ВШЭ АиСД 2021. Списки (noSTL)

4 дек 2021, 18:54:25
старт: 8 окт 2021, 12:00:00
финиш: 17 окт 2021, 23:59:59
длительность: 9д. 11ч.
начало: 8 окт 2021, 12:00:00
конец: 17 окт 2021, 23:59:59

C. Двусвязный список (0.2)

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод
Вывод	стандартный вывод

Необходимо реализовать **двусвязный список** (класс) с поддержкой следующих операций:

- вставка элемента в начало;
- вставка элемента в конец;
- вставка элемента в середину;
- удаление элемента из некоторой позиции;
- проход по списку от начала к концу;
- проход по списку от конца к началу.

Операции над списком выполняются в соответствии с командами, которые подаются на вход.

Формат ввода

Сначала на вход подаются два целых положительных числа N ($1 \le N \le 2 \cdot 10^4$) и M ($1 \le M \le 100$), разделенных пробелом - количество элементов в начальном списке и количество последующих команд.
Затем на вход подается строка с N числами, разделенными пробелами - начальными элементами двусвязного списка. Гарантируется, что начальные элементы по модулю не превышают 10^9.

Далее на вход подаются M строк, содержащих команды, которые необходимо выполнить над поступившим списком.
Формат команд:

- `push_front <value>` – вставка элемента в начало списка, где `value` – значение, вставляемое в список ($|value| \le 10^9$).
- `push_back <value>` – вставка элемента в конец списка, где `value` – значение, вставляемое в список ($|value| \le 10^9$).
- `insert <value> <position>` – вставка элемента в список, где `value` – значение, вставляемое в список, а `position` – номер элемента, **после которого** надо вставить данный элемент ($|value| \le 10^9$; $0 \le position \le size - 2$).
- `remove <position>` – удаление элемента из списка, где `position` – номер элемента, элемент **после которого** нужно удалить ($0 \le position \le size - 2$).

Нумерация элементов **с нуля**.

Формат вывода

Необходимо вывести элементы связного списка после произведенных над ним операций в обоих направлениях.

Пример 1

Ввод **Вывод**

Ввод	Вывод
10 4 1 2 3 4 5 6 7 8 9 0 push_front 11 remove 2 insert 12 3 push_back 7	11 1 2 4 12 5 6 7 8 9 0 7 7 0 9 8 7 6 5 12 4 2 1 11

Пример 2

Ввод	Вывод
5 4 5 4 3 2 1 remove 0 insert 0 1 remove 3 push_front 6	6 5 3 0 2 2 0 3 5 6

Примечания

Использование массивов **запрещено**.
Таким образом, получается, что список в любой момент времени не пустой (так как 0-ый элемент удалить не получится). Так как изначально в условии про удаление 0-го элемента ничего не было сказано, пусть будет так.

Набрать здесь | Отправить файл

```cpp
#include <iostream>
#include <string>

struct Node {
    Node *nextNode;
    Node *previousNode;
    int value;

    explicit Node(int value_input) {
        value = value_input;
        nextNode = nullptr;
        previousNode = nullptr;
    }
};

class List {
private:
    int size_;
    Node *head_element_;
    Node *tail_element_;

public:
    List() {
        head_element_ = nullptr;
        tail_element_ = nullptr;
        size_ = 0;
    }

    ~List() {
        while (head_element_) {
            tail_element_ = head_element_->nextNode;
            delete head_element_;
            head_element_ = tail_element_;
        }
    }

    bool isEmpty() {
        return head_element_ == nullptr;
```

Отправить

Предыдущая | Следующая