно встречаются и в строке и в заданном слове.
19. С клавиатуры вводятся натуральные числа. Признак конца ввода – 0. Сформировать множество четных цифр, входящих в запись введенных чисел только один раз.
20. Даны два множества  $x_1$  и  $x_2$ , содержащие элементы типа *Byte*. Сформировать новое множество  $y$ , равное разности заданных множеств и выделить из него подмножество  $y_1$ , которое содержит элементы множества  $y$ , являющиеся делителями числа 50.
21. Назовем два слова, у которых совпадают множества символов, эквивалентными. Вывести каждую группу эквивалентных слов данной строки.
22. Дана матрица  $a$ . Описать функцию  $sum(A, s_1, s_2)$ , вычисляющую сумму тех элементов матрицы  $a$ , номера строк и столбцов которых принадлежат соответственно непустым множествам  $s_1$  и  $s_2$  (множества задаются пользователем).
23. Дан массив из  $n$  натуральных чисел. Сформировать множество цифр, которых нет в записи хотя бы одного из чисел массива.
24. Сформировать множество целых чисел из диапазона 1..255, представимых в виде  $n^2 + m^2$ . Выведите на печать полученное множество.
25. Дана строка  $s$ . Сформировать множество символов: цифры, '+', '-', '\*', '\_', входящих в строку  $s$ , а также множество вхождений этих символов.
26. Дана последовательность целых неотрицательных чисел, не превышающих 255. Сформировать множество чисел Фибоначчи содержащихся в данной последовательности, если таковых нет, то вывести соответствующее сообщение.
27. Пусть дан текст из латинских строчных букв. Сформировать множество букв, входящие в текст по одному разу.
28. С клавиатуры вводятся натуральные числа  $\leq 255$ . Признак конца ввода – 0. Сформировать множество из чисел, в записи которых встречаются все цифры, имеющиеся в записи числа  $n$ , но отсутствующие в записи числа  $m$ .
29. Дана строка символов, признак конца строки – точка. Сформировать множество гласных букв, входящих в строку не менее двух раз.
30. С клавиатуры вводятся три натуральных числа, не превышающие 255. Получить множество общих делителей введенных чисел.



## Пример выполнения

### Условие задачи

В первой строке вводится число  $n$ . В следующих  $n$  строках вводятся множества слов, разделённых пробелом. Далее вводится число  $m$ . За ним вводится текст, состоящий из  $n$  символов. Необходимо определить процент принадлежности текста к каждому из множеств.

### Код программы

```
def read_sequences():
    n = int(input())
    words = []

    for i in range(n):
        words.append(set(input().split()))

    return words

def read_text():
    m = int(input())
    text = []

    for i in range(m):
        text += input().split()

    return text

def handle_text(sets, text):
    similarity = [0 for i in range(len(sets))]

    for i in range(len(sets)):
        for word in text:
            if word in sets[i]:
                similarity[i] += 1

    for i in range(len(similarity)):
        similarity[i] /= len(text)

    return similarity

# Тело программы
sets = read_sequences()
text = read_text()

similarity = handle_text(sets, text)

print(*similarity, sep="\n")
# В данном случае звездочка (*) используется для передачи элементов списка similarity
в виде отдельных аргументов функции print
```

