Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего

образования «Южно-Уральский государственный университет (национальный

исследовательский университет)»

Высшая школа экономики и управления

Кафедра Информационных технологий в экономике

Программирование на языке Python (курс молодого бойца)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

по дисциплине «ПРОГРАММИРОВАНИЕ»

ЮУрГУ – 380305.2022.341. ПЗ КР

Рецензент, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Руководитель, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г. «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.

Нормоконтролёр, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Автор, студент группы ЭУ-142

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Назаров К.А./\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.

Работа защищена с оценкой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2019г.

Челябинск 2019

# АННОТАЦИЯ

Назаров К.А.

Программирование на языке Python (курс молодого бойца)

Челябинск: ЮУрГУ, ЭУ-142, 2019

В данной работе раскрывается основные возможности языка программирования Python, библиотеки Numpy.

ОГЛАВЛЕНИЕ

[АННОТАЦИЯ 1](#_Toc11579900)

[ВВЕДЕНИЕ 3](#_Toc11579901)

[МЕТОДЫ СОРТИРОВКИ 4](#_Toc11579902)

[SELECT SORT 4](#_Toc11579903)

[BUBBLE SORT 4](#_Toc11579904)

[INSERT SORT 5](#_Toc11579905)

[ИСЛЕДОВАНИЕ АЛГОРИТМОВ 5](#_Toc11579906)

[БИБЛИОТЕКА NUMPY 8](#_Toc11579907)

[ЧТО ТАКОЕ NUMPY 8](#_Toc11579908)

[НАБИВАЕМ РУКУ 8](#_Toc11579909)

[РЕШЕНИЕ СЛАУ 34](#_Toc11579910)

[ЗАКЛЮЧЕНИЕ 37](#_Toc11579911)

[СПИСОК ЛИТЕРАТУРЫ 38](#_Toc11579912)

# ВВЕДЕНИЕ

Python — современный язык программирования, работающий на всех распространенных операционных системах для настольных компьютеров. Язык программирования Питон разрабатывается чуть более 20 лет. В настоящее время активно используется две версии языка — более старая версия 2 и современная версия. Версия 2 более не развивается, но до сих пор еще используется, поскольку очень много программного обеспечения и библиотек разработано именно для версии 2. Между версиями есть существенная несовместимость, в том числе в синтаксисе команд ввода-вывода (программа на языке Python 2-й версии может не работать в 3-й версии и наоборот), но в целом они очень похожи. Мы будет использовать именно версию 3, как более современную и совершенную.

Python — современный универсальный интерпретируемый язык программирования. Его достоинства:

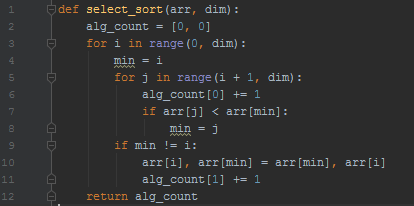
* Кроссплатформенность и бесплатность.
* Простой синтаксис и богатые возможности позволяют записывать программы очень кратко, но в то же время понятно.
* По простоте освоения язык сравним с бейсиком, но куда более богат возможностями и значительно более современен.
* Богатая стандартная библиотека, возможность разработки промышленных приложений (для работы с сетью, GUI, базами данных и т.д.)

# МЕТОДЫ СОРТИРОВКИ

## SELECT SORT

Сортировка выбором — здесь, чтобы отсортировать массив, находим элемент с минимальным значением, затем сравниваем его со значением первой неотсортированной позиции. Если этот элемент меньше, то он становится новым минимумом и их позиции меняются.

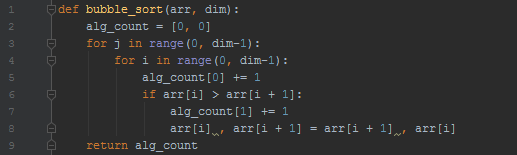
Пример реализации:



## BUBBLE SORT

Суть алгоритма в том, что совершается несколько проходов по массиву. При проходе последовательно сравниваются пары элементов в массиве и в случае несоответствия выбранному порядку меняются местами. Если пары элементов находятся в верном порядке, то ничего не происходит. В результате первого прохода максимальный элемент окажется в конце, то есть всплывет словно пузырек. Затем все повторяется до того момента пока весь массив не будет отсортирован.Последний проход будет по отсортированному массиву.

Пример реализации:

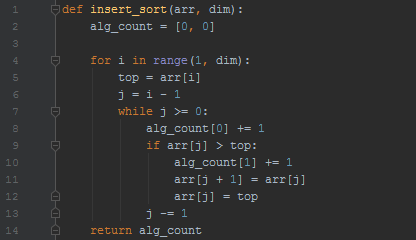


## INSERT SORT

Алгоритм примерно такой:

* из массива последовательно берется каждый элемент
* вставляется в его отсортированную часть(например в начале массива)

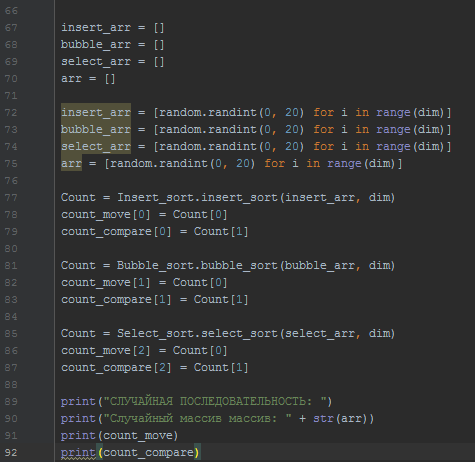
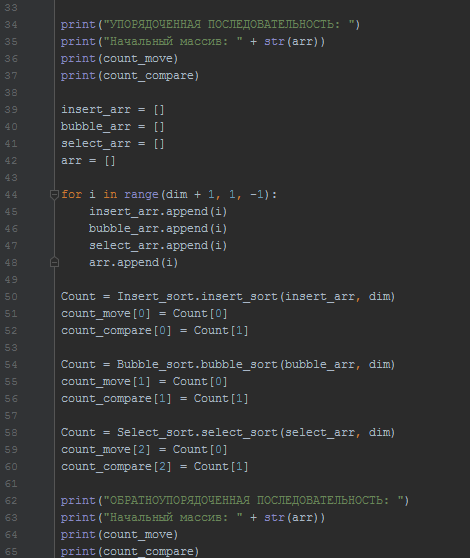
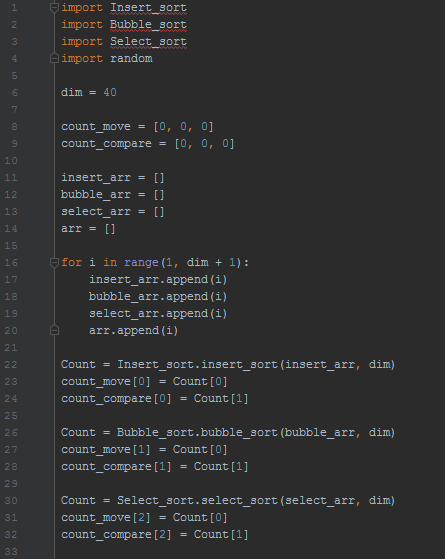
Пример реализации:



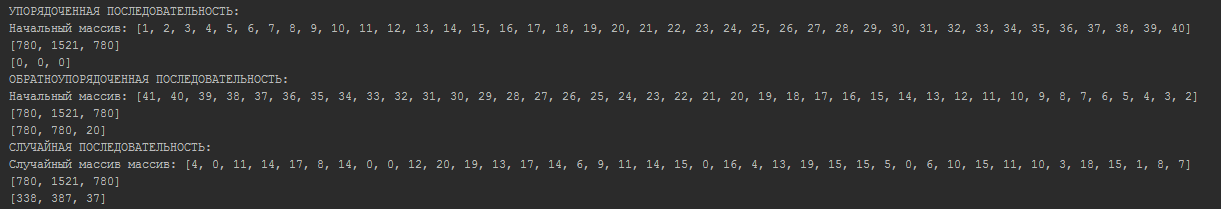
## ИСЛЕДОВАНИЕ АЛГОРИТМОВ

В данной части курсовой работы была написана программа, которая определяет эффективность трех алгоритмов сортировки (Сортировка включением, Обменная сортировка, Сортировка выбором). Каждый алгоритм сортировки оформлен в виде функции, которые подключены к основной программе. Чем меньше количество перемещений и сравнений, тем эффективней алгоритм

Основная программа:



Результат:



Исходя из проделанной работы можно сделать вывод. Самым эффективным методом сортировки для массива является метод "Select"

# БИБЛИОТЕКА NUMPY

## ЧТО ТАКОЕ NUMPY

NumPy - это фундаментальный пакет для научных вычислений на Python. Он содержит:

* мощный N-мерный массивы
* сложные (вещательные) функции
* инструменты для интеграции C / C ++ и кода Fortran
* полезная линейная алгебра, преобразование Фурье и возможности случайных чисел

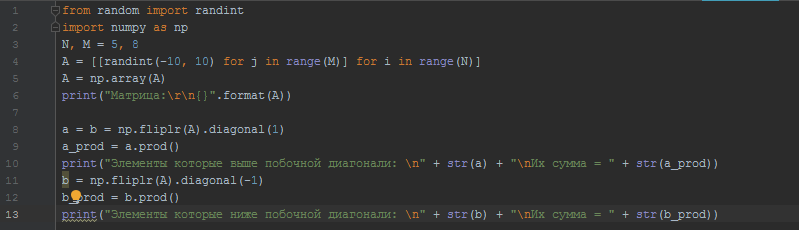
Помимо очевидного научного использования, NumPy также может использоваться как эффективный многомерный контейнер общих данных. Произвольные типы данных могут быть определены. Это позволяет NumPy легко и быстро интегрироваться с широким спектром баз данных.

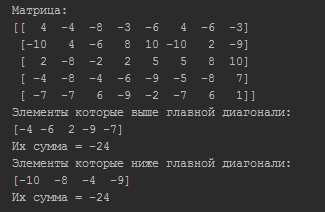
NumPy лицензируется по лицензии BSD, что позволяет использовать ее с небольшими ограничениями.

## НАБИВАЕМ РУКУ

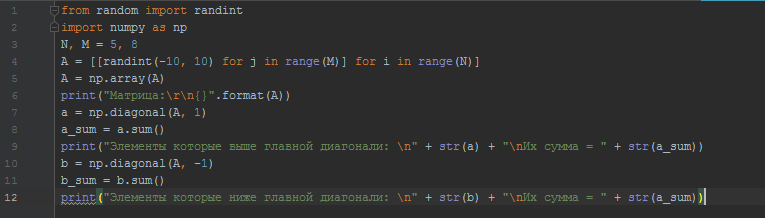
Для набивания руки нам представлены 31 задача, которые должны быть решены при помощи языка программирования Python и библиотеки NumPy.

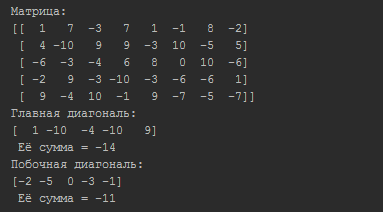
1. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Найти наибольший элемент столбца матрицы A, для которого сумма абсолютных значений элементов максимальна.



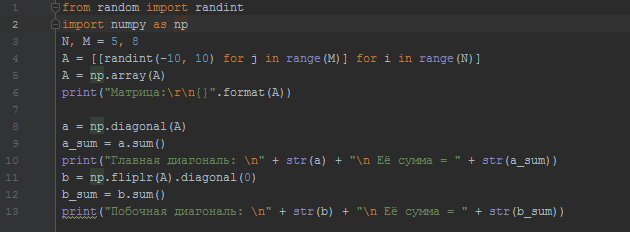


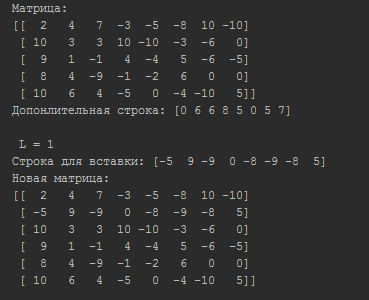
2. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Найти наибольшее значение среди средних значений для каждой строки матрицы.



