МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Дальневосточный федеральный университет»**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра прикладной математики, механики, управления и программного обеспечения**

**О Т Ч Е Т**

по лабораторной работе №5 по дисциплине “Структуры и алгоритмы компьютерной обработки данных”

|  |
| --- |
| Выполнил студент гр. Б8204  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д. А. Федоренко |
| Проверил к.т.н. доцент  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Остроухова |

г. Владивосток

2018

**Неформальная постановка задачи**

Разработать шаблонный класс кольцевой очереди.

**Спецификация класса**

template <class T>

class Queue

{

public:

Queue(int n = 10); // Конструктор

Queue(const Queue&); // Конструктор копирования

int pushElement (T); // Функция добавления элемента в очередь

T popElement (); // Функция удаления чтения элемента из очереди

int length (); // Функция, определения длины очереди

void printAll(); // Функция печати массива array

int print(); // Функция печати очереди

~Queue(); // Деструктор

private:

T\* array = nullptr;

int SIZE; // Переменная, хранящая длину массива

H = 0; // Переменная, указывающая на начало очереди

T = -1; // Переменная, указывающая на последний добавленный элемент

};

**Спецификация функций**

int **pushElement** (T **newElement**) - Функция добавления элемента в очередь.

На вход принимает элемент **newElement** и возвращает значение, которое находится в ячейке массива с индексом **H**:

* 1, если очередь полная
* 0, если добавление элемента **newElement** прошло успешно

T **popElement** () - Функция удаления чтения элемента из очереди.

На вход ничего не принимает и возвращает значение, которое находится в ячейке массива **array** с индексом **H**

int **length** () – Функция определения длины очереди.

На вход ничего не принимает и возвращает целочисленное значение равное длине очереди

void **printAll**() - Функция печати массива **array** на экран.

На вход ничего не принимает и ничего не возвращает.

void **print**() - Функция печати очерди на экран.

На вход ничего не принимает и ничего не возвращает.

**Тесты**

**addRecord** (Hash **hash**)**:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ситуация | Входные данные | Таблица до | Таблица после | Значение функции |
| Запись **hash** попадает в таблицу при первом хешировании | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  - - ------ ------- --- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1 | 0 |
| Запись **hash** попадает в таблицу при хешировании вторичной хеш-функцией 1 раз | 2 E737 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 - ---- ------ ---- -1 0 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 2 E737 bbbbb aaa 9 1 | 0 |
| Запись **hash** попадает в таблицу при хешировании вторичной хеш-функцией несколько раз | 2 G737 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 - ---- ------ ---- -1 0 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1  4 - ---- ------ ---- -1 0  5 - ---- ------ ---- -1 0  6 - ---- ------ ---- -1 0  7 - ---- ------ ---- -1 0  8 - ---- ------ ---- -1 0  9 2 G737 bbbbb aaa 9 1 | 0 |
| Запись **hash** уже содержится в таблице | 2 D738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | -1 |
| Совпадение ключей | 2 D738 aaaaa bbb | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | -2 |

**searchRecord** (Hash **hash**)**:**

|  |  |  |  |
| --- | --- | --- | --- |
| Ситуация | Входные данные | Таблица | Значение функции |
| Запись **hash** есть в таблице | 2 G738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 3 |
| Записи **hash** нет в таблице | 2 F738 asdfgh ccc | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | -1 |
| Записи **hash** удалена | 2 G738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 2 | -1 |

**deleteRecord** (Hash **hash):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ситуация | Входные данные | Таблица до | Таблица после | Значение функции |
| Запись **hash** есть в таблице | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 1  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 2  3 2 G738 bbbbb aaa 3 1 | 0 |
| Записи **hash** нет в таблице | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 - ---- ------ ---- -1 0  3 2 G738 bbbbb aaa 3 1 | 1 |
| Запись **hash** уже удалена | 2 F738 bbbbb aaa | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 2  3 2 G738 bbbbb aaa 3 1 | 0 2 D738 bbbbb aaa 0 1  1 2 E738 bbbbb aaa 1 1  2 2 F738 bbbbb aaa 2 2  3 2 G738 bbbbb aaa 3 1 | 1 |