### Блок-схема

#### Процедура read\_sequences

Назначение: считывание последовательности

Входные данные: нет

Возвращаемые данные: последовательность

### Процедура read\_text

Назначение: считывание текста

Входные данные: нет

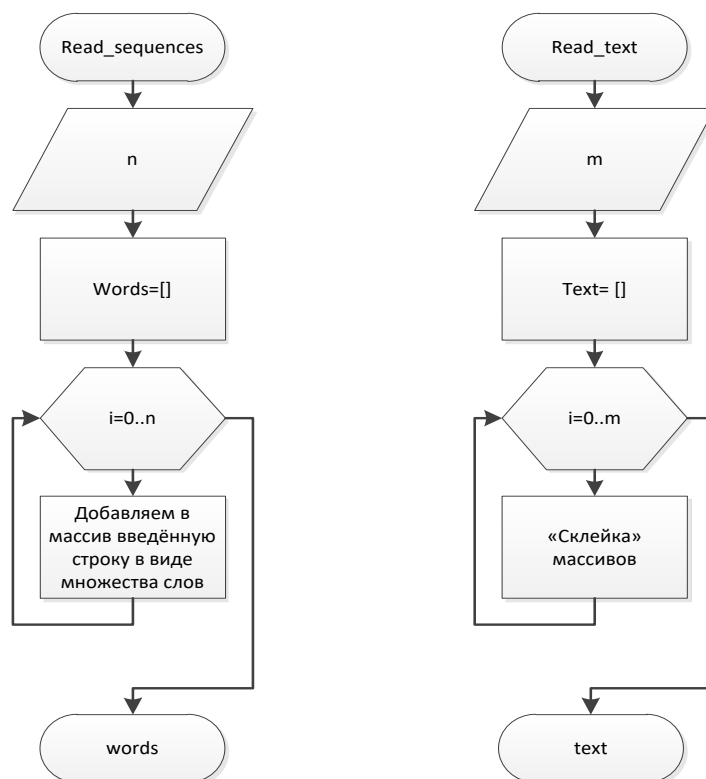
Возвращаемые данные: текст

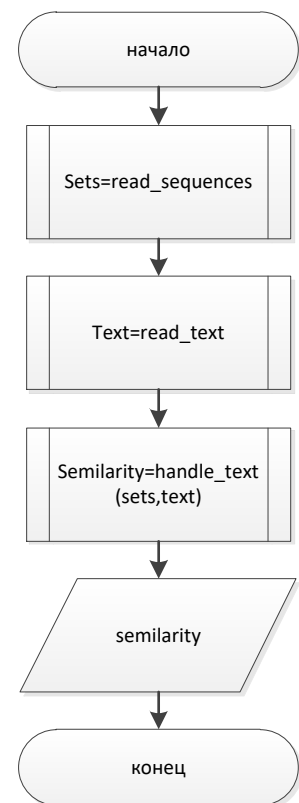
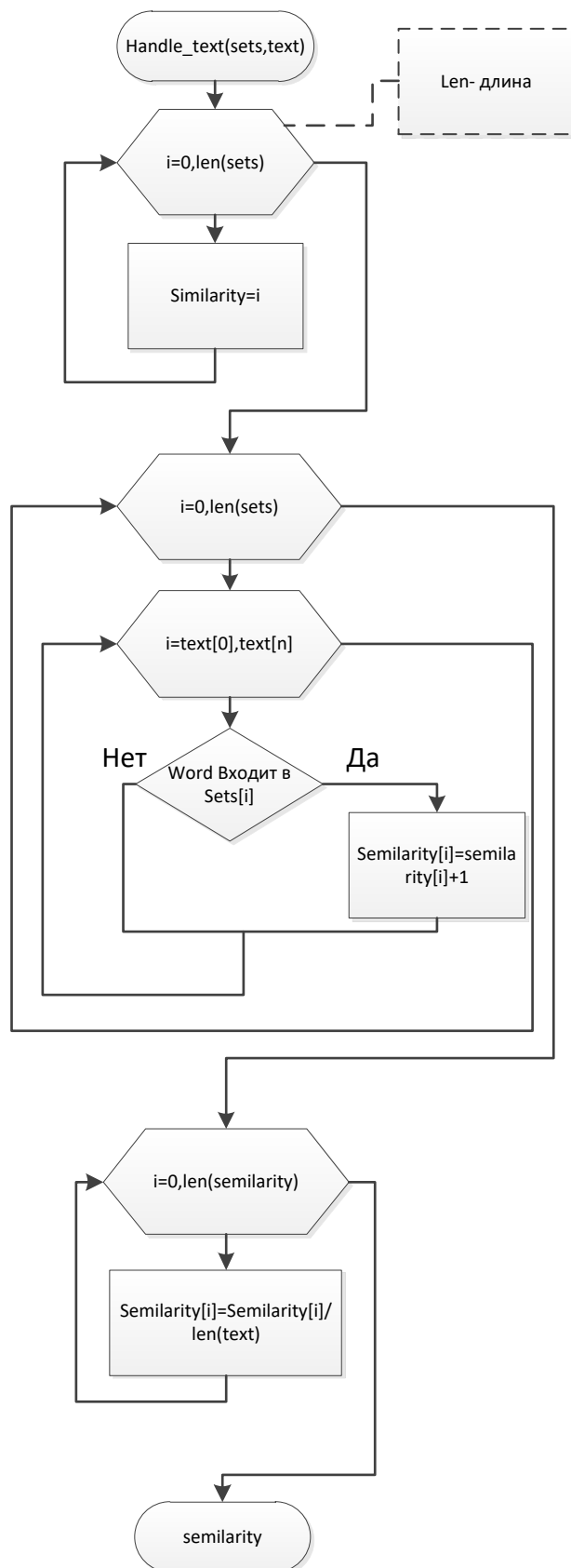
### Процедура handle\_text

Назначение: вычисление процента принадлежности

Входные данные: множество, текст

Возвращаемые данные: нет





### Контрольные вопросы

1. Что такое множество?
2. Как описать в Python переменную множественного типа?
3. Что такое словарь?