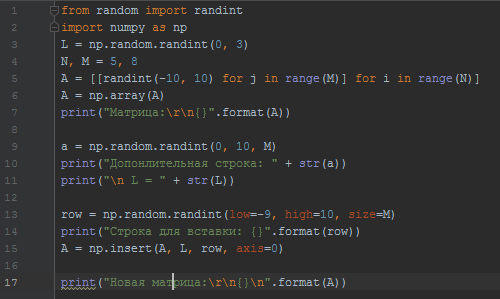


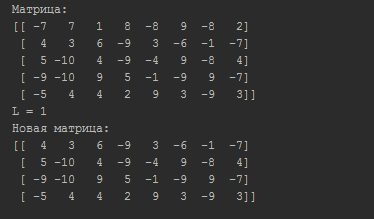
3. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Найти наименьший элемент столбца матрицы A, для которого сумма абсолютных значений элементов максимальна.



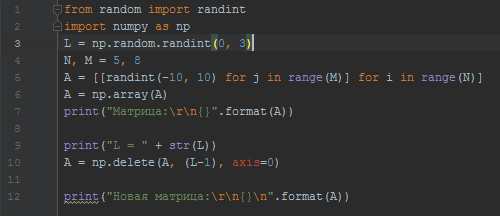


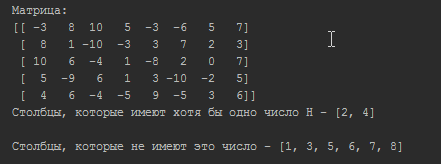
4. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Найти наименьшее значение среди средних значений для каждой строки матрицы.



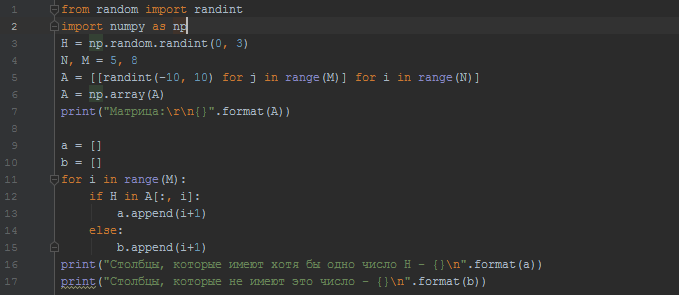


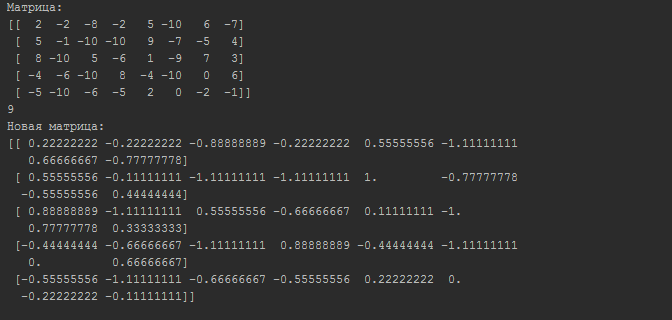
5. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Определить средние значения по всем строкам и столбцам матрицы. Результат оформить в виде матрицы из N + 1 строк и M + 1 столбцов.



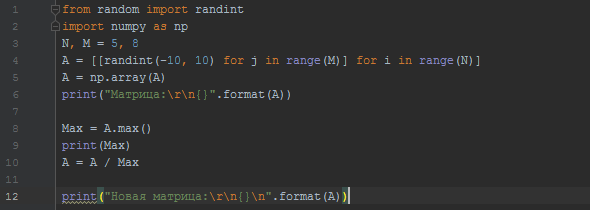


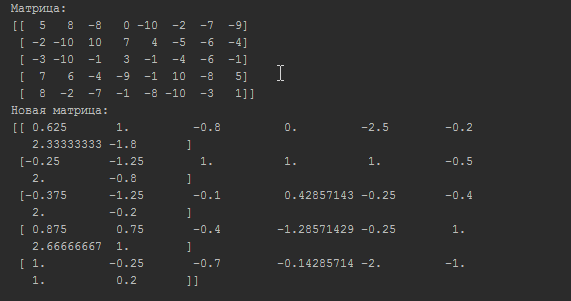
6. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Найти сумму элементов всей матрицы. Определить, какую долю в этой сумме составляет сумма элементов каждого столбца. Результат оформить в виде матрицы из N + 1 строк и M столбцов.



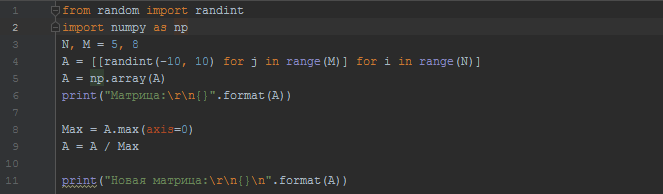


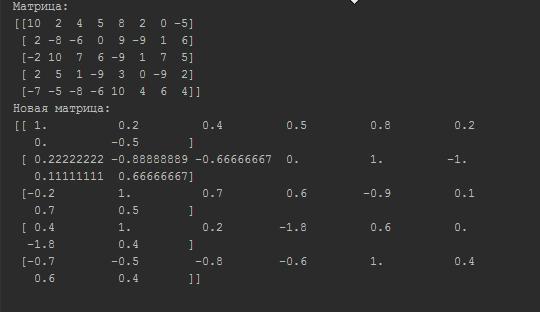
7. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Найти сумму элементов всей матрицы. Определить, какую долю в этой сумме составляет сумма элементов каждой строки. Результат оформить в виде матрицы из N строк и M+1 столбцов.



