

## Задание 1.

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

Шаги алгоритма:

1. находим номер минимального значения в текущем списке
2. производим обмен этого значения со значением первой неотсортированной позиции (обмен не нужен, если минимальный элемент уже находится на данной позиции)
3. теперь сортируем хвост списка, исключив из рассмотрения уже отсортированные элементы

## Задание 2.

Напишите библиотеку, осуществляющую работу со структурой данных — стек. В стеке 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 stack;           //Структура для работы со стеком
struct stack_data;      //Структура данных стека

//Создать стек
struct stack* create_stack();
//Добавить элемент в конец
void push(struct stack* stack_ptr, struct stack_data *data);
//Достать первый элемент с удалением
struct stack_data* pop(struct stack* stack_ptr);
//Получить первый элемент без удаления
struct stack_data* first(struct stack* stack_ptr);
//Удалить стек вместе с данными
void remove_stack(struct stack* stack_ptr);
```

**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»).

Чаще всего принцип работы стека сравнивают со стопкой тарелок: чтобы взять вторую сверху, нужно снять верхнюю.