4. Как в Python описать словарь?
5. Какое множество называется пустым, какое эквивалентным?
6. Что называется мощностью множества?
7. Перечислите основные операции над множествами.

## ЛАБОРАТОРНАЯ РАБОТА №7

### Работа с файлами.

Цель работы: получить навыки работы с файлами.

#### Варианты заданий

1. Дан текстовый файл. Удалить из него однобуквенные слова и лишние пробелы, вывести все буквы данного текстового файла, входящие в файл не менее двух раз на монитор.
2. Дан текстовый файл и строка  $s$ . Если самая длинная строка файла не совпадает со строкой  $s$ , то строку  $s$  дописать в файл.
3. Дан текстовый файл, представляющий собой запись арифметического выражения, операндами которого являются однозначные числа. Число операций в выражении не больше двух. Вычислить значение этого выражения и дописать его в конец этого файла.
4. В текстовом файле хранятся квадратные целые матрицы порядка  $n$ . Преобразовать файл, удалив из каждой матрицы последнюю строку и последний столбец.
5. В файле хранятся строки. Найти строку, содержащую наименьшее количество символов и создать новый файл, который будет начинаться с этой строки и ею заканчиваться. Между началом и концом нового файла записать строки из данного файла в обратном порядке. Количество строк не превышает десяти.
6. Целочисленная матрица, в которой число строк  $n$  вдвое меньше числа столбцов, хранится в текстовом файле. Сохранить в этом файле матрицу порядка  $[n \times n]$ , удалив последние  $n$  столбцов.
7. В файле хранятся строки. Преобразовать строку, заменяя каждую цифру соответствующим ей числом пробелов.
8. Дан файл со строками (их количество неизвестно). Если в 3 или в 5 строке 4 элемент пробел, то скопировать этот файл в другой, заменив эти символы нулем. Если это условие не выполняется, то удалить эти строки, создав новый файл.
9. В текстовом файле записаны вещественные числа в форме с фиксированной точкой. Преобразовать файл, представив каждое число в форме с плавающей точкой, сохранив две цифры после точки в мантиссе.
10. В текстовом файле хранятся квадратные целочисленные матрицы порядка  $n$ . Преобразовать файл, удалив из каждой матрицы элементы побочной диагонали.
11. Даны два файла. Вывести начальные буквы строк, которые есть в каждом из данных файлов, в новый файл, причем буквы должны чередоваться.
12. Дан текстовый файл  $f$ . Записать в файл  $g$  символы файла  $f$  в обратном порядке, оставив только первые вхождения каждого символа.
13. Дан файл. Удалить из него однобуквенные слова и лишние пробелы.
14. Преобразовать файл, удалив последнее слово в каждой строке.
15. Дан файл, содержащий неизвестное количество строк. Найти самое длинное слово среди слов, начинающихся буквой «а». Если таких слов нет, сообщить об этом. Слова, начинающиеся буквой «а», записать в другой файл.