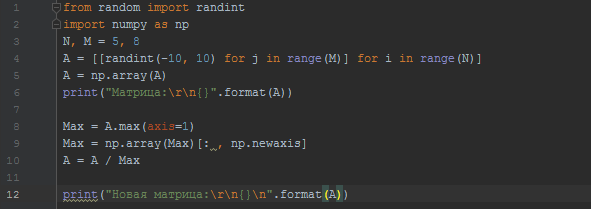


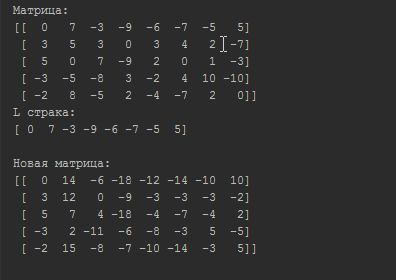
8. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Определить, сколько отрицательных элементов содержится в каждом столбце и в каждой строке матрицы. Результат оформить в виде матрицы из N + 1 строк и M + 1 столбцов.



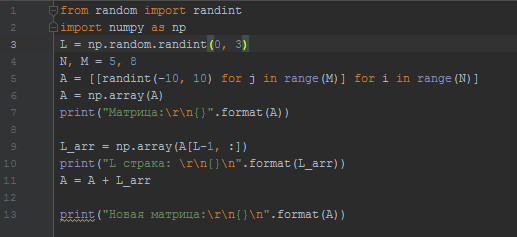


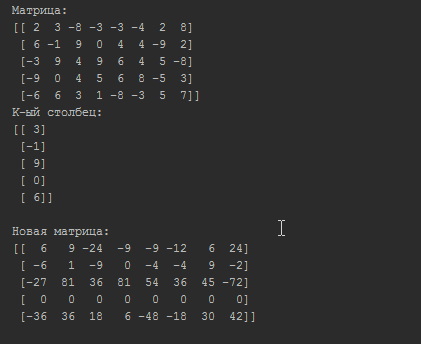
9. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Определить, сколько нулевых элементов содержится в верхних L строках матрицы и в левых К столбцах матрицы.



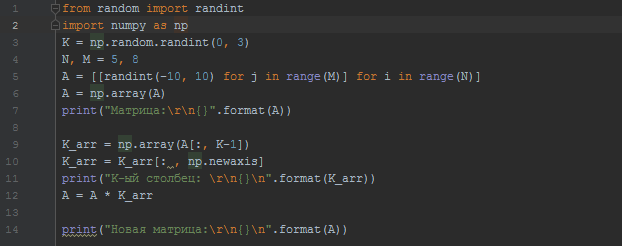


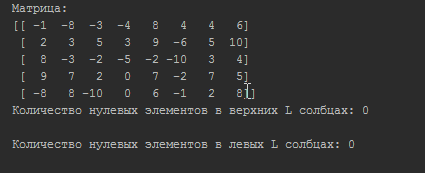
10. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Перемножить элементы каждого столбца матрицы с соответствующими элементами K-го столбца.



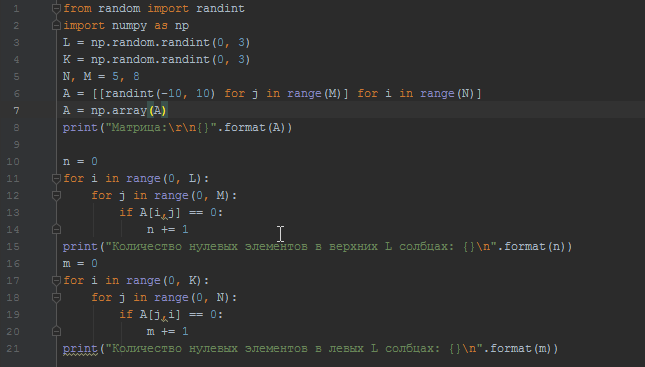


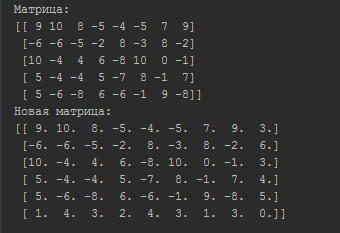
11. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Просуммировать элементы каждой строки матрицы с соответствующими элементами L-й строки.



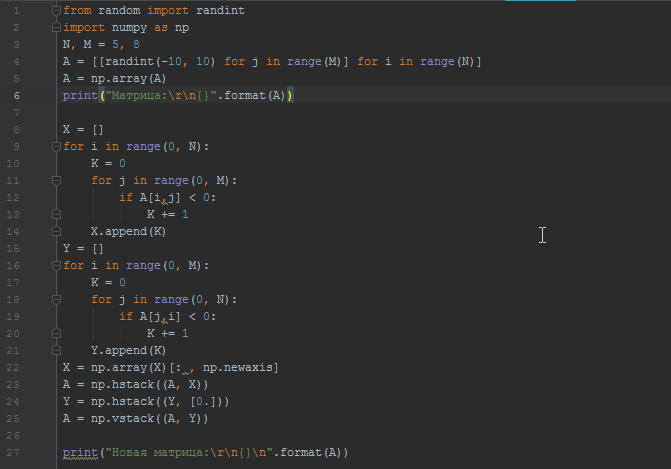


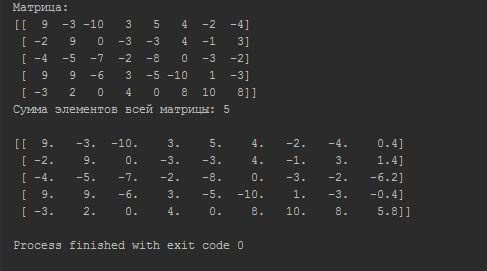
12. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Разделить элементы каждой строки на элемент этой строки с наибольшим значением.



