

Тренировочный контест — бэкенд

[Задачи](#) [Посылки](#) [Сообщения](#)

B. Через терминал к клиенту

✓ Полное решение </> OK Go 1.23.0

^ Исходный код



```
1 package main
2
3 import (
4     "bufio"
5     "fmt"
6     "maps"
7     "os"
8     "slices"
9     "strconv"
10    "strings"
11 )
12
13 type Time struct {
14     day    int
15     hour   int
16     minute int
17 }
18
19 func (t *Time) add(day, hour, minute int) {
20     t.day += day
21     t.hour += hour
22     t.minute += minute
23 }
24
25 func (t *Time) sub(day, hour, minute int) {
26     t.day -= day
27     t.hour -= hour
28     t.minute -= minute
29 }
30
31 func (t *Time) countMinutes() int {
32     return (t.day*24+t.hour)*60 + t.minute
33 }
34
35 type LogEntry struct {
36     day    int
37     hour   int
38     minute int
39     id     int
40     letter string
41 }
42
43 func NewLogEntry(data []string) *LogEntry {
44     return &LogEntry{
45         day:    mustAtoi(data[0]),
46         hour:   mustAtoi(data[1]),
47         minute: mustAtoi(data[2]),
48         id:     mustAtoi(data[3]),
49         letter: data[4],
50     }
51 }
```

```

52
53 func mustAtoi(s string) int {
54     i, err := strconv.Atoi(s)
55     if err != nil {
56         panic(err)
57     }
58     return i
59 }
60
61 func main() {
62     file, err := os.Open("input.txt")
63     if err != nil {
64         panic(err)
65     }
66     defer file.Close()
67
68     scan := bufio.NewScanner(file)
69     scan.Scan()
70     rocketsTime := make(map[int]*Time, 0)
71
72     for scan.Scan() {
73         line := scan.Text()
74         log := NewLogEntry(strings.Fields(line))
75
76         if log.letter == "B" {
77             continue
78         }
79
80         r, exists := rocketsTime[log.id]
81
82         if !exists {
83             rocketsTime[log.id] = new(Time)
84             r = rocketsTime[log.id]
85         }
86
87         if log.letter == "A" {
88             r.sub(log.day, log.hour, log.minute)
89         } else {
90             r.add(log.day, log.hour, log.minute)
91         }
92     }
93
94     sortedKeys := slices.Sorted(maps.Keys(rocketsTime))
95     for _, key := range sortedKeys {
96         fmt.Printf("%d ", rocketsTime[key].countMinutes())
97     }
98 }
99

```

▼ Отличия от предыдущей посылки

▼ Лог компиляции

№	Вердикт	Ресурсы	Баллы	
1	ok	3ms / 2.02Mb	—	▼
2	ok	3ms / 2.11Mb	—	
3	ok	3ms / 2.00Mb	—	
4	ok	3ms / 2.13Mb	—	
5	ok	3ms / 2.18Mb	—	
6	ok	3ms / 2.00Mb	—	
7	ok	3ms / 1.94Mb	—	

Nº	Вердикт	Ресурсы	Баллы
8	ok	3ms / 2.00Mb	—
9	ok	3ms / 2.00Mb	—
10	ok	3ms / 2.10Mb	—
11	ok	3ms / 1.99Mb	—
12	ok	3ms / 2.00Mb	—
13	ok	3ms / 2.00Mb	—
14	ok	3ms / 2.08Mb	—
15	ok	3ms / 2.06Mb	—
16	ok	3ms / 2.00Mb	—
17	ok	3ms / 1.99Mb	—
18	ok	3ms / 2.00Mb	—
19	ok	80ms / 4.90Mb	—
20	ok	80ms / 4.90Mb	—
21	ok	355ms / 17.41Mb	—
22	ok	82ms / 4.90Mb	—
23	ok	228ms / 13.67Mb	—
24	ok	82ms / 4.90Mb	—
25	ok	198ms / 8.90Mb	—
26	ok	82ms / 4.90Mb	—
27	ok	161ms / 5.90Mb	—
28	ok	84ms / 4.90Mb	—
29	ok	150ms / 5.80Mb	—
30	ok	84ms / 4.90Mb	—
31	ok	123ms / 5.29Mb	—
32	ok	85ms / 4.90Mb	—
33	ok	124ms / 9.51Mb	—
34	ok	85ms / 5.03Mb	—
35	ok	114ms / 5.16Mb	—
36	ok	85ms / 5.03Mb	—
37	ok	106ms / 5.03Mb	—
38	ok	85ms / 4.90Mb	—
39	ok	103ms / 5.03Mb	—
40	ok	87ms / 4.90Mb	—
41	ok	101ms / 5.03Mb	—

Nº	Вердикт	Ресурсы	Баллы
42	ok	85ms / 4.90Mb	—
43	ok	98ms / 5.03Mb	—
44	ok	85ms / 4.90Mb	—
45	ok	96ms / 5.00Mb	—
46	ok	86ms / 5.03Mb	—
47	ok	96ms / 4.90Mb	—
48	ok	87ms / 4.90Mb	—
49	ok	94ms / 5.03Mb	—
50	ok	86ms / 4.90Mb	—
51	ok	89ms / 4.90Mb	—
52	ok	85ms / 5.03Mb	—
53	ok	88ms / 4.90Mb	—
54	ok	85ms / 4.90Mb	—
55	ok	88ms / 4.90Mb	—
56	ok	86ms / 4.90Mb	—
57	ok	88ms / 4.90Mb	—
58	ok	87ms / 4.90Mb	—
59	ok	89ms / 5.29Mb	—