

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра Информационные Системы

ОТЧЕТ
по лабораторной работе №1
по дисциплине «Алгоритмы и структуры данных»
Тема: Списочные структуры

Студент гр. 1373

Быско Н.С.

Преподаватель

Пелевин М.С.

Санкт-Петербург

2022

Цель работы

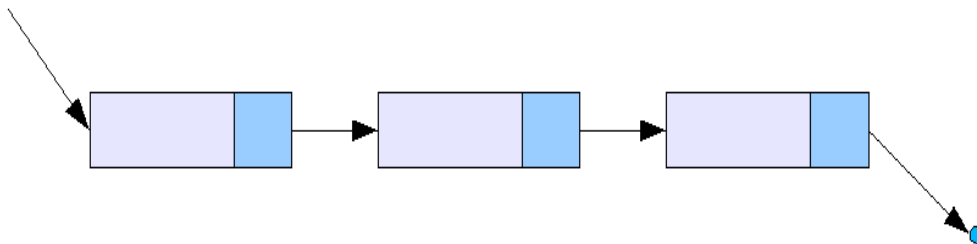
Реализовать следующие структуры: односвязный список, динамический массив и стек. Стек можно реализовать как на базе списка, так и отдельно.

Использовать стек для реализации алгоритма сортировочной станции.

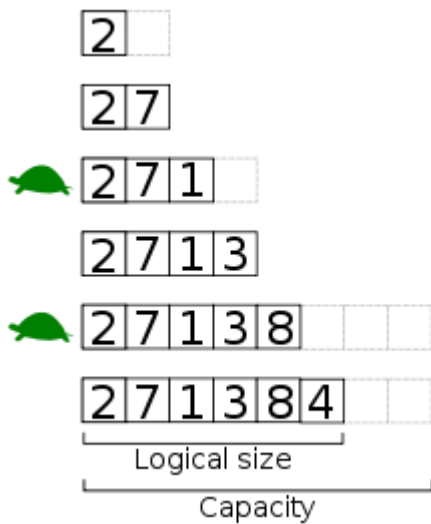
Разрешённые символы в исходном выражении: +, -, *, /, ^, sin, cos, (,), 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Для упрощения разбиения входной строки на токены разрешается отделять каждый символ пробелом.

Основные теоретические положения

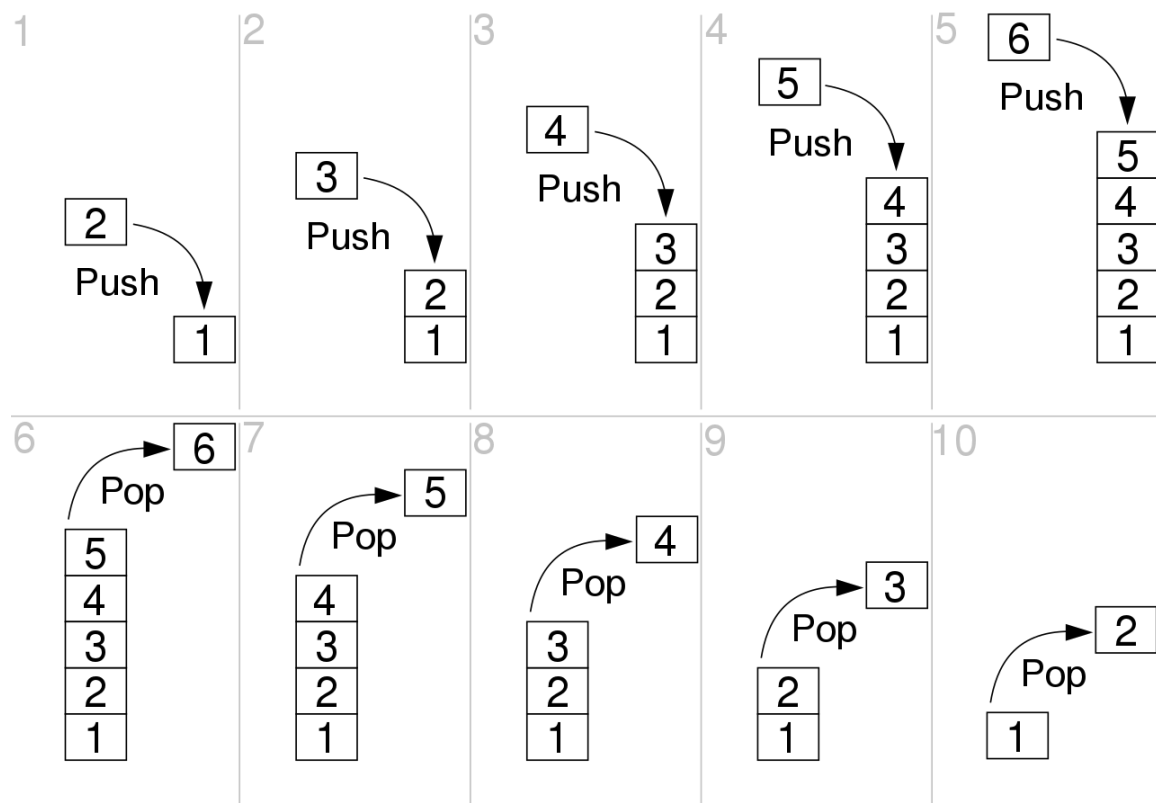
Линейный однонаправленный список — это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей. Каждый элемент списка имеет указатель на следующий элемент. Последний элемент списка указывает на NULL.