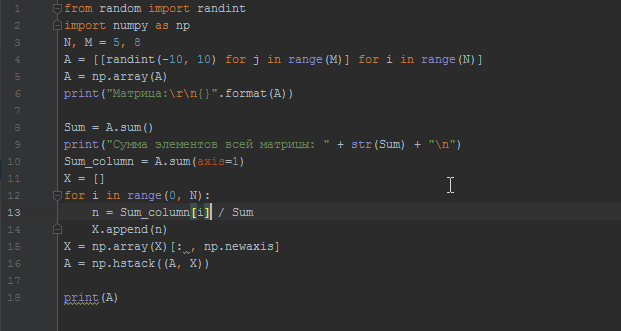


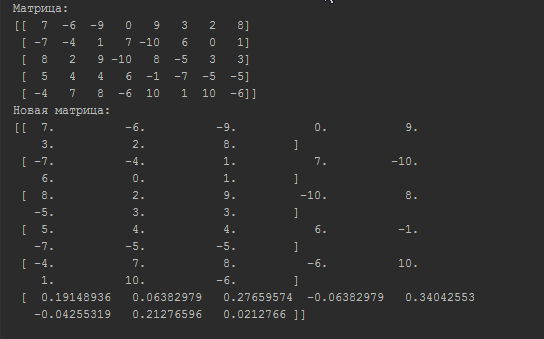
13. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Разделить элементы каждого столбца матрицы на элемент этого столбца с наибольшим значением



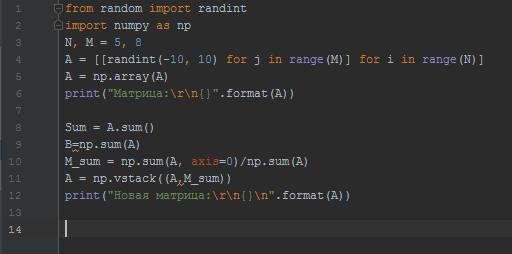


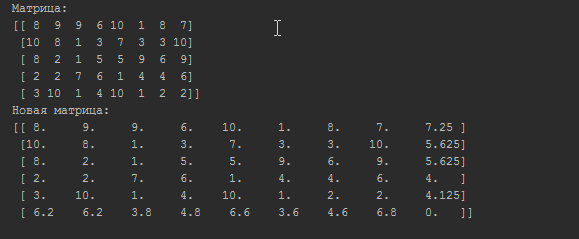
14. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Разделить элементы матрицы на элемент матрицы с наибольшим значением.



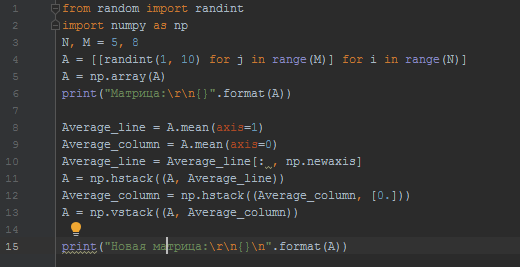


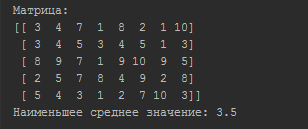
15. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Все элементы имеют целый тип. Дано целое число H. Определить, какие столбцы имеют хотя бы одно такое число, а какие не имеют.



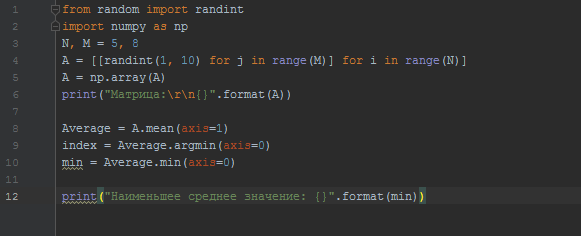


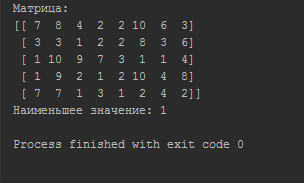
16. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Исключить из матрицы строку с номером L. Сомкнуть строки матрицы.



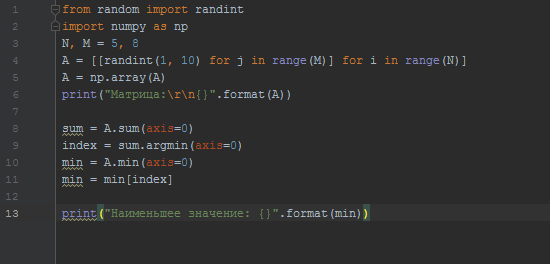


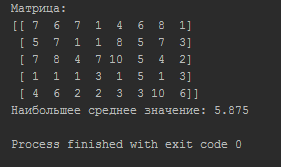
17. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Добавить к матрице строку и вставить ее под номером L.



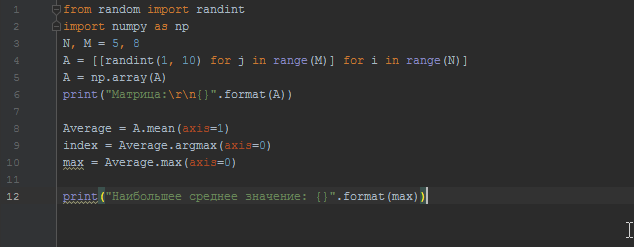


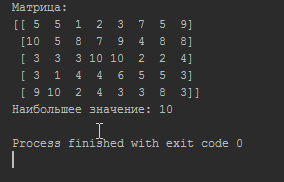
18. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Найти сумму расположенных параллельно побочной диагонали (ближайшие к побочной). Элементы побочной диагонали имеют индексы от [N,0] до [0,N].



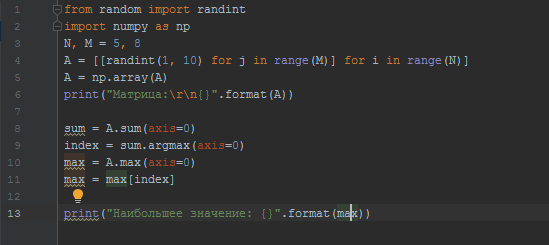


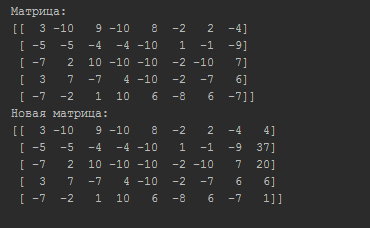
19. Создать квадратную матрицу A, имеющую N строк и N столбцов со случайными элементами. Определить сумму элементов, расположенных параллельно главной диагонали (ближайшие к главной). Элементы главной диагонали имеют индексы от [0,0] до [N,N].



