

1. Реализовать на C++ хеш-таблицу целых чисел по методу списков (с хеш-функцией на основе деления). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление числа в хеш-таблицу, удаление числа из хеш-таблицы, поиск числа в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу и заполняющей её заданным количеством случайных целых чисел.
2. Реализовать на C++ хеш-таблицу целых чисел по методу линейных проб (с хеш-функцией на основе деления). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление числа в хеш-таблицу, удаление числа из хеш-таблицы, поиск числа в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу и заполняющей её заданным количеством случайных целых чисел.
3. Реализовать на C++ хеш-таблицу целых чисел по методу списков (с хеш-функцией на основе середины квадрата). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление числа в хеш-таблицу, удаление числа из хеш-таблицы, поиск числа в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу и заполняющей её заданным количеством случайных целых чисел.
4. Реализовать на C++ хеш-таблицу целых чисел по методу линейных проб (с хеш-функцией на основе середины квадрата). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление числа в хеш-таблицу, удаление числа из хеш-таблицы, поиск числа в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу и заполняющей её заданным количеством случайных целых чисел.
5. Реализовать на C++ хеш-таблицу целых чисел по методу списков (с хеш-функцией на основе умножения). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление числа в хеш-таблицу, удаление числа из хеш-таблицы, поиск числа в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу и заполняющей её заданным количеством случайных целых чисел.

6. Реализовать на C++ хеш-таблицу целых чисел по методу линейных проб (с хеш-функцией на основе умножения). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление числа в хеш-таблицу, удаление числа из хеш-таблицы, поиск числа в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу и заполняющей её заданным количеством случайных целых чисел.
7. Реализовать на C++ хеш-таблицу строк `std::string` по методу списков (с хеш-функцией на основе деления). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление строки в хеш-таблицу, удаление строки из хеш-таблицы, поиск строки в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и заполняющей её ими.
8. Реализовать на C++ хеш-таблицу строк `std::string` по методу линейных проб (с хеш-функцией на основе деления). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление строки в хеш-таблицу, удаление строки из хеш-таблицы, поиск строки в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и заполняющей её ими.
9. Реализовать на C++ хеш-таблицу строк `std::string` по методу списков (с хеш-функцией на основе середины квадрата). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление строки в хеш-таблицу, удаление строки из хеш-таблицы, поиск строки в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и заполняющей её ими.
10. Реализовать на C++ хеш-таблицу строк `std::string` по методу линейных проб (с хеш-функцией на основе середины квадрата). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление строки в хеш-таблицу, удаление строки из хеш-таблицы, поиск строки в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и заполняющей её ими.

11. Реализовать на C++ хеш-таблицу строк `std::string` по методу списков (с хеш-функцией на основе умножения). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление строки в хеш-таблицу, удаление строки из хеш-таблицы, поиск строки в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и заполняющей её ими.
12. Реализовать на C++ хеш-таблицу строк `std::string` по методу линейных проб (с хеш-функцией на основе умножения). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление строки в хеш-таблицу, удаление строки из хеш-таблицы, поиск строки в хеш-таблице. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и заполняющей её ими.
13. Реализовать на C++ хеш-таблицу пар {строка `std::string`, целое число} по методу списков (с хеш-функцией на основе деления). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление пары в хеш-таблицу, удаление пары из хеш-таблицы, поиск пары в хеш-таблице, увеличение/уменьшение числа, соответствующего строке. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и подсчитывающей для каждого слова количество его использований в файле.
14. Реализовать на C++ хеш-таблицу пар {строка `std::string`, целое число} по методу линейный проб (с хеш-функцией на основе деления). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление пары в хеш-таблицу, удаление пары из хеш-таблицы, поиск пары в хеш-таблице, увеличение/уменьшение числа, соответствующего строке. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и подсчитывающей для каждого слова количество его использований в файле.

15. Реализовать на C++ хеш-таблицу пар {строка std::string, целое число} по методу списков (с хеш-функцией на основе середины квадрата). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление пары в хеш-таблицу, удаление пары из хеш-таблицы, поиск пары в хеш-таблице, увеличение/уменьшение числа, соответствующего строке. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и подсчитывающей для каждого слова количество его использований в файле.
16. Реализовать на C++ хеш-таблицу пар {строка std::string, целое число} по методу линейный проб (с хеш-функцией на основе середины квадрата). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление пары в хеш-таблицу, удаление пары из хеш-таблицы, поиск пары в хеш-таблице, увеличение/уменьшение числа, соответствующего строке. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и подсчитывающей для каждого слова количество его использований в файле.
17. Реализовать на C++ хеш-таблицу пар {строка std::string, целое число} по методу списков (с хеш-функцией на основе умножения). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление пары в хеш-таблицу, удаление пары из хеш-таблицы, поиск пары в хеш-таблице, увеличение/уменьшение числа, соответствующего строке. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и подсчитывающей для каждого слова количество его использований в файле.
18. Реализовать на C++ хеш-таблицу пар {строка std::string, целое число} по методу линейный проб (с хеш-функцией на основе умножения). Необходимые методы: инициализация хеш-таблицы (конструктор), удаление хеш-таблицы (деструктор), добавление пары в хеш-таблицу, удаление пары из хеш-таблицы, поиск пары в хеш-таблице, увеличение/уменьшение числа, соответствующего строке. Также необходима реализация функции, создающей хеш-таблицу, считывающей слова из текстового файла и подсчитывающей для каждого слова количество его использований в файле.