

# Тренировочный контест — бэкенд

🕒 16 янв 2024, 02:00:03  
старт: 11 янв 2024, 20:29:48  
...

Объявления жюри

Завершить

Задачи    **Посылки**    Сообщения

## В. Через тернии к клиенту

	Все языки	Clang 16.0.0 C++20	GNU GCC 12.2 C++20
Ограничение времени	5 секунд	1 секунда	1 секунда
Ограничение памяти	512Mb	512Mb	512Mb
Ввод	стандартный ввод или input.txt		
Вывод	стандартный вывод или output.txt		

- ✓

A. Хитрый шифр
- ✓

**В. Через тернии к клиенту**
- ✓

С. Приснится же такое...
- ✓

D. Лей, лей, не жалей
- ✓

E. Занимательная алхимия

Известная компания Тындекс идёт в ногу со временем — с началом активных космических перелётов в компании открылся сервис Тындекс.Ракетакси: пользователю необходимо лишь указать координаты начала и конца перелёта, после чего за ним вылетит персональная ракета.

По сути любой заказ можно описать в виде событий четырёх типов:

- A** (accepted) - заказ принят в работу (ракета вылетела за клиентом);
- B** (boarding) - клиент сел в ракету;
- S** (success) - заказ успешно завершён (клиент вышел на планете назначения);
- C** (cancelled) - заказ отменён.

Все происходящие с ракетами события отправляются на сервера, где сразу логируются. Вот только из-за проблем со связью (метеоритные потоки, вспышки на звездах и т.д.) отправка событий иногда затягивается, из-за чего записи в получившемся логе могут идти не по порядку.

Гарантируется, что все описанные в логе события задают один из следующих сценариев:

- A - B - S**
- A - B - C**
- A - C**

Вам, как главному аналитику (и по совместительству главному программисту) ракетопарка, необходимо исследовать лог за прошедший год и определить для каждой ракеты суммарное время движения (в минутах) в течение заказов.

В каждый момент времени ракета выполняет только один заказ. Будем считать, что каждая ракета в каждый момент времени:

- или стоит на месте в ожидании заказа,
- или перемещается по космосу, выполняя заказ.

Движение начинается после принятия заказа и завершается после отмены или завершения заказа. За одну минуту не может произойти несколько событий, связанных с одной и той же ракетой.

### Формат ввода

В первой строке дано целое число  $N(2 \leq N \leq 200\,000)$  — количество записей в логе.

Следующие  $N$  строк содержат записи в логе в формате *day hour minute id status*:

- day*( $1 \leq d \leq 365$ ) — номер дня (сквозная нумерация с начала календарного года);
- hour*( $0 \leq h < 24$ ) — часы;
- minute*( $0 \leq m < 60$ ) — минуты;
- id*( $0 \leq id \leq 10^9$ ) — уникальный идентификатор ракеты;
- status*  $\in \{\mathbf{A}, \mathbf{B}, \mathbf{S}, \mathbf{C}\}$  — буква, обозначающая тип события.

### Формат вывода

В единственной строке выведите через пробел суммарное время движения на заказах для каждой упомянутой в логе ракеты. Данные необходимо выводить в порядке возрастания идентификаторов ракет.

### Пример

Ввод 	Вывод 
8 50 7 25 3632 A 14 23 52 212372 S 15 0 5 3632 C 14 21 30 212372 A 50 7 26 3632 C 14 21 30 3632 A 14 21 40 212372 B 14 23 52 3632 B	156 142

### Примечания

#### Ракета №3632

- в 14-й день года в 21:30 получила заказ (шестая запись в логе);
- забрала пассажира в 23:52 того же дня (восьмая запись в логе);
- после чего заказ был отменён в 15-й день года в 00:05 (третья запись в логе);
- в 50-й день года в 7:25 получила заказа (первая запись в логе);
- заказ был отменён уже через минуту (четвёртая запись в логе).

Таким образом ракета №3632 провела в движении с 14-го дня 21:30 до 15-го дня 00:05 и с 50-го дня 7:25 до 50-го дня 7:26 — всего 156 минут.

#### Ракета №212372

- в 14-й день года в 21:30 получила заказ (третья запись в логе);
- через 10 минут забрала пассажира (седьмая запись в логе);
- в 23:52 прибыла на место назначения (вторая запись в логе).

Всего ракета №212372 провела в движении с 14-го дня 21:30 до 14-го дня 23:52 — 142 минуты.

Язык Python 3.9 (PyPy 7.3.11) ▾

Набрать здесь

Отправить файл

```
1 from collections import defaultdict
2
3 fin = open('input.txt')
4
5 N = int(fin.readline())
6
7 rockets = defaultdict(int)
8 for _ in range(N):
9     day, hour, minute, id, status = fin.readline().split()
10    day, hour, minute, id = [int(x) for x in (day, hour, minute, id)]
11
12    all_in_minute = day*24*60+hour*60+minute
13    if status == 'A':
14        rockets[id] += all_in_minute
15    elif status == 'S' or status == 'C':
16        rockets[id] -= all_in_minute
17
18 for key in sorted(rockets.keys()):
19     print(rockets[key], end=' ')
```

Отправить

🔢 осталось 99 попыток

Предыдущая

Следующая

Время посылки	ID	Задача	Компилятор	Вердикт	Тип посылки	Время	Память	Тест	Баллы
13 янв 2024, 18:54:13	104833081	В	Python 3.9 (PyPy 7.3.11)	ОК	-	0.546s	42.90Mb	-	-