

Задание 1.

Напишите программу, реализующую сортировку массива целых чисел методом пузырька и выводящую результат на экран. Размер массива можно указать через `#define N`, ввод массива можно реализовать с клавиатуры или инициализовать в коде.

Сортировка пузырьком

Алгоритм состоит из повторяющихся проходов по сортируемому массиву. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный, выполняется обмен элементов. Проходы по массиву повторяются раз или до тех пор, пока на очередном проходе не окажется, что обмены больше не нужны, что означает — массив отсортирован. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции, как пузырёк в воде, отсюда и название алгоритма).

Задание 2.

Напишите библиотеку, осуществляющую работу со структурой данных — очередь. В очереди можно хранить целочисленные значения. Библиотека должна собираться при помощи утилиты `make`, при этом должна быть возможность указать с помощью правил статическую или динамическую сборку.

Возможный интерфейс:

```
struct fifo;           //Структура для работы с очередью
struct fifo_data;      //Структура данных очереди

//Создать стек
struct fifo* create_fifo();
//Добавить элемент в конец
void queue(struct fifo* fifo_ptr, struct fifo_data *data);
//Достать первый элемент с удалением
struct fifo_data* dequeue(struct fifo *fifo_ptr);
//Получить первый элемент без удаления
struct fifo_data* first(struct fifo *fifo_ptr);
//Удалить очередь вместе с данными
void remove_fifo(struct fifo* fifo_ptr);
```

Очередь — абстрактный тип данных с дисциплиной доступа к элементам «первый пришёл — первый вышел» (FIFO, First In — First Out). Добавление элемента (принято обозначать словом `enqueue` — поставить в очередь) возможно лишь в конец очереди, выборка — только из начала очереди (что принято называть словом `dequeue` — убрать из очереди), при этом выбранный элемент из очереди удаляется.

