

paiza - C031 : 문제

paizabook는 친구가 게시 한 일기 등을 볼 수 있지만, 사용자의 거주 지역에 따라 표시되는 게시 시간을 수정해야 합니다. 예를 들어, 런던에 살고있는 사용자가 현지 시간 8:30에 게시 한 일기의 시간은 일본에 살고있는 사용자의 화면에서 17:30로 표시되어야 합니다. 이것은 일본 시간 런던 시간보다 그냥 9 시간 진행되기 때문입니다. 또한 시간은 00:00부터 23:59까지 24 시간 제로 표시한다고 합니다.

N 개의 도시의 차이를 나타내는 정보와 그 안에 있는 도시에 사는 사용자의 일기 게시 시간 (현지 시간)이 입력되면 각각의 도시에 사는 사용자의 화면에 표시되는 게시 시간을 출력하는 프로그램을 작성하십시오.

- 입력되는 값
 - 1 번째 줄 N : 도시의 수를 나타내는 정수
 - 이어지는 N행 가운데 i행($1 \leq i \leq N$)에는 i번째 도시의 이름을 나타내는 문자열 p_i , 세계표준시와 그 도시의 현지시각의 차이(단위:시)를 나타내는 정수 s_i 가 공백을 구분으로 하여 주어집니다. s_i 가 음수이면 그 절대 값만큼 시간이 지연되고있는 것을 나타냄.
 - 다음 행에는 글을 게시한 사용자의 도시의 이름을 나타내는 문자열 q와 글을 올린 시간의 게시 시간을 나타내는 문자열 t가 이 순서대로 공백으로 구분되어 주어짐
 - t는 0 부터 두 자리 숫자로 시간 = hh, 분 = mm로 hh : mm 형식으로 주어짐
 - 입력은 총 N + 2 행이며, 입력 마지막 줄의 마지막에 개행이 하나 들어갑니다.
 - 각각의 값은 문자열로 표준 입력으로부터 전달됩니다

조건

모든 테스트 케이스에서 다음의 조건을 충족합니다.

- $1 \leq N \leq 100$
- 각 i ($1 \leq i \leq N$)에 대해
 - p_i 는 영문 소문자로 구성된 문자열
 - $1 \leq (p_i \text{ 길이}) \leq 20$
 - $-12 \leq s_i \leq 14$
- 각 i, j ($1 \leq i, j \leq N$)에 대해, $i \neq j$ 때 $p_i \neq p_j$
- q는 p_1, p_2, \dots, p_N 중 하나와 일치하는 문자열
- t는 24 시간에 실제로 존재 시간을 나타내는

paiza - C031 : 문제

기대하는 출력

도시 q의 사용자가 시간 t에서 올린 게시물에 대해 각 사용자의 화면에 표시되는 게시 시간을 다음 형식으로 출력합니다.

d_1

d_2

...

d_N

- 기대하는 출력은 N 행으로 구성됩니다.

- 출력의 i 번째 줄 ($1 \leq i \leq N$)에 i 번째 도시에 사는 사용자가 봤을 때의 게시물 시간을 나타내는 문자열 d_i 필요시 0을 넣어 두 자리 숫자로 시간 = hh 분 = mm로 hh : mm 형식으로 출력합니다.

- N 라인의 마지막에 개행을 하나 넣고 불필요한 문자 빈 행을 포함하지 마십시오.

입력 예 1

6

tokyo 9

beijing 8

singapore 7

london 0

newyork -5

osaka 9

singapore 19:38

출력 예 1

21:38

20:38

19:38

12:38

07:38

21:38

paiza - C031 : 문제

입력 예 2

27

howland -12

samoa -11

hawaii -10

alaska -9

california -8

arizona -7

texas -6

massachusetts -5

santiago -4

brasilgia -3

greenland -2

verde -1

morocco 0

london 1

paris 2

athens 3

moscow 4

islamabad 5

astana 6

bangkok 7

hongkong 8

seoul 9

guam 10

kuril 11

southpole 12

nukualofa 13

tokelau 14

southpole 00:00

출력 예 2

00:00

01:00

02:00

03:00

04:00

05:00

06:00

07:00

08:00

09:00

10:00

11:00

12:00

13:00

14:00

15:00

16:00

17:00

18:00

19:00

20:00

21:00

22:00

23:00

00:00

01:00

02:00

paiza - C031 : 채점 결과

제출 코드 결과 상세

테스트 번호	입력 케이스 번호	심판 결과	실행 시간
테스트 1	케이스 1 (기본 데이터)	 통과	0.09 초
테스트 2	케이스 1 (기본 데이터)	 통과	0.09 초
테스트 3	케이스 1 (기본 데이터)	 통과	0.10 초
테스트 4	케이스 1 (기본 데이터)	 통과	0.09 초
테스트 5	케이스 1 (경계 값 데이터)	 통과	0.08 초
	사례 2 (기본 데이터)	 통과	0.09 초
테스트 6	케이스 1 (기본 데이터)	 통과	0.09 초
테스트 7	케이스 1 (기본 데이터)	 통과	0.09 초
테스트 8	케이스 1 (기본 데이터)	 통과	0.10 초
테스트 9	케이스 1 (기본 데이터)	 통과	0.09 초
테스트 10	케이스 1 (기본 데이터)	 통과	0.09 초

paiza - C031 : 코드

C031.java

```
3 import java.util.*;
4
5 public class C031 {
6
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9
10        // 입력 부분
11        int cityCount = Integer.parseInt(sc.nextLine());
12        List<String> inputCity = new ArrayList<>();
13        List<Integer> inputTime = new ArrayList<>();
14        // nextLine으로 받은 데이터를 공백을 구분자로 split해 List에 각각 저장
15        for(int i = 0; i < cityCount; i++) {
16            String[] input = sc.nextLine().split(" ");
17            inputCity.add(input[0]);
18            inputTime.add(Integer.parseInt(input[1]));
19        }
20        // 글을 게시한 유저의 도시 이름과 시간 데이터를 split으로 쪼개 str에 저장
21        String[] str = sc.nextLine().trim().split(" ");
22        // str[0]에 저장된 도시 이름을 postUserCityName에 저장
23        String postUserCityName = str[0];
24        String strLocalTime = str[1];
25
26        // inputCity data에서 postUserCityName과 같은 이름 찾기
27        int indexEqualCity = 0;
28        for(int i = 0; i < cityCount; i++) {
29            if(postUserCityName.equals(inputCity.get(i))) {
30                indexEqualCity = i;
31            }
32        }
33    }
34 }
```

paiza - C031 : 코드

C031.java

```
26 // inputCity data에서 postUserCityName과 같은 이름 찾기
27 int indexEqualCity = 0;
28 for(int i = 0; i < cityCount; i++) {
29     if(postUserCityName.equals(inputCity.get(i))) {
30         indexEqualCity = i;
31     }
32 }
33
34 int hourUTC = Integer.parseInt(strLocalTime.substring(0, 2))
35     - inputTime.get(indexEqualCity);
36 List<Integer> standardHour = new ArrayList<>();
37 for(int i = 0; i < cityCount; i++) {
38     if(inputTime.get(i) + hourUTC >= 24) { // 합이 24 이면 1일을 올리고 나머지를 가져온다
39         standardHour.add((inputTime.get(i) + hourUTC) % 24);
40     } else if(inputTime.get(i) + hourUTC < 0) {
41         standardHour.add(24 + inputTime.get(i) + hourUTC);
42     } else {
43         standardHour.add(inputTime.get(i) + hourUTC);
44     }
45 }
46 for(int i = 0; i < cityCount; i++) {
47     if(standardHour.get(i) < 10) {
48         System.out.print("0" + standardHour.get(i));
49     } else {
50         System.out.print(standardHour.get(i));
51     }
52     System.out.println(strLocalTime.substring(2));
53 }
54 }
55
56 }
```