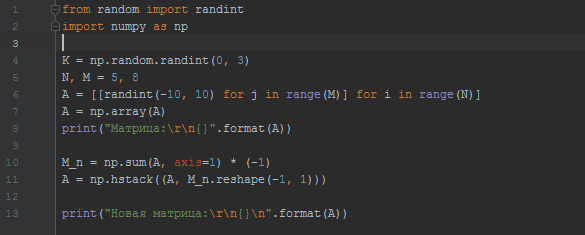


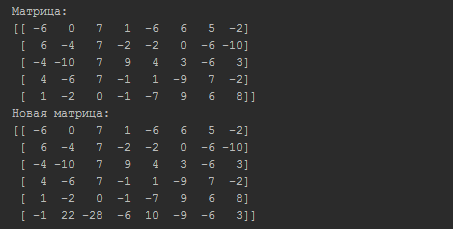
20. Создать квадратную матрицу A, имеющую N строк и N столбцов со случайными элементами. Определить произведение элементов, расположенных параллельно побочной диагонали (ближайшие к побочной). Элементы побочной диагонали имеют индексы от [N,0] до [0,N].



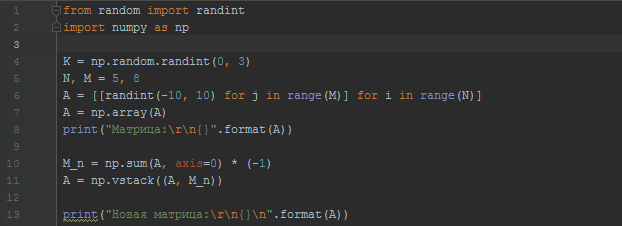


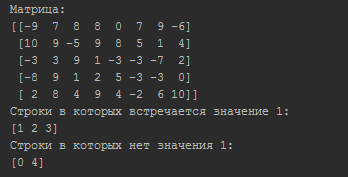
21. Создать квадратную матрицу A, имеющую N строк и N столбцов со случайными элементами. Каждой паре элементов, симметричных относительно главной диагонали (ближайшие к главной), присвоить значения, равные полусумме этих симметричных значений (элементы главной диагонали имеют индексы от [0,0] до [N,N]).



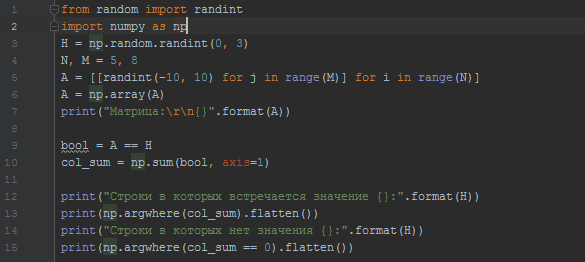


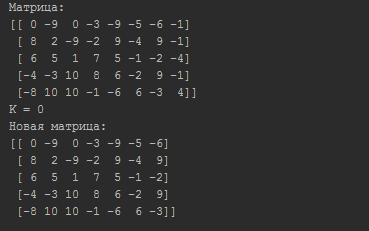
22. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Исходная матрица состоит из нулей и единиц. Добавить к матрице еще один столбец, каждый элемент которого делает количество единиц в каждой строке чётным.



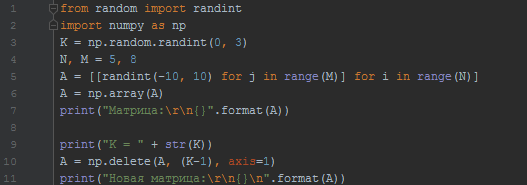


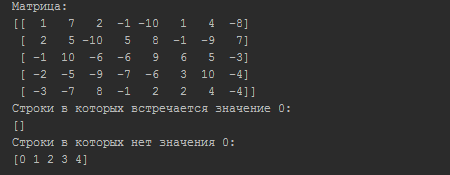
23. Создать квадратную матрицу A, имеющую N строк и N столбцов со случайными элементами. Найти сумму элементов, расположенных выше главной диагонали, и произведение элементов, расположенных выше побочной диагонали (элементы главной диагонали имеют индексы от [0,0] до [N,N], а элементы побочной диагонали — от [N,0] до [0,N]).



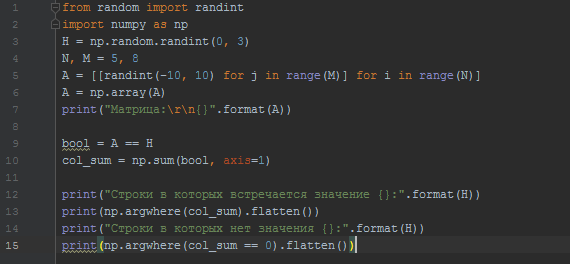


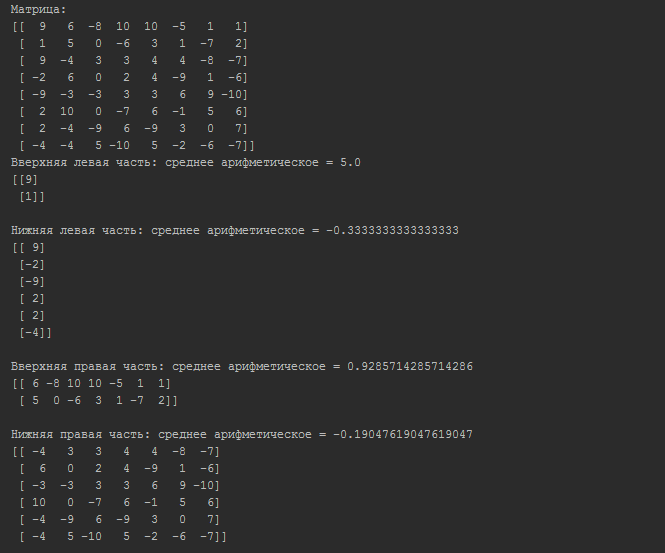
24. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Дан номер строки L и номер столбца K, при помощи которых исходная матрица разбивается на четыре части. Найти сумму элементов каждой части.



