Міністерство освіти і науки України

Львівський національний університет імені Івана Франка

Факультет прикладної математики та інформатики

**ЗВІТ**

**Виконали:**

Студенти групи ПМп-22

Настосяк Віталій

Лозинський Степан

Гаєвська Ірина

Стахів Юлія

Львів 2017

* **ПОСТАНОВКА ЗАВДАННЯ**

Написати бібліотеку, що дає змогу працювати з масивами, одно- і двозв'язними списками,

стеками, чергами, передбачити створення, додавання/вилучення елемента в кінець, на початок,

на конкретну позицію, дописування одного до іншого, обертання порядку елементів, сортування,

пошук. Для деяких типів, наприклад, стека, не всі ці операції можливі, тому необхідно обдумати,

що має зміст реалізовувати, а що - ні.

* **ШЛЯХИ РЕАЛІЗАЦІЇ**

Для того, щоб реалізувати програму, необхідно розуміти, що саме являє собою стек, черга, одно- та двозв’язні списки.

Отож:

Стек — різновид лінійного списку, структура даних, яка працює за принципом (дисципліною) «останній прийшов — перший пішов».

Черга (англ. queue) — динамічна структура даних, що працює за принципом «перший прийшов — перший пішов».

Однозв'язний список — вид зв'язаного списку, який складається з вузлів, кожен з яких містить у собі данні (інформаційну частину) та посилання на наступний вузол.

Двозв'язний список — вид зв'язаного списку, у якому посилання в кожному вузлі вказують на попередній і на подальший вузол у списку.

* **ІНСТРУКЦІЯ З ВИКОРИСТАННЯ**

Отже, ця програма дає змогу працювати із однозв'язними списками, двоз'язними списками, стеками та чергами.

В програмі описані такі операції із стеками:

\* pushBack («заштовхнути елемент»): елемент додається в стек та розміщується в його верхівці. Розмір стеку збільшується на одиницю.

\* pop («виштовхнути елемент»): отримує елемент з верхівки стеку. При цьому він видаляється зі стеку і його місце в верхівці стеку займає наступний за ним відповідно до правила LIFO(Last In, First Out), а розмір стеку зменшується на одиницю.

Функції оголошені у файлі Stack.h, а їх опис у файлі Stack.cpp.

Щоб доступитися до цих функцій, в Source.cpp необхідно створити об'єкт типу Stack, і доступатися до функцій через крапочку (".").

Клас Stack шаблонний (це означає, що ви можете задавати різні типи даних).

\* \* \*

Операції із чергами такі ж як і з стеками, і доступаються до них аналогічно.

Функції оголошені у файлі Queue.h, а їх опис у файлі Queue.cpp.

\* \* \*

В програмі описані такі операції із однозв'язними списками:

\* pushBack - елемент додається в список та розміщується в його верхівці.

\* pop - отримує елемент з верхівки списку.

\* leftpop - видаляє перший елемент зліва.

\* clear - видаляє всі елементи списку.

\* head\_on\_tail - міняє місцями перший елемент списку з останнім.

\* remove - видаляє елемент списку, вибраний користувачем (користувач задає елемент який хоче видалити в одиничних лапках, наприклад pushback('4')).

\* insert - вставляє елемент у список на певну позицію (користувач задає елемент(в одиничних лапках), який хоче вставити у список і номер позиції після якої цей елемент повинен стояти; наприклад insert ('T',1), внаслідок цієї функції елемент Т вставиться на 2 позицію у списку, при цьому розмір списку збільшиться на 1).

\* reverse - розвертає список.

\* sort - сортує список (методом швидкого сортування).

Також можливе додавання 2 списків.

Функції оголошені в файлі List.h, а їх опис у файлі List.cpp.

Доступаємося до цих функцій, так як і з чергами та стеками.

\* \* \*

Операції їз двозв'язними списками такі ж як із однозв'язними.

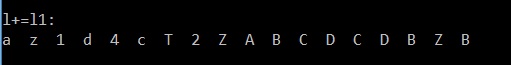
Функції описані в файлі Doubly\_list.h, а їх опис у файлі Doubly\_list.cpp.

