# Задание по программированию: Генератор компараторов

Условие

Давайте представим, что вы разрабатываете инновационный сервис поиска авиабилетов AviaScanner. В вашем сервисе авиабилет представляется в виде структуры

struct Date {

  int year, month, day;

};

struct Time {

  int hours, minutes;

};

struct AirlineTicket {

  string from;

  string to;

  string airline;

  Date departure\_date;

  Time departure\_time;

  Date arrival\_date;

  Time arrival\_time;

  uint64\_t price;

};

В данный момент вы работаете над функцией сортировки результатов поиска. Пользователь вводит свой запрос и получает список подходящих билетов. Дальше он может задавать параметры сортировки этого списка. Например, сначала по цене, затем по времени вылета и, наконец, по аэропорту прилёта.

void SortTickets(vector<AirlineTicket>& tixs) {

  stable\_sort(begin(tixs), end(tixs), [](const AirlineTicket& lhs, const AirlineTicket& rhs) {

    return lhs.to < rhs.to;

  });

  stable\_sort(begin(tixs), end(tixs), [](const AirlineTicket& lhs, const AirlineTicket& rhs) {

    return lhs.departure\_time < rhs.departure\_time;

  });

  stable\_sort(begin(tixs), end(tixs), [](const AirlineTicket& lhs, const AirlineTicket& rhs) {

    return lhs.price < rhs.price;

  });

}

Как видите, в примере выше есть дублирование кода — нам пришлось написать три лямбда-функции, которые отличаются только полем, по которому выполняется сортировка. От этого дублирования можно избавиться, написав макрос SORT\_BY и применив его следующим образом:

void SortTickets(vector<AirlineTicket>& tixs) {

  stable\_sort(begin(tixs), end(tixs), SORT\_BY(to));

  stable\_sort(begin(tixs), end(tixs), SORT\_BY(departure\_time));

  stable\_sort(begin(tixs), end(tixs), SORT\_BY(price));

}

Напишите макрос SORT\_BY, который принимает в качестве параметра имя поля структуры AirlineTicket. Вызов sort(begin(tixs), end(tixs), SORT\_BY(some\_field)) должен приводить к сортировке вектора tixs по полю some\_field.

Вам дан файл airline\_ticket.h, содержащий объявления структур Time, Date и AirlineTicket, а также заготовка решения в виде cpp-файла sort\_by.cpp. Пришлите на проверку cpp-файл, который

* подключает заголовочный файл airline\_ticket.h
* содержит макрос SORT\_BY
* содержит определения операторов, необходимых для использования классов Date и Time в алгоритме сортировки и макросе ASSERT\_EQUAL (формат вывода в поток можете выбрать произвольный)

[airline\_ticket.h](https://d3c33hcgiwev3.cloudfront.net/fPFKHWAuEeiEZRKxXgWFpg_7d46e380602e11e88d73c38a9a838951_airline_ticket.h?Expires=1617062400&Signature=HSemr3pRKkTyz3aruoaZs99baDvYj2bFCUw4l-WzoqdcZ8PGWDX2je0SQAU6ohc3WoVOHufo2s3lvm3zENJNDi~genELBlLTe4VReks0otn8uJSr9ln85hMhyEU5anjKPHB3o8kW1pu-SbASjx1icIsrkLjWBxvFf2rfDKn20Bg_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A" \t "_blank)

[sort\_by.cpp](https://d3c33hcgiwev3.cloudfront.net/fPA4gWAuEeiKNQ6MFIim6g_7d4535d0602e11e8a0a4333adb2f82d0_sort_by.cpp?Expires=1617062400&Signature=CMyxYLoZptuzMPdhYmFNiNB3MhlR56WPoE~QnOjT~y3t4DmxxToQrEjTgFp7FOkeSd1FL8nHmDWTxiIvrGyJ5zkT20r9uKAgjOrHuTE~KMqWLhSBX5ildktIpldV3pPapIypdd53rJyG0sZ7WNwysFbR18b8pnGbA6-jrGPOPKc_&Key-Pair-Id=APKAJLTNE6QMUY6HBC5A)