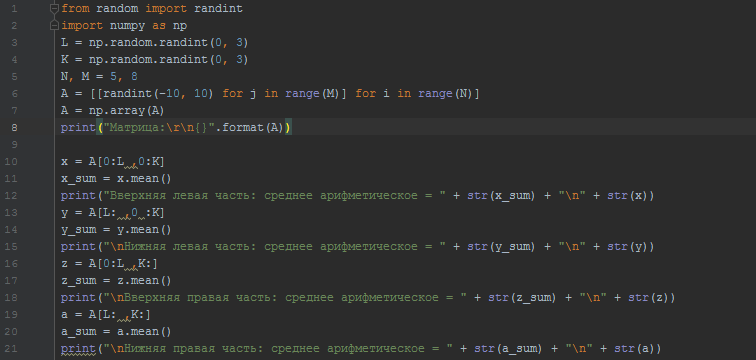


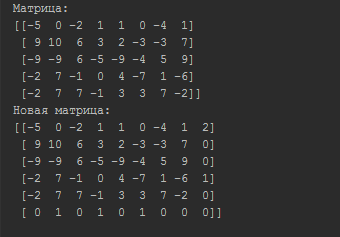
25. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Определить, сколько нулевых элементов содержится в каждом столбце и в каждой строке матрицы. Результат оформить в виде матрицы из N + 1 строк и M + 1 столбцов.



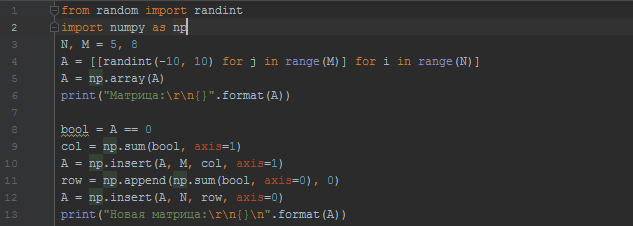


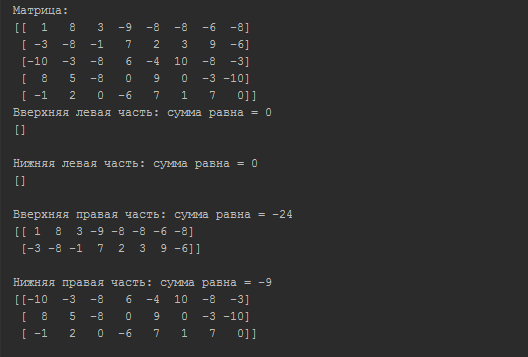
26. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Дан номер строки L и номер столбца K, при помощи которых исходная матрица разбивается на четыре части. Найти среднее арифметическое элементов каждой части.



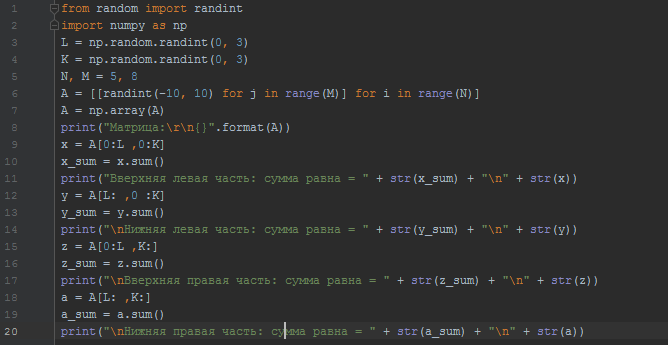


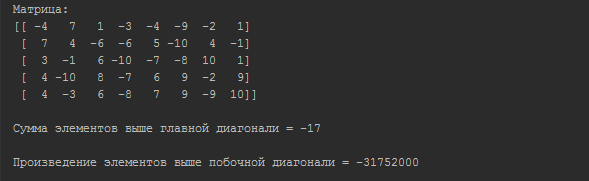
27. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Все элементы имеют целый тип. Дано целое число H. Определить, какие строки имеют хотя бы одно такое число, а какие не имеют.



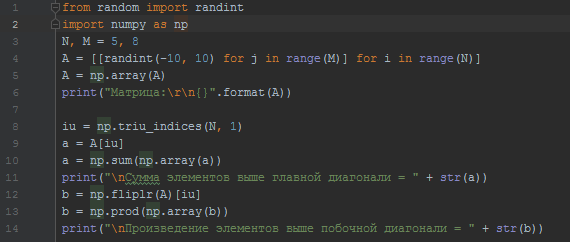


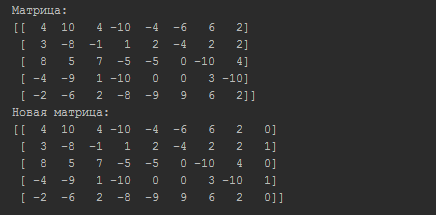
28. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Исключить из матрицы столбец с номером K. Сомкнуть столбцы матрицы.



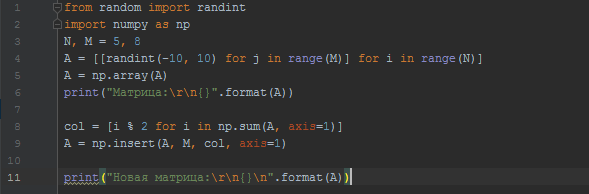


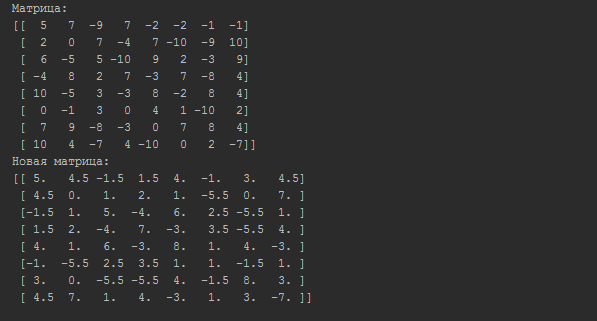
29. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Добавить к матрице столбец чисел и вставить его под номером K.