16. Даны файлы f и g. Записать в файл h все совпадающие строки файлов f и g, если таких строк нет, то в файл h записать строки из файлов f и g, чередуя их, начиная с первой из файла f.
17. Дан файл. Преобразовать его, удалив из каждой строки слова, встречающиеся более двух раз, если таких строк нет, то вывести об этом сообщение и скопировать этот файл в другой, дописав в его конец выведенное сообщение.
18. Дан файл. Преобразовать его, оставив в каждой строке только самое длинное слово
19. Дан файл, содержащий строки целых чисел. Получить файл, содержащий последние числа каждой строки, если число меньше заданного n, то заменить его нулем.
20. Даны файлы f и g, содержащие строки. Дописать в файл f строки из файла g, которых нет в файле f. Файл g уничтожить.
21. Удалить слова из данного файла, длина которых превышает данное число n, если таких строк нет, то заменить третью строку строкой, введенной пользователем.
22. Дан файл, содержащий строки. Если третья строка не является самой длинной или самой короткой, то скопировать в новый файл строки из данного, дописав в конец каждой, первые три символа последующей.
23. В файле, содержащем строки, содержатся русские имена. Проверить, все ли они начинаются с прописной буквы, если нет, то исправить эту ошибку. Записать полученные данные в новый файл.
24. В файле содержится текст, представляющий собой последовательность строк. Записать в новый файл заданный текст в порядке возрастания длин строк. Если есть строки одинаковой длины, то записать их в обратном порядке.
25. Файл содержит текст из латинских строчных букв. Вывести в новый файл все буквы, входящие в текст по одному разу и дописать в конец файла строки, длина которых не превышает n.
26. Дан файл, содержащий строки. Выясните, состоят ли любые две его строки из множества одинаковых символов. Если можно, то записать в новый файл эти строки, если нет, то выдать сообщение о том, что таких строк нет и записать это сообщение в новый файл.
27. Дан файл, содержащий строки. Удалить из строк все символы, не являющиеся буквами, а прописные буквы заменить строчными, и записать в новый файл строки, начиная с конца.
28. Дан файл, содержащий строки. Записать в новый файл строки, изменив порядок следования символов в каждом слове строки на обратный.
29. Дан файл, содержащий матрицу символов (количество строк равно их длинам). Если в главной диагонали этой матрицы содержатся пробелы, то в новый файл записать транспонированную матрицу данной матрицы. Если пробелов не содержится, то в новый файл скопировать эту матрицу, поменяв местами первую и предпоследнюю строки местами.
30. Дан файл, содержащий строки. Если количество строк больше заданного n, то создать новый файл и записать в него строки из данного файла. Если количество строк меньше или равно n, то скопировать в новый файл строки по такой.

#### Пример выполнения

##### Условие задачи

Дан файл input.dat, содержащий слова, разделённые символом пробела. Вывести в файл output.dat все слова, не являющиеся палиндромами.

##### Код программы

```
input_file = "input.dat"  
output_file = "output.dat"
```

```

# Функция считывания данных из файла
def read_file(filename):
    input_stream = open(filename, "r")
    result_list = []

    # Последовательно считываем строки из файла
    for line in input_stream:
        line = line.strip().split(" ") # Создаём массив слов
        result_list += line # Склеиваем финальный массив и считанный массив слов

    return result_list

# Функция проверки слова на палиндром
def check_palindrome(word):
    palindrome = True # Иници. флаг

    # Проверяем на палиндром
    for i in range(len(word) // 2):
        if word[i] != word[len(word) - 1 - i]:
            palindrome = False
            break
    return palindrome

# Функция удаления лишних палиндромов
def handle_list(_list):
    result_list = []
    for word in _list:
        if not check_palindrome(word):
            result_list.append(word)
    return result_list

# Функция вывода в новый файл
def print_list(_list, filename):
    output_stream = open(filename, "w")
    print(*_list, file=output_stream)

# Тело программы
local_list = read_file(input_file)
final_list = handle_list(local_list)
print_list(final_list, output_file)

