

1

2

3

정확성 시간 제한 / 메모리 제한

10초 / 2GB

문제 설명

당신은 외출을 할 때, 항상 마스크를 착용합니다. 당신은 마스크 구매 비용을 아끼기 위해, 구매 가능한 마스크의 정보와, 외출할 가능성이 있는 날짜를 조사해보았습니다.

구매 가능한 마스크의 정보

가격	내구도
3200	4
2300	2
4200	6
1100	1

- 내구도는 마스크를 처음 사용한 날을 포함하여 며칠간 사용할 수 있는지를 의미합니다.
- 예를 들어, 내구도가 4인 마스크를 2022/07/30에 처음 사용했다면, 2022/08/02까지 사용 가능합니다.
 - 포장을 뜯은 마스크는 오염이 되기 시작하므로, 2022/07/31, 2022/08/01에 사용하지 않았어도 2022/08/02까지만 사용 가능합니다.
- 각각의 마스크는 종류별로 충분한 재고가 있다고 가정하므로, 무한대로 구매 가능합니다.

외출 가능성이 있는 날짜

- 2022/05/02, 2022/05/01, 2022/05/07, 2022/05/05, 2022/05/08, 2022/05/13~2022/05/15, 2022/05/14~2022/05/17, 2022/05/01~2022/05/02, 2022/05/16
 - 외출 가능성이 있는 날짜들끼리 서로 겹칠 수 있습니다.

위의 날짜들을 달력에 밑줄로 표기하면 다음과 같습니다.

2022년 5월

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21

아래와 같은 방법으로 마스크를 사용하면, 총 9600원이 필요하며, 이것이 외출할 가능성이 있는 모든 날에 마스크를 사용할 수 있도록 하기 위한 최소 비용입니다.

- 2022/05/01 : (1100원, 내구도 1) 마스크 사용
- 2022/05/02 : (1100원, 내구도 1) 마스크 사용
- 2022/05/05, 2022/05/07, 2022/05/08 : (3200원, 내구도 4) 마스크 사용
- 2022/05/13~2022/05/17 : (4200원, 내구도 6) 마스크 사용

마스크의 가격과 내구도를 담은 2차원 정수 배열 `masks` , 당신이 외출할 가능성이 있는 날짜들을 담은 문자열 배열 `dates` 가 매개변수로 주어집니다. 당신이 외출할 가능성이 있는 모든 날에 마스크를 사용할 수 있도록 마스크를 구매하는 최소 비용을 return 하도록 solution 함수를 완성해주세요.

제한사항

- 본 문제에서 윤년은 존재하지 않는다고 가정합니다. 따라서, 1년은 항상 365일입니다.
 - 1월, 3월, 5월, 7월, 8월, 10월, 12월은 31일이 마지막 날입니다.
 - 4월, 6월, 9월, 11월은 30일이 마지막 날입니다.
 - 2월은 28일이 마지막 날입니다.
- `masks` 의 길이는 1 이상 1,000 이하입니다.
 - `masks` 의 각 원소는 [가격, 내구도]입니다.
 - 1 ≤ 가격 ≤ 100,000
 - 1 ≤ 내구도 ≤ 2,000
- `dates` 의 길이는 1 이상 1,000 이하입니다.
 - `dates` 의 원소는 특정 날짜(년/월/일)를 나타내는 "YYYY/MM/DD" 또는 기간을 나타내는 "YYYY/MM/DD~YYYY/MM/DD" 형식입니다.
 - 따라서, `dates` 의 원소는 길이가 10 또는 21인 문자열입니다.
 - 2021 <= YYYY <= 2030
 - 즉, 2021년 01월 01일부터 2030년 12월 31일 사이의 날짜와 기간이 입력으로 주어집니다.
 - 다음의 예처럼, 날짜와 기간이 똑같거나 날짜와 기간이 서로 겹치는 경우의 입력도 주어질 수 있습니다.
 - 예) ["2023/01/01", "2023/01/01", "2023/01/01~2023/12/31", "2023/01/01~2023/12/31"]
 - 다음의 예처럼, 잘못된 날짜는 입력으로 주어지지 않습니다.
 - 예) ["2021/13/25" , "2021/04/31"]
 - 다음의 예처럼, 끝나는 날짜가 시작 날짜보다 같거나 이른 기간은 입력으로 주어지지 않습니다.
 - 예) ["2021/02/28~2021/02/27"]

입출력 예

masks	dates	result
[[3200, 4], [2300, 2], [1100, 1], [4200, 6]]	["2022/05/02", "2022/05/01", "2022/05/07", "2022/05/05", "2022/05/08", "2022/05/13~2022/05/15", "2022/05/14~2022/05/17", "2022/05/01~2022/05/02", "2022/05/16"]	9600
[[600, 2], [500, 1], [1015, 400]]	["2023/01/01~2023/01/02", "2021/12/31"]	1015
[[3651, 365], [10, 1]]	["2025/01/01~2025/12/31"]	3650

입출력 예 설명

입출력 예 #1

문제 예시와 같습니다.

입출력 예 #2

2021/12/31에 (1015원, 내구도 400) 마스크를 사용하면, 2023/01/02까지 사용할 수 있습니다.

입출력 예 #3

방법 1 : 2025/01/01에 (3651원, 내구도 365) 마스크를 사용하면, 2025/12/31까지 사용할 수 있습니다.

방법 2 : 365일 동안 매일 새로운 (10원, 내구도 1) 마스크를 사용할 수 있습니다. 총 구입 비용은 3650원이고, 이것이 최소 비용입니다.

테스트 케이스 추가하기

3. 프로그래밍

9분 전 저장됨

Java

```
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

int month = Integer.parseInt(dates[i].substring(5, 7));
int day = Integer.parseInt(dates[i].substring(8, 10));

int s = (year - 2021) * 365 + monthArr[month - 1] + day;
int e = s;

if(dates[i].length() > 10) {
    int endYear = Integer.parseInt(dates[i].substring(11, 15));
    int endMonth = Integer.parseInt(dates[i].substring(16, 18));
    int endDay = Integer.parseInt(dates[i].substring(19, 21));

    e = (endYear - 2021) * 365 + monthArr[endMonth - 1] + endDay;
}

schedule[s] += 1;
schedule[e + 1] -= 1;
}

return answer;
```

실행 결과

가장 값 : 1015

실행 결과 : 실행한 결과값 0이 가장 값 1015과 다릅니다.

테스트 3

입력값 : [[3651, 365], [10, 1]], ["2025/01/01~2025/12/31"]

가장 값 : 3650

실행 결과 : 실행한 결과값 0이 가장 값 3650과 다릅니다.

테스트 결과 (~")~

3개 중 0개 성공

내 답안

코드 초기화

코드 실행

제출 후 재입력하기