

# ВШЭ АиСД 2021. Строки

4 дек 2021, 19:00:32

старт: 23 сен 2021, 11:30:00

финиш: 3 окт 2021, 23:59:59

длительность: 10д. 12ч.

начало: 23 сен 2021, 11:30:00

конец: 3 окт 2021, 23:59:59

## D. Таймер (0.2)

Ограничение времени	1 секунда
Ограничение памяти	64.0 Мб
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Таймер - это часы, которые умеют подавать звуковой сигнал по прошествии некоторого периода времени. Напишите программу, которая определяет, когда должен быть подан звуковой сигнал.

### Формат ввода

В первой строке входного файла записано текущее время в формате ЧЧ:ММ:СС (с ведущими нулями). При этом оно удовлетворяет ограничениям:  $00 \leq \text{ЧЧ} \leq 23$ ,  $00 \leq \text{ММ} \leq 60$ ,  $00 \leq \text{СС} \leq 60$ .

Во второй строке записан интервал времени, который должен быть измерен. Интервал записывается в формате Ч:М:С (где Ч, М и С - от 0 до  $10^9$ , без ведущих нулей).

Дополнительно, если Ч = 0 (или Ч = 0 и М = 0), то они могут быть опущены. Например, 100:60 на самом деле означает 100 минут 60 секунд, что то же самое, что 101:0 или 1:41:0. В таком формате записи единственное число 42 обозначает 42 секунды. Запись 100:100:100 - это 100 часов, 100 минут, 100 секунд, что то же самое, что и 101:41:40.

### Формат вывода

В выходной файл выведите в формате ЧЧ:ММ:СС время, во сколько прозвучит звуковой сигнал. При этом если сигнал прозвучит не в текущие сутки, то дальше должна следовать запись «<кол-во добавленных дней> days».

Например, если сигнал прозвучит уже на следующий день, то необходимо вывести «+1 days».

#### Пример 1

Ввод	Вывод
23:60:60	00:01:00+1 days
0	

#### Пример 2

Ввод	Вывод
05:05:05	05:10:06
5:1	

### Примечания

Реализуйте вычисление времени срабатывания сигнала таймера в виде **отдельной функции**, которая возвращает **строку**.

```
1 #include <iostream>
2 #include <string>
3
4 std::string PrintTimer(const int64_t &hoursStart, const int64_t &minutesStart,
5                       const int64_t &secondsStart, const int64_t &counterDays) {
6     std::string timer;
7     if (hoursStart == 0) {
8         timer += "00";
9     } else if (hoursStart > 0 && hoursStart < 10) {
10        timer += "0" + std::to_string(hoursStart);
11    } else {
12        timer += std::to_string(hoursStart);
13    }
14
15    timer += ":";
16
17    if (minutesStart == 0) {
18        timer += "00";
19    } else if (minutesStart > 0 && minutesStart < 10) {
20        timer += "0" + std::to_string(minutesStart);
21    } else {
22        timer += std::to_string(minutesStart);
23    }
24
25    timer += ":";
26
27    if (secondsStart == 0) {
28        timer += "00";
29    } else if (secondsStart > 0 && secondsStart < 10) {
30        timer += "0" + std::to_string(secondsStart);
31    } else {
32        timer += std::to_string(secondsStart);
33    }
34
35    if (counterDays > 0) {
36        timer += "+" + std::to_string(counterDays) + " days";
37    }
38 }
```