```

#### Блок-схема

##### Процедура read\_file

Назначение: считывание из файла

Входные данные: имя файла

Возвращаемые данные: список

##### Процедура check\_palindrome

Назначение: проверка слова на палиндромность

Входные данные: слово

Возвращаемые данные: является ли слово палиндромом

### Процедура handle\_list

Назначение: удаление палиндромов

Входные данные: список

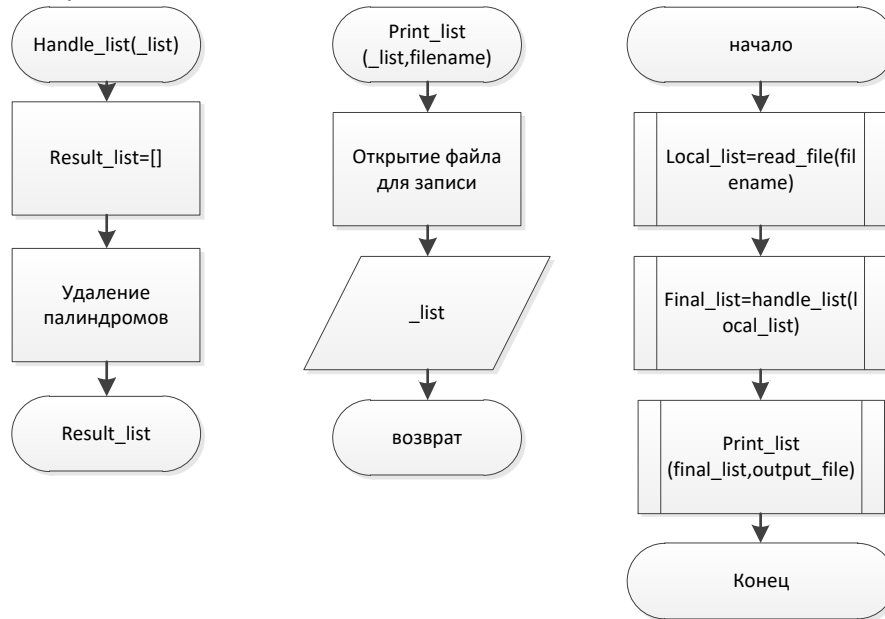
Возвращаемые данные: список

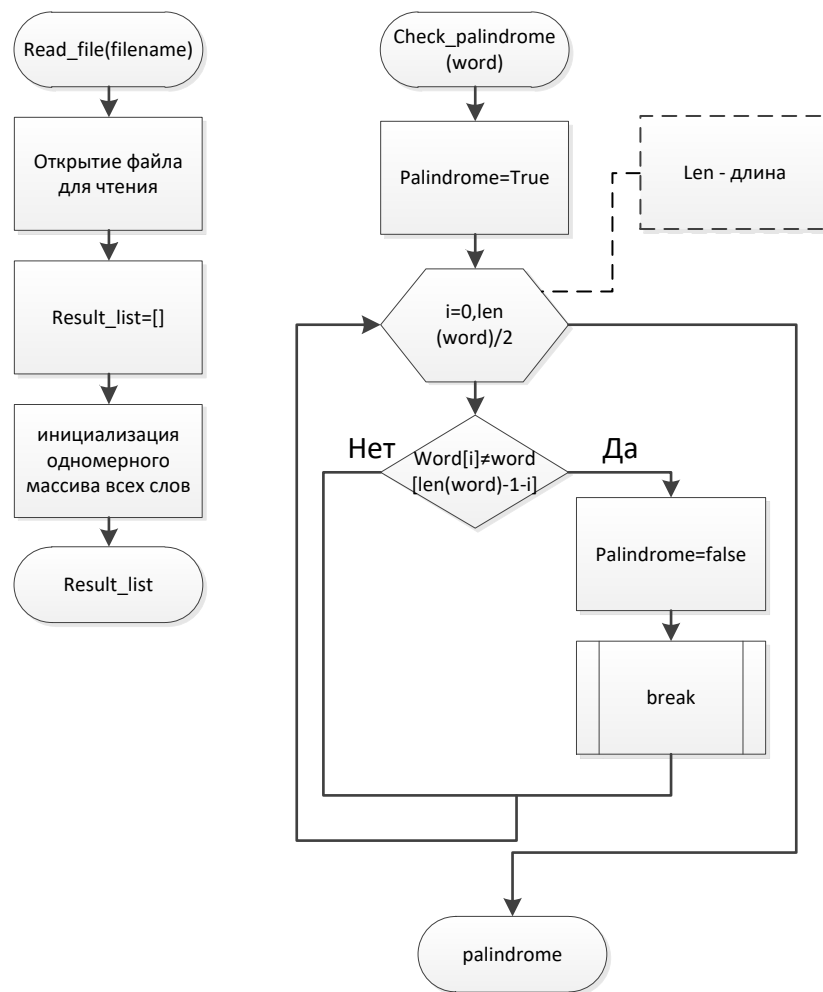
### Процедура print\_list

Назначение: запись в файл

Входные данные: список, имя файла

Возвращаемые данные: нет





#### Контрольные вопросы

1. Какие виды файлов существуют в Python?
2. Файловые типы языка Python. Их описание в программе.
3. Операции ввода-вывода при работе с файлами.
4. Стандартные процедуры, используемые для выполнения установочных и завершающих действий над файлами.



## ЛАБОРАТОРНАЯ РАБОТА №8

### Работа с графикой. Классы.

Цель работы: Приобрести навыки работы с графикой, классами.

#### Содержание работы

Создать класс, отрисовывающий графический интерфейс программы соответствующего варианта. Также необходимо создать основную программу, использующую данный класс.

