

С вопросами 1, 2 и 6 задачам четвертого ДЗ можно писать Дмитрию Пензару. По 3, 4, 5 и 7 - Дианкину Игорю. Дедлайн: 6 ноября.

Домашнее задание №4

Задание 1 (2 балла)

Мнемоника: *ClosedHash*

Необходимо написать хэш-таблицу с закрытой адресацией.

Запрещается: Запрещается использовать словари, множества и прочие подобные им объекты. Списки можно. Встроенную функцию `hash()` тоже можно.

Все ключи - строки. Все значения - числа

Должны присутствовать функции:

1. **append key value** (добавляет пару ключ значение в таблицу, возвращает Done)
2. **print** (выдает все пары записанные в таблице)
3. **delete key** (удаляет пару по ключу из таблицы, при успехе возвращает Done, если пары в таблице нет, возвращает None)
4. **get key** (возвращает значение по ключу при успехе, если пары в таблице нет, возвращает None)

In:	Набор строк, где на каждой строке записана команда, которую необходимо выполнить
Out:	Результаты выполнения команд

Input	append a 1 append b 2 print delete a delete c get b print
Output	INPUT: append a 1 OUTPUT: Done INPUT: append b 2 OUTPUT: Done INPUT: print OUTPUT: (a 1) @ (b 2) INPUT: delete a OUTPUT: Done INPUT: delete c OUTPUT: None

	INPUT: get b OUTPUT: 2 INPUT: print OUTPUT: (b 2)
--	--

Задание 2 (2 балла)

Мнемоника: Bloom

Фильтр Блума.

Реализовать эту структуру данных и использовать его для хранения информации о наличии/отсутствии пар ключ-значение в динамическом массиве (можно использовать Python list) Встроенную функцию hash() тоже можно.

Реализовать функции:

1. **append key, value** - Добавление пары key-value в динамический массив. *Если ключ уже есть в таблице - перезаписывает значение*
2. **check key** - Проверка наличия ключа key. Выполняется только на уровне фильтра Блума. Возвращает ответ фильтра - NO, если элемента нет, MAYBE иначе. Будет проверять то, что ваш фильтр не дает ложноотрицательных срабатываний.
3. **find key** - если ключ есть в хранилище, то возвращает value. Если его нет - то возвращает None. Должна быть выполнена с использованием фильтра Блума. Таким образом, поиск объекта, который есть выполняется за $O(N)$, где N - число добавленных ключей. Если объекта нет - то в среднем поиск должен производиться за $O(1)$ (ваш фильтр Блума должен ошибаться нечасто)

Запрещается: Запрещается использовать словари, множества и прочие подобные им объекты. Списки можно.

In:	Набор строк, где на каждой строке записана команда, которую необходимо выполнить
Out:	Результаты выполнения команд

Input	find UCGACT find TCUUTG append CCAGTC 9429
-------	--

	append TCGTUC 3646 append GCUTGC 2386 find CTUTAA find UACCTG append CUTCCT 9166 append AAGACA 9830 append UGGTCA 4624 check CUTCCT
Output	None None Done Done Done None None Done Done Done MAYBE

Задание 3 (2 балла)

Мнемоника: rabin

Поиск подстроки в строке алгоритмом Рабина-Карпа.

Инпут читается из файла **in.txt**, записывается в файл **out.txt**

В файле одна строка, в ней строка и паттерн разделенные пробелом.

В случае отсутствия находок писать пустую строку.

Алфавит состоит из букв A T G C.

Input	ATGATGATG ATG
Output	0 3 6

Задание 4 (1 балл)

Мнемоника: prefsuf

Префикс функция

Инпут читается из файла **in.txt**, записывается в файл **out.txt**

В файле одна строка, в ней слово, в аутпутный файл необходимо записать все значения суффикс-префикс функции через пробел.

Input	abcabcabc
Output	0 0 0 1 2 3 4 5 6

Задание 5 (1 балл)

Мнемоника: kmp

Поиск подстроки в строке алгоритмом Кнута-Морриса-Пратта.

Инпут читается из файла **in.txt**, записывается в файл **out.txt**

В файле одна строка, в ней строка и паттерн разделенные пробелом.

В случае отсутствия находок писать пустую строку.

Input	abcabcabc abc
Output	0 3 6

*Задание 6 (3 балла)

Мнемоника: OpenHash

(Дополнительное) Необходимо написать хэш-таблицу с открытой адресацией.

Запрещается: Запрещается использовать словари, множества и прочие подобные им объекты. Списки можно.

Все ключи - строки. Все значения - числа

Должны присутствовать функции:

1. **append key value** (добавляет пару ключ-значение в таблицу, возвращает Done. Если ключ уже есть в таблице - перезаписывает значение)
2. **print** (выдает все пары ключ-значение, записанные в таблице, **в порядке их первого добавления в таблицу**)
3. **delete key** (удаляет пару по ключу из таблицы, при успехе возвращает Done, если пары в таблице нет, возвращает None)
4. **get key** (возвращает значение по ключу при успехе, если пары в таблице нет, возвращает None)

In:	Набор строк, где на каждой строке записана команда, которую необходимо выполнить
Out:	Результаты выполнения команд

Input	append a 1 append b 2 print delete a delete c get b print
Output	INPUT: append a 1 OUTPUT: Done INPUT: append b 2 OUTPUT: Done INPUT: print OUTPUT: (a 1) @ (b 2) INPUT: delete a OUTPUT: Done INPUT: delete c OUTPUT: None INPUT: get b OUTPUT: 2 INPUT: print OUTPUT: (b 2)

*Задание 7 (3 балла)

Мнемоника: palindrome

(Дополнительное) Реализовать поиск палиндромов с использованием хеширования.

Вам подается строка и требуется найти количество всех подстрок-палиндромов в ней.

Подстрока может быть равна самой строке. Можно использовать встроенную функцию hash().

Решение должно работать за $O(n \log n)$

Input	AB
Output	0

Input	ABABA
Output	4