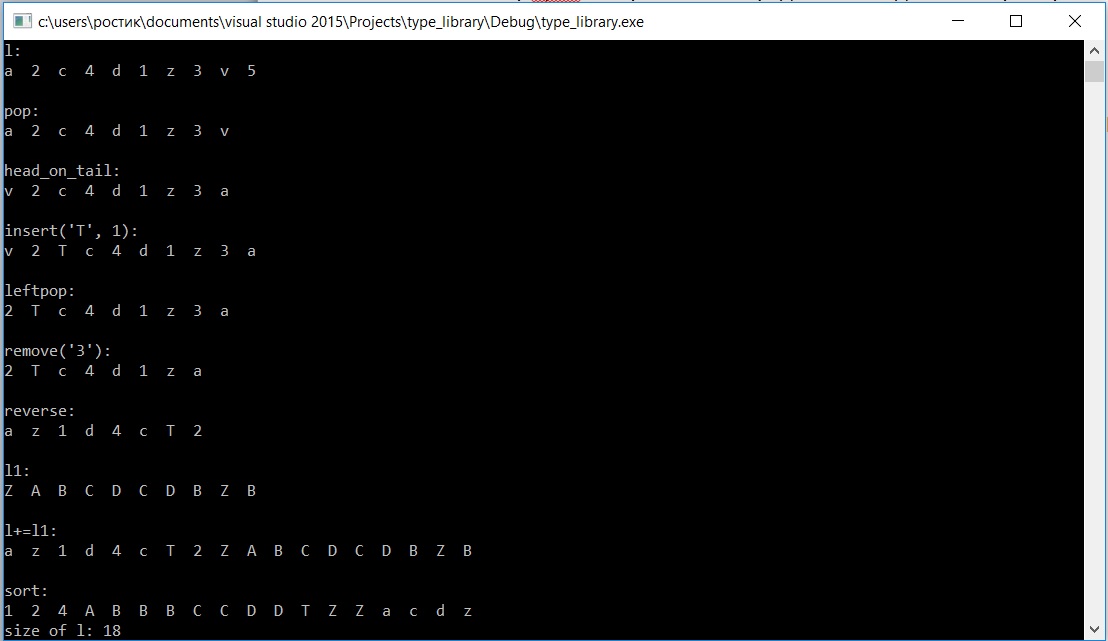
* **ПРЕЗЕНТАЦІЯ РЕЗУЛЬТАТІВ**

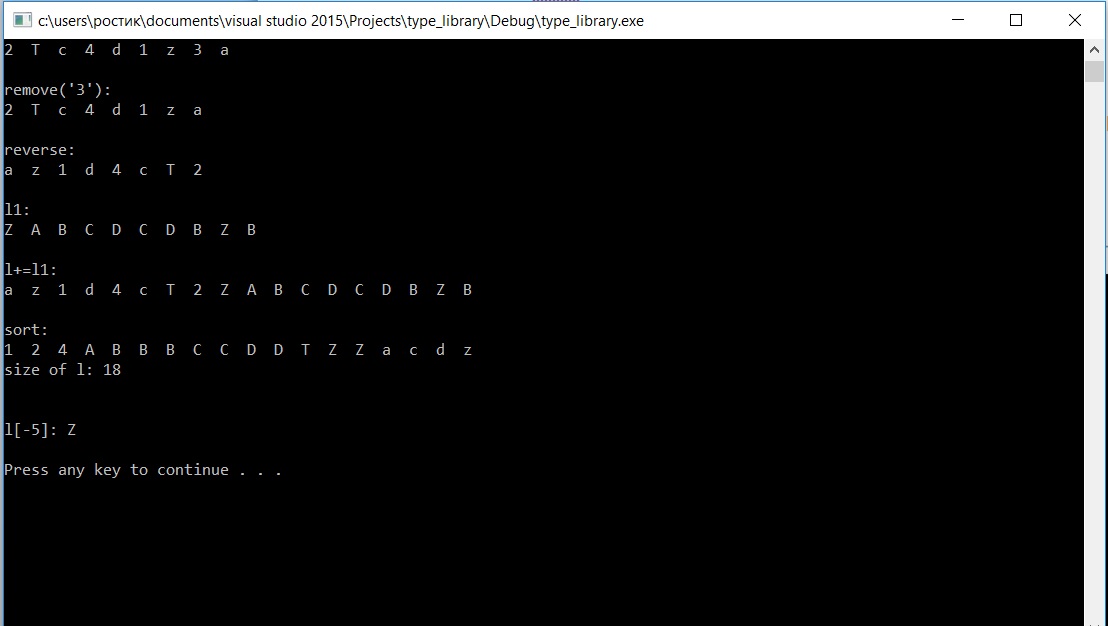
Приклад реалізації програми.

Створений однозв’язний список ***l*** і продемонстровані всі операції, які можна виконувати із цим списком.

Також створений ще один список ***l1*** для представлення функції додавання 2 списків.



****

****

* **СПИСОК УЧАСНИКІВ ТА ЇХ ВНЕСКИ В РЕАЛІЗАЦІЮ ПРОГРАМИ**

**Лозинський Степан** реалізував роботу із чергами.

**Стахів Юлія** реалізувала роботу із стеками.

**Настосяк Віталій** реалізував роботу із одно- та двозв’язними списками і звів програму до робочого стану.