#### Варианты заданий

1. Нарисовать флаг России, располагающийся в круге.
2. Вывести на экран фамилию преподавателя. Каждая буква должна иметь свой узор.
3. Нарисовать светофор.
4. Составить программу «Звездное небо»: в черном окне случайным образом появляются белые точки. Работа программы заканчивается по нажатию клавиши.
5. Написать универсальную процедуру построения графика функции  $y = F(x)$  точечным методом. Процедура должна иметь следующие Параметры:  $X_{min}$ ,  $X_{max}$ ,  $Y_{min}$ ,  $Y_{max}$ ,  $Xgmin$ ,  $Xgmax$ ,  $Ygmin$ ,  $Ygmax$ . Функция  $F(x)$  описывается во внешней подпрограмме-функции.
6. Исследовав область определения и выбрав расположение координатных осей, построить на экране графики функций используя процедуру построения графика функции  $y = F(x)$  точечным методом.

$$y = \frac{1}{x + 1}$$

7. Нарисовать флаг Германии, располагающийся в круге.
8. Нарисовать флаг Польши, располагающийся в круге.
9. Вывести на экран цифры по шаблону, заданному на почтовом конверте.
10. Нарисовать поздравление с совершеннолетием. В центре экрана должно находиться большое число 18, закрашенное красным цветом.
11. Составить программу, которая выводит на экран олимпийские кольца.
12. Исследовав область определения, построить график функции.

$$y = \frac{1}{x^3}$$

13. Построить на экране синусоиду. Не менее 10 раз изменить ее цвет.
14. Составить программу рисования на экране шахматного поля.
15. Создать неподвижный графический объект мяч, лежащий на какой-либо поверхности.
16. Нарисовать снеговика, раскрашенного голубым цветом.
17. Разработать программу перемещения «Дом» и внести такие изменения, в результате которых дом окажется раскрашенным в разные цвета.
18. Разработать программу перемещения «Самолет» и внести такие изменения, в результате которых самолет окажется раскрашенным в разные цвета.
19. Создать программу "Звездное небо". В случайных местах на черном фоне появляются белые точки. Когда количество точек достигает определенного значения, предыдущие исчезают.
20. Нарисовать елку с гирляндой, переливающейся огоньками.
21. Разработать программу перемещения «Машина» и внести такие изменения, в результате которых машина окажется раскрашенным в разные цвета.

22. Нарисовать флаг Финляндии, располагающийся в круге.
23. Исследовав область определения, построить график функции.
 
$$y = \frac{x}{x^3 - x^2 + 1}$$
24. Смоделировать движение мяча при падении.
25. Разработать программу перемещения фигуры «Облако» и создать неподвижные облака.
26. Нарисовать флаг Италии, располагающийся в круге.
27. Разработать программу перемещения «Робот» и внести такие изменения, в результате которых робот окажется раскрашенным в разные цвета.
28. Разработать программу передвижение шашки по шахматной доске.
29. Используя линии и другие графические примитивы, составить программу, рисующую дом, который перемещается.
30. Составить программу, которая выводит на экран компьютера ваши инициалы с помощью процедур рисования линий, окружностей и т.п. Каждая буква должна быть нарисована разным цветом.

#### Пример выполнения

##### Условие задачи

В консоль подаётся два числа a, b. В графическое окно вывести 100 случайных овалов.

##### Код программы

```
from Drawer import *
from random import *
import time

window = Drawer(600, 600)
colors = ['aqua', 'blue', 'fuchsia', 'green', 'maroon', 'orange',
          'pink', 'purple', 'red', 'yellow', 'violet', 'indigo',
          'chartreuse', 'lime']

for i in range(100):
    color = choice(colors)
    radius = randint(10, 300)
    window.draw_circle(randint(0, 300 - radius), randint(0, 600 - radius), radius,
                        color)

import tkinter

class Drawer:

    graph = tkinter.Tk()
    canvas = tkinter.Canvas(graph, width=100, height=100)

    def __init__(self, canvas_x, canvas_y):
        self.canvas = tkinter.Canvas(self.graph, width=canvas_x, height=canvas_y)
        self.canvas.pack()

    def draw_circle(self, x, y, radius, color):
        self.canvas.create_oval(x, y, x + 2 * radius, y + 2 * radius, fill=color)
        self.canvas.update()
```

## Блок-схема

### Процедура \_\_init\_\_

Назначение: инициализация объекта класса

Входные данные: координаты холста

Возвращаемые данные: нет

### Процедура draw\_circle

Назначение: рисование овалов

Входные данные: координаты центра, радиус, цвет

Возвращаемые данные: нет

