Практикум по информатике: 8 факультет, 1 курс, 2 семестр 2011/12 уч. года.
Лабораторная работа №26 по курсам «Языки и методы программирования»/ «Алгоритмы и структуры данных»
Абстрактные типы данных. Рекурсия. Модульное программирование на языке Си

Составить и отладить модуль определений и модуль реализации по заданной схеме модуля определений для абстрактного (пользовательского) типа данных (стека, очереди, списка или дека, в зависимости от варианта задания). Составить программный модуль, сортирующий экземпляр указанного абстрактного типа данных заданным методом, используя только операции, импортированные из модуля UUDT.

УКАЗАНИЯ:

- стек, очередь, список или дек отображаются на массив;
- в программе по возможности должна быть использована рекурсия;
- метод сортировки реализовать с использованием указанной вспомогательной процедуры;
- использование итераторов для навигации по сериальным структурам приветствуется!

Схема модуля определений (UDT означает Стек, Очередь, Список или Дек, в соответствии с вариантом задания):

```
#ifndef _UDT_H_
#define _UDT_H_

#include <stdbool.h>

typedef struct {
  key_type key;
  value_type value;
} data_type;

typedef struct { ... } udt;

void udt_create(udt *);
bool udt_is_empty(const udt *);
void udt_push_front(udt *);
void udt_push_back(udt *);
void udt_pop_front(udt *);
void udt_pop_back(udt *);
void udt_print(const udt *);
size_t udt_size(const udt *);
void udt_insert(udt *, const data_type);
void udt_erase(udt *, const key_type);

#endif
```

Префикс UDT — простое средство от потенциального конфликта имен. В C++ для этой цели используются пространства имен.

Вариант задания определяется номером студента N по списку в группе: номер АТД равен $(N + 1) \% 4 + 1$:
1. Стек. 2. Очередь. 3. Дек. 4. Линейный список.

Номер процедуры и метода определяется как $(N + 1)$ **mod** $6 + 1$:

1. Процедура: поиск и удаление максимального (для стека, дека, списка) или минимального (для очереди) элемента
   Метод: сортировка линейным выбором
2. Процедура: Вставка элемента в стек, дек, список или очередь, упорядоченные по возрастанию, с сохранением порядка
   Метод: сортировка простой вставкой
3. Процедура: конкатенация двух стеков, деков, списков или очередей
   Метод: быстрая сортировка Хоара
4. Процедура: поиск в очереди, списке, стеке или деке двух элементов, идущих подряд, первый из которых больше второго. Если такие элементы найдены, их перестановка
   Метод: сортировка методом пузырька
5. Процедура: слияние двух стеков, деков, списков или очередей, упорядоченных по возрастанию, с сохранением порядка
   Метод: сортировка слиянием
6. Процедура: поиск в очереди, списке, стеке или деке первого от начала элемента, который меньше своего непосредственного предшественника. Если такой элемент найден, смещение его к началу до тех пор, пока он не станет первым или больше своего предшественника
   Метод: вариант метода вставки

Примечание. АТД, метод сортировки и вспомогательная процедура должны быть согласованы!