

# 7. SNTP

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Для того чтобы компьютеры поддерживали актуальное время, они могут обращаться к серверам точного времени SNTP (Simple Network Time Protocol). К сожалению, компьютер не может просто получить время у сервера, потому что информация по сети передаётся не мгновенно: пока сообщение с текущим временем дойдёт до компьютера, оно потеряет свою актуальность. Протокол взаимодействия клиента (компьютера, запрашивающего точное время) и сервера (компьютера, выдающего точное время) выглядит следующим образом:

1. Клиент отправляет запрос на сервер и запоминает время отправления A (по клиентскому времени).

2. Сервер получает запрос в момент времени B (по точному серверному времени) и отправляет клиенту сообщение, содержащее время B.

3. Клиент получает ответ на свой запрос в момент времени C (по клиентскому времени) и запоминает его. Теперь клиент, из предположения, что сетевые задержки при передаче сообщений от клиента серверу и от сервера клиенту одинаковы, может определить и установить себе точное время, используя известные значения A, B, C.

Вам предстоит реализовать алгоритм, с точностью до секунды определяющий точное время для установки на клиенте по известным A, B и C. При необходимости округлите результат до целого числа секунд по правилам арифметики (в меньшую сторону, если дробная часть числа меньше 1/2, иначе в большую сторону).

Возможно, что, пока клиент ожидал ответа, по клиентскому времени успели наступить новые сутки, однако известно, что между отправкой клиентом запроса и получением ответа от сервера прошло менее 24 часов.

## Формат ввода

Программа получает на вход три временные метки A, B, C, по одной в каждой строке. Все временные метки представлены в формате «hh:mm:ss», где «hh» – это часы, «mm» – минуты, «ss» – секунды. Часы, минуты и секунды записываются ровно двумя цифрами каждое (возможно, с дополнительными нулями в начале числа).

## Формат вывода

Программа должна вывести одну временную метку в формате, описанном во входных данных, – вычисленное точное время для установки на клиенте. В выводе не должно быть пробелов, пустых строк в начале вывода.

## Пример

<b>Ввод</b> <input type="text"/>	<b>Вывод</b> <input type="text"/>
15:01:00	18:10:05
18:09:45	
15:01:40	

Язык 

GNU GCC 12.2 C++20

Набрать здесь

Отправить файл

```
1 #include <cstdlib>
2 #include <iostream>
3 #include <iomanip>
4
5 struct Data {
6     int hh_ = 0;
7     int mm_ = 0;
8     int ss_ = 0;
9
10    void print() {
11        std::cout << std::setfill('0') << std::setw(2) << hh_
12            << ":" << std::setw(2) << mm_ << ":" << std::setw(2) << ss_;
13    }
14
15    Data(int hh, int mm, int ss) : hh_(hh), mm_(mm), ss_(ss) {};
16
17    Data(const std::string& s) {
18        hh_ = std::atoi(&s.begin());
19        mm_ = std::atoi(&s.begin()+3);
```

```
20         ss_ = std::atoi(&*s.begin()+6);
21     }
22
23     ~Data(){};
24
25     Data operator-(const Data& rhs) const{
26         int t_hh = hh_ - rhs.hh_;
27         int t_mm = mm_ - rhs.mm_;
28         int t_ss = ss_ - rhs.ss_;
29         if (t_ss < 0) {
30             t_ss+=60;
31             t_mm--;
32         }
33         if (t_mm < 0) {
34             t_mm+=60;
35             t_hh--;
36         }
37         if (t_hh < 0) {
38             t_hh+=24;
```

[Отправить](#)[Предыдущая](#)[Следующая](#)