

ЛАБОРАТОРНАЯ РАБОТА N1

ОТРАБОТКА ЗНАНИЙ НА ЯЗЫКЕ PYTHON

Цель: ОТРАБОТКА ЗНАНИЙ НА ЯЗЫКЕ PYTHON

Теория:

Для выполнения работы не нужны не IDE, не программа для выполнения код. Нам только понадобится лишь онлайн компиляторы.

Зачем нужны онлайн компиляторы

Тренд последних пяти лет - онлайн компиляторы. Традиционным оффлайн компиляторам на смену приходят интерактивные онлайн-сервисы. Теперь, для того чтобы выполнить программу, написанную на Паскале или C++, Питоне или Java, вовсе не обязательно ставить на компьютере соответствующий софт. Можно просто зайти на сайт, добавить или написать свой код и отправить его на компиляцию.

5 причин использовать онлайн компилятор:

- требуется быстро проверить некую идею или алгоритм;
- во время занятий онлайн надо показать учащимся работу кода;
- под рукой нет компьютера, а есть только смартфон и надо выполнить код;
- надо в интерактивную лекцию встроить онлайн пример кода;
- онлайн компилятор нужен эпизодически и нет смысла ставить его на компьютер;

Кроме того, онлайн компиляторы часто позволяют сохранить ссылку на выполненный код и передать эту ссылку другому пользователю. Например, так преподаватель может показать ученику как надо было правильно решить задачу. Да и сами коды сохраняются в облачном хранилище, что также удобно.

Недостатков в использовании онлайн-компиляторов немного: требуется подключение к интернету; приходится дольше ждать результат компиляции и, самое главное, приходится довольствоваться той конфигурацией компилятора, которая есть - нельзя добавить свои библиотеки или пакеты. Однако для учебных целей эти недостатки не существенны, зато у школьников или студентов меньше проблем с их инсталляцией.

В завершение приведем пару ссылок на очень простые и удобные компиляторы с возможностью ввода своих данных. Отличный и одновременно простой онлайн компилятор C++. Также новичкам в программировании пригодится Python онлайн для тренировки и обучения. К указанным компиляторам еще как бонус есть примеры кодов типовых задач, которые приходится решать буквально всем, кто начинает учить программирование.

Для выбора рекомендуется к использованию <https://www.onlinegdb.com>, где можно авторизоваться и сохранять код, либо <https://replit.com/languages/Python3>, либо аналогичные

Предупреждение: не для всех онлайн компиляторов работает русский язык

Ход работы:

1. Изучить теорию
2. Изучить теорию по языку python (если у кого нет таких знаний)
3. Решить задачи
4. По окончании написать отчёт по каждому блоку, цель, фото
5. Защитить работы перед преподавателем
6. Выложить отчёт

Защита происходит следующим образом: как только работа готова вся, а не 1-2 задачи, зовём преподавателя. На выбор несколько задач будут спрашиваться как вы их решили и как они работают.

НАПОМИНАНИЕ: Списанная работа, любым способом, либо не будет проверяться, либо будет даваться аналогичная в три раза сложнее.

Работа считается выполненная, если вы сдали и защитили её на паре.

Требование к оформлению отчёта:

Отчет должен содержать следующее:

1. Титульный лист. Берём отсюда <https://vksit.ru/#/edumaterials>, изменяем кто сделал, кому, л\р название, дата сдачи
2. Цель работы.
3. По каждому заданию оформить:
 - Текст задания
 - Работа программы
 - Назначение переменных
 - Текст программы
- Тестирование (скрин запуска работы и её функциональности)
4. Вывод (В ходе данной лабораторной работы были освоены/получены/изучены ...). Пишется по цели + обозначаются основные результаты вашей работы.

Защита лабораторной: готовый отчет в электронном виде и программы для тестирования показываем преподавателю. Если замечаний нет или они исправлены, выкладываем на портал отчет в формате .pdf (сами программы

выкладывать не надо).

Практика:

1 блок. [Ввод и вывод данных](#)

Заходим на сайт, если нужно регистрируемся, читаем теорию, если нужно.