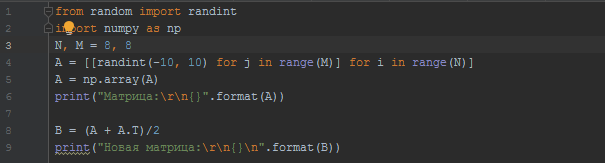


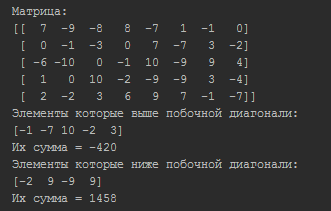
30. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Добавить к элементам каждого столбца такой новый элемент, чтобы сумма положительных элементов стала бы равна модулю суммы отрицательных элементов. Результат оформить в виде матрицы из N + 1 строк и M столбцов.





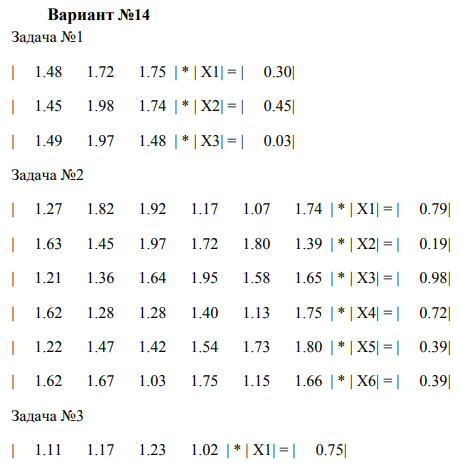
31. Создать прямоугольную матрицу A, имеющую N строк и M столбцов со случайными элементами. Добавить к элементам каждой строки такой новый элемент, чтобы сумма положительных элементов стала бы равна модулю суммы отрицательных элементов. Результат оформить в виде матрицы из N строк и M + 1 столбцов.

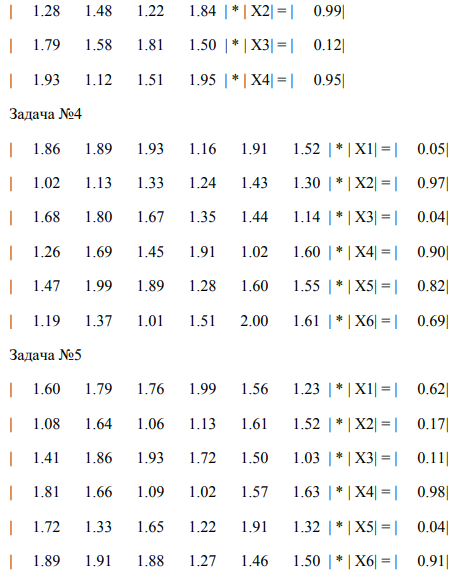




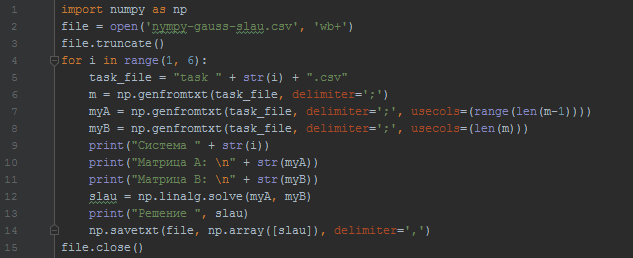
# РЕШЕНИЕ СЛАУ

В данной части нам предстоит решить данные системы методом Гаусса:

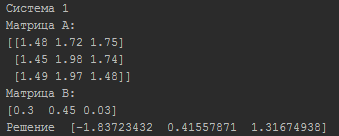
****

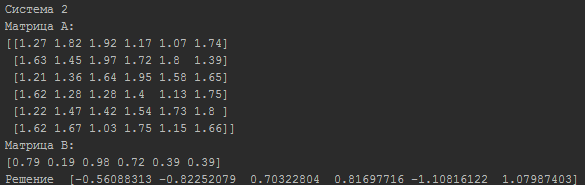
****

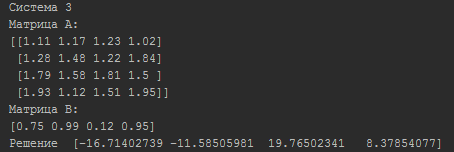
Для того что бы решить эти системы, была написанная программа которая считывает с cvs-файла систему и решает её.

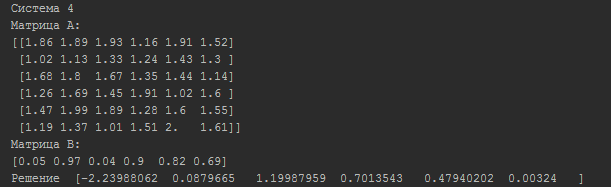


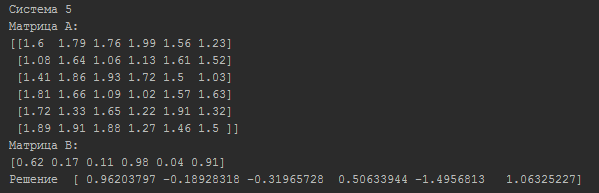
Результат:











# ЗАКЛЮЧЕНИЕ

Несмотря на ряд проблем исторически присущих Python, он продолжает оставаться лидирующим инструментом в ряде ниш:

* Разработка веб-приложений (тут у нас бесусловный лидер в виде Django).
* Анализ данных и машинное обучение (пакеты scipy, scikit-learn, pandas, numpy признанные мировым ученым сообществом).
* Введение в программирование (pygame, turtle хорошо помогают мотивировать детей начинать кодить).
* Быстрое прототипирование идей в бизнесе за счёт обилия готовых библиотек, низкого порога вхождения в язык и высокой продуктивности программистов, пишущих на Python.
* Написание скриптов (сценариев) для автоматизации задач. Python по-умолчанию поставляется со всеми дистрибутивами unix-like систем и является отличной заменой Bash во всех смыслах.

# СПИСОК ЛИТЕРАТУРЫ

* Марк Лутц. Изучаем Python. 4-е издание.
* Марк Саммерфилд. Программирование на Python 3.
* www.python.org
* www.numpy.org