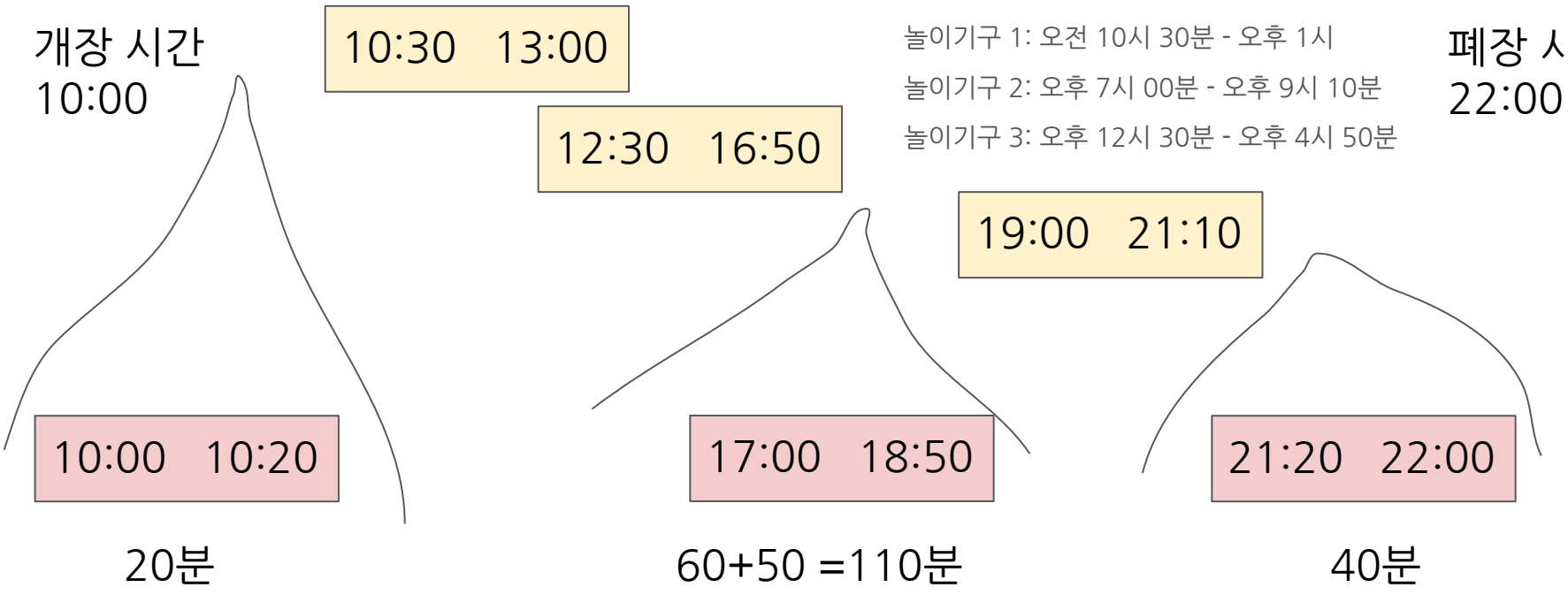
그렇게
됐습니다

2594: 놀이공원

놀이공원에서 여러 개의 놀이기구를 맡아 일하는 세혁이와 근영이는 서로 좋아하는 사이이다. 그들은 쉬는 시간을 이용하여 둘만의 시간을 가지기를 원한다. 그래서 매일 일과 시작 전에 놀이기구의 운영 일정을 보고, 그날 둘이 함께할 수 있는 가장 긴 휴식시간이 언제인지를 찾으려고 한다.

놀이공원에서 일하는 모든 사람들은 어떤 놀이기구가 작동을 시작하기 10분 전부터, 모든 놀이기구가 작동을 멈춘 후 10분 후까지는 쉴 수 없고, 그 나머지 일과 시간에만 쉴 수 있다.



2594: 놀이공원


```
public static class Ride{  3 usages
    int startTime;
    int endTime;

    public Ride(int startTime, int endTime) {  1 usage
        this.startTime = startTime;
        this.endTime = endTime;
    }
}
```

놀이기구 시간을 리스트에 넣을 때 시작 시간 -10, 종료시간 +10 해서 넣고
10시부터 쉬는 시간 체크하면 됨

```
// hhmm -> min 으로 바꿔서 리스트에 넣기
start = toMin(start) - 10;
end = toMin(end) + 10;
rides.add(new Ride(start, end));
```

```
// 정렬 시작시간이 빠른 순, 시작시간이 같으면 끝 시간이 빠른 순
rides.sort(( Ride o1,  Ride o2) -> {
    if(o1.startTime==o2.startTime){
        return Integer.compare(o1.endTime, o2.endTime);
    }
    return Integer.compare(o1.startTime, o2.startTime);
});
```

람다로 정렬하기

2594: 놀이공원


```java
int prevEnd = 600; // 10시 == 600(60*10)
int answer = 0; // 가장 긴 쉬는 시간

for (int i = 0; i < n; i++) {
    Ride curr = rides.get(i);

    if(curr.startTime > prevEnd){
        answer = Math.max(answer, curr.startTime-prevEnd);
    }

    // 구간 겹침 방지
    prevEnd = Math.max(prevEnd, curr.endTime);
}

// 폐장 시간 22시(22*60 = 1320)
if(prevEnd<1320){
    answer = Math.max(answer, 1320-prevEnd);
}
System.out.println(answer);
```

그냥 전부 분(min)으로 바꿔서 넣고, 분으로 계산함.
(10분 단위 boolean 배열 만들어서 할 수도 있을듯)
prevEnd는 마지막으로 계산한 놀이기구의 끝나는 시간
-> 현재 시간- 마지막 놀이기구 시간을 빼면 쉬는 시간 나옴

개장 시간
10:00
600

폐장 시간
22:00
1320

마지막 계산 놀이기구 끝나는 시간
ex)650

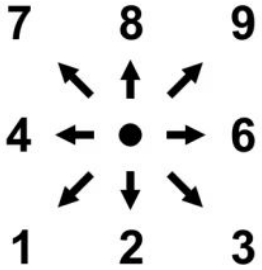
지금 놀이기구 시작 시간
ex)1000

리스트에 넣을 때 시작시간-10, 종료시간+10 해줬어서 그냥 빼주면 됨
위에 예시는 1000-650 = 350분 쉬는 시간이겠지

8972: 미친 아두이노

.:빈칸, R: 미친 아두이노, I: 종수
1. 종수가 움직이고 싶은 방향으로 움직인다(입력으로 주어짐)
  - 종수의 아두이노가 미친 아두이노 칸으로 이동하면 패
2. 미친 아두이노는 8방향 중 종수의 아두이노와 가장 가까워지는 방향으로 한 칸 이동
  - 미친 아두이노가 종수의 아두이노 칸으로 이동하면 패배
  - 2개 이상의 미친 아두이노가 같은 칸으로 이동하면 해당 칸의 아두이노는 모두 파괴

주어진 방향대