

Тема: объекты в C++: шаблоны.

Вариант: 1

Задача: Разработать шаблонный класс :

```
template <typename K, typename T>
class HashMap {
    ...
}
```

реализующий хеш-таблицу, где K и T типы ключа и значения соответственно.

В качестве хеш-функции использовать `std::tr1::hash` из стандартной библиотеки.

Коллизии разрешать **методом цепочек**.

В классе `HashMap` реализовать:

- 1) Добавление нового элемента по паре <ключ, значение>. Если элемент с таким ключом уже присутствует в хеш-таблице, обновить его значение.
- 2) Удаления элемента по ключу.
- 3) Поиск элемента по ключу.
- 4) Класс-итератор, позволяющий перебирающий элементы хэш-таблицы. Для итератора реализовать операторы `++`, `*` и `->`
- 5) Механизм перехэширования таблицы при большом количестве элементов.

В качестве демонстрационного примера написать программу, считывающую из файла набор команд для работы с хеш-таблицей, выполняющих их и выводящих результат.

Входные данные:

В первой строчке входного файла заданы два символа, указывающих тип ключа и значения соответственно: I -> int, D -> double, S -> std::string; (вместо std::string можно использовать собственный класс string из задачи №1, вариант №2)

Во второй строке задается число N – количество команд.

В следующих N строках описываются команды. Команды могут быть двух типов:

- 1) “A key value” -- добавление в хеш-таблицу пары <key, value>. Если элемент с таким ключом уже есть в хеш-таблице, его значение должно быть обновлено на value.
- 2) “R key” – удаление из хеш-таблицы элемента с ключом key, если таковой имеется.

Выходные данные:

В выходной файл записать два числа: общее количество элементов в таблице после выполнение команд, и количество уникальных элементов в таблице после выполнения команд.

Под “уникальными” понимаются элементы с разными value.

Пример входных и выходных данных:

input.txt	output.txt
I S 5 A 783 Ivanov A 999 Petrova A 986 Sidorov R 986 A 111 Kuznecov	3 3
S D 4 A Ivanov 1.84 A Petrova 1.66 A Ivanov 1.90 A Sidorov 1.90	3 2