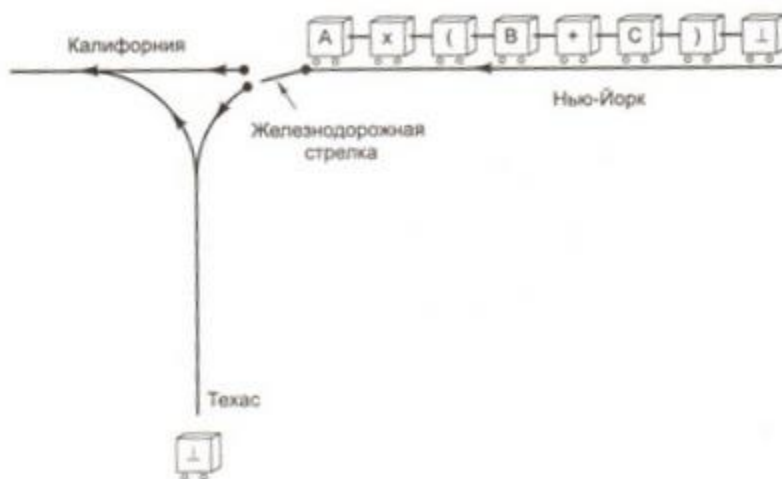
Динамическим называется массив, размер которого может изменяться во время исполнения программы. Возможность изменения размера отличает динамический массив от статического, размер которого задаётся на момент компиляции программы.



Стек (англ. stack — стопка; читается стэк) — абстрактный тип данных, представляющий собой список элементов, организованных по принципу LIFO (англ. last in — first out, «последним пришёл — первым вышел»).



Обратная польская запись (англ. **Reverse Polish notation, RPN**) — форма записи математических и логических выражений, в которой операнды расположены перед знаками операций. Также именуется как обратная бескобочная запись, постфиксная нотация, бескобочная символика Лукасевича, польская инверсная запись, ПОЛИЗ.



Реализация

Каждая структура данных (односвязный список, динамический массив, стек) реализована в отдельном header-файле. В отдельном файле реализована сортировка и тесты, проверяющие корректность работы сортировки.

Для односвязного списка реализованы методы `insert`(добавление по индексу), `remove` (удаление по индексу), `find` (поиск индекса элемента), `getElement`(доступ к значению элемента по заданному индексу), `push_front` (добавление элемента в начало списка), `push_back` (добавление элемента в конец списка), `pop_front` (удаление первого элемента в списке), `GetSize` (получение размера списка).

Для динамического массива реализованы методы `getElement` (получение элемента по индексу), `findValue` (поиск индекса элемента), `Delete` (удаление элемента), `Insert` (добавление элемента), `resize` (изменение размера массива), `size` (размер массива).

Для стека `push` (добавление в стек), `pop`(удаление из стека), `peek`(получение значения без снятия), `resize`(изменение размера стека), `empty`(проверка пустоты стека).

На базе стека реализован алгоритм сортировки.

Пример работы программы

```
enter the expression using 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, ^  
3 * ( sin ( 4 ) - 1 ) + 4 * ( cos ( 2 ) * 7 )  
reverse polish notation:  
3 4 sin 1 - * 4 2 cos 7 * * +  
Для продолжения нажмите любую клавишу . . .
```

Вывод

В ходе работы мы научились создавать и использовать различные структуры данных. Реализовали алгоритм сортировки.