

일정관리 시스템

일정관리 시스템을 만들 계획이다.

기능은 일정추가, 수정, 삭제, 검색이 있다.

```
void Add(int id, char* title, char* location, char* start_date, char* end_date){ ... }
```

```
void Modify(int id, char* title, char* location, char* start_date, char* end_date){ ... }
```

```
void Delete(int id){ ... }
```

```
void Find(int search_type, char* str){ ... }
```

함수정의는 다음과 같이 하였다.

Add
일정 index : id 일정 제목 : title 장소 : location 일정 기간 : start_date에서 시작하여 end_date에 끝난다. Title과 Location은 최소 20자 소문자 알파벳으로 구성되어 있다 Date들은 YYYYMMDDHHMM 형식으로 년월시분으로 주어진다.

Modify
id 가 같은 일정을 찾아서 입력받은 정보로 수정한다.

Delete
주어진 id를 찾아서 일정을 삭제한다.

Find
Search_type의 값에 따라 검색 방법이 다르다. 0 : 제목이 str인 일정들을 id순으로 출력. 1 : 장소가 str인 일정들을 id순으로 출력.

2 : 일정이 str인 경우 id순으로 출력 (YYYYMMDD로 입력된다 시간 분이 없음을 유의)

다음 형식으로 출력

Id title location start_date end_date

예) 15 dinner restaurant 201611201230 201611201400

추가 사항:

ADD에서 Id는 **1부터 순서대로** 입력되며 범위는 **1이상 10000미만**입니다.

Add 또는 Mod시 title과 location이 같고 date가 겹치는 구간이 있을 경우 일정들을 합칩니다.

합쳐진 일정의 id는 합쳐진 일정들 중에 제일 작은 id로 결정됩니다.

1 abc edf 201708291300 201708291700

2 abc edf 201708282300 201708291400

는

1 abc edf 201708282300 201708291700

으로 합쳐져야 합니다.

경우에 따라 여러 개의 일정들이 합쳐지는 경우가 있음을 유의.

년도는 1970년부터 2100년까지 주어짐

Start_date에 시작하여 end_date전까지 일정이 있음.

[Start_date, end_date) : end_date는 구간에 정확히 포함되지 않음

끝나는 일이 12시 13분인 일정과 시작하는 일이 12시 14분인 일정은 연속하지 않음.

12시 13분 0초부터 59초까지 비어 있음

끝나는 일이 12시 14분인 일정과 시작하는 일이 12시 14분인 일정은 연속

제약 사항

50개 테스트케이스에 대해 5초안에 실행되어야 하고, 메모리 256MB 이하 사용해야함.

Add, Mod, Delete, Find 함수들은 5:1:1:1 비율로 주어짐

함수 호출은 4개 함수 모두 합쳐 최대 10000개

모순 되는 데이터는 없다고 가정

(삭제된 일정을 수정하는 경우 등)

데이터

1~10 데이터 개수가 작고 < 20 파악하기 쉬운 데이터

11~20 100개짜리 무작위 랜덤 데이터 title location은 최대 5개

21~30 1000개짜리 일정이 특정 년도에 집중 되어 있고 title location은 최대 10개

31~40 10000개짜리 Add와 Find만 있는 데이터 title location 최대 10000개

41~50 10000개짜리 무작위 데이터 title location 최대 100개

Input :

가장 처음에 명령어 개수 N개를 입력받는다

그 이후 9개의 명령이 입력된다.

Outout :

find할 경우 해당하는 일정을 id순서대로 출력한다.

input	output
9	1 abc def 201708291300 201708291700
0 1 abc def 201708291300 201708291700	1 abc def 201708291300 201708291700
0 2 efgd ert 201705291300 201706291700	3 abc def 201712111300 201712301700
3 0 abc	1 abc def 201707110800 201712301700
0 3 abc def 201712111300 201712301700	1 abc def 201707110800 201712301700
3 1 def	2 efgd ert 201705291300 201706291700
0 4 abc def 201707110800 201712151700	
3 0 abc	
3 1 def	
3 2 20170601	

Base Source

```
#include <stdio.h>

void Add(int id, char* title, char* location, char* start_date, char* end_date) {
}

void Modify(int id, char* title, char* location, char* start_date, char* end_date) {
}

void Delete(int id) {
}

void Find(int search_type, char* str) {
}

int main() {
    int inputType;
    char title[256];
    char location[256];
    char start_date[256];
    char end_date[256];
    char str[256];
    int id, searchType;
    int N;
```

```

//freopen("input.txt", "r", stdin);

scanf("%d", &N); //input data 갯수
for (int i = 0; i < N; i++) {
    scanf("%d", &inputType);
    if (inputType == 0) {

        scanf("%d", &id);
        scanf("%s", title);
        scanf("%s", location);
        scanf("%s", start_date);
        scanf("%s", end_date);
        Add(id, title, location, start_date, end_date);
    }
    else if (inputType == 1) {
        scanf("%d", &id);
        Delete(id);
    }
    else if (inputType == 2) {
        scanf("%d", &id);
        scanf("%s", title);
        scanf("%s", location);
        scanf("%s", start_date);
        scanf("%s", end_date);
        Modify(id, title, location, start_date, end_date);
    }
    else if (inputType == 3) {
        scanf("%d", &searchType);
        scanf("%s", str);
        Find(searchType, str);
    }
}
return 0;
}

```