Для успешного закрытия этого блока делаем 5 задач

2 блок [Условия](#)

Для успешного закрытия этого блока делаем 8 последних задач

Плюс:

1. На вход подаётся любое число. Вывести чётным оно является или нет, для отрицательных тоже.

3 блок [Вычисления](#)

Для успешного закрытия этого блока делаем 8 любых задач

Плюс:

1. Напишите программу, которая принимает значение (x) и возвращает «Bang», если число делится на 3, «Boom», если оно делится на 5, «BangBoom», если оно делится на 3 и 5, и «Miss», если оно не делится ни на одно из них. Примечание. Ваша программа должна возвращать только одно значение.

Пример: Ввод: 105 --> Вывод: "BangBoom"

Пример: Ввод: 9 --> Вывод: " Bang "

Пример: Ввод: 25 --> Вывод: " Boom "

4 блок [Цикл for](#)

Для успешного закрытия этого блока делаем все задачи

Плюс:

1. Посчитать факториал, через цикл

*Пример 4 Factorial = $4! = 4 * 3 * 2 * 1 = 24$*

*6 Factorial = $6! = 6 * 5 * 4 * 3 * 2 * 1 = 720$*

2. Если мы перечислим все натуральные числа до 10, кратные 3 или 5, мы получим 3, 5, 6 и 9. Сумма этих кратных равна 23.

Завершите решение так, чтобы оно возвращало сумму всех чисел, кратных 3 или 5, **меньше** переданного числа. Кроме того, если число отрицательное, вернуть 0

Примечание. Если число кратно **и** 3, и 5, считайте его только *один раз*.

5 блок. [Строки](#)

Для успешного закрытия этого блока делаем все задачи

Плюс:

1. Вводится строка произвольного текста, затем вводится произвольный символ. Найти все вхождения этого символа в строку (вывести индексы элементов строки, которые являются этим символом). Если данного символа нет в строке, вывести -1.
2. Вводится строка символов. Удвоить все пробелы. Вывести преобразованную таким образом строку.
3. Вводится строка символов, которые разделены на слова. Пробелы являются разделителями между словами. Вывести на экран самое короткое слово. Если таких слов несколько, вывести одно из них.
4. Вводится строка символов, которые разделены на слова. Пробелы являются разделителями между словами. Поменять местами последний символ первого слова и первый символ последнего слова. Вывести преобразованную таким образом строку. Например, было «мама мыла раму», должно получиться «мамр мыла ааму»

К этому блоку на доп. оценку:

Задание 1. Вводится строка символов, которые разделены на слова.

Пробелы являются разделителями между словами. Вывести на экран все слова, состоящие из четырех букв.

Задание 2. Вводится строка символов, которые разделены на слова.

Пробелы являются разделителями между словами. Во втором слове, если оно есть, поменять местами первый символ с последним. Вывести преобразованную таким образом строку.

Задание 3. Вводится строка символов, которые разделены на слова.

Пробелы являются разделителями между словами. Вывести на экран все слова, состоящие из n-букв (n вводится с клавиатуры).

Задание 4. Вводится строка символов, которые разделены на слова.

Пробелы являются разделителями между словами, причем слова могут разделяться более чем одним пробелом, в начале и конце строки пробелы также допускаются. Удвоить гласные буквы в предпоследнем слове введенной строки. Если в строке только одно слово или совсем нет слов – оставить строку без изменения. Вывести преобразованную таким образом строку

Задание 5. Вводится строка символов, которые разделены на слова.

Пробелы являются разделителями между словами. Вывести на экран самое длинное слово. Если таких слов несколько, вывести одно из них.

Задание 6. Вводится строка символов, которые разделены на слова.

Пробелы являются разделителями между словами. Вывести на экран все слова, заканчивающиеся на букву, введенную с клавиатуры.

6 блок. [Цикл while](#)

Для успешного закрытия этого блока делаем половину задач (8)

Плюс: с помощью данного цикла

1. Возвращает количество гласных в заданной строке. Мы рассмотрим а, е, и, о, и как гласные для этого (но не у английские).
Входная строка будет состоять только из строчных букв и/или пробелов.
2. Ваша задача — написать функцию, которая принимает строку и возвращает новую строку, в которой удалены все гласные.
Например, строка «Этот сайт для неудачников, LOL!» станет «Ths wbst s fr lsrs LL!».
Примечание: у не считается гласной.

К этому блоку на доп. оценку: доделать до конца

7 блок. [Списки](#)

Для успешного закрытия этого блока делаем половину задач (8)

Плюс:

1. Вводится строка произвольного текста с пробелами. Пробелы убираются и каждое отдельное слово записывается в элемент списка.
2. С предыдущей задачи, вывести все строчные буквы из каждого элемента списка. Если таких букв нет, вывести -1.
3. С 1 задачи. Удвоить все цифры в каждом элементе списка. Вывести преобразованную таким образом список со всеми строками.
4. Дополнить предыдущую задачу, если в каком-то элементе списка нет цифр, в конец поставить 0, остальное по 3 задаче

К этому блоку на доп. оценку: доделать до конца

8 блок. [Функции и рекурсия](#)

Для успешного закрытия этого блока делаем всё

Плюс:

1. Напишите функцию, которая принимает массив из 10 целых чисел (от 0 до 9), которая возвращает строку этих чисел в виде номера телефона. **Пример**

```
createPhoneNumber([1, 2, 3, 4, 5, 6, 7, 8, 9, 0]) // => returns "(123) 456-7890"
```

2. Дополнить предыдущую задачу. Пропускать числа делимые на 3. Переходить к следующему элементу, но с таким же условием, от 0 до 9, 10 чисел.

На до оценку:

ОПИСАНИЕ:

Вы, наверное, знаете систему «лайков» из Facebook и других страниц. Люди могут «лайкать» сообщения в блогах, изображения или другие элементы. Мы хотим создать текст, который должен отображаться рядом с таким элементом.

Реализуйте функцию, которая принимает массив, содержащий имена людей, которым понравился элемент. Он должен возвращать отображаемый текст, как показано в примерах:

```
[ ] --> "no one likes this"
["Peter"] --> "Peter likes this"
["Jacob", "Alex"] --> "Jacob and Alex like this"
["Max", "John", "Mark"] --> "Max, John and Mark like this"
["Alex", "Jacob", "Mark", "Max"] --> "Alex, Jacob and 2 others like this"
```

1. Примечание. Для 4 и более имен число в "and 2 others" просто увеличивается.
2. Дано число, минимум трёхзначное. Написать функцию переворота числа и преобразования в текст
Пример: 123 – «321»
3. Изменить предыдущую задачу, так что бы можно было вводить несколько чисел через пробел в 1 строку. А выводило по отдельности с последнего до 1

Дан массив целых чисел, найдите то, которое встречается нечетное количество раз.

Всегда будет только одно целое число, которое встречается нечетное количество раз.

Примеры

[7] должен вернуться 7, потому что это происходит 1 раз (что нечетно).
[0] должен вернуться 0, потому что это происходит 1 раз (что нечетно).
[1, 1, 2] должен вернуться 2, потому что это происходит 1 раз (что нечетно).
[0, 1, 0, 1, 0] должен вернуться 0, потому что это встречается 3 раза (что нечетно).
[1, 2, 2, 3, 3, 3, 4, 3, 3, 3, 2, 2, 1] должен вернуться 4, потому что он появляется 1 раз (что нечетно).

- 4.
5. С клавиатуры вводится строка символов, представляющая из себя произвольное арифметическое выражение. Проверить баланс открывающихся и закрывающихся скобок. Вывести каких скобок больше, если баланс не соблюден. Предусмотреть вариант, когда скобки в выражении отсутствуют.

9 блок. [Двумерные массивы](#)

Для успешного закрытия этого блока делаем всё

Плюс:

10 блок. [Множества](#)

Для успешного закрытия этого блока делаем всё

Плюс:

11 блок. [Словари](#)

Для успешного закрытия этого блока делаем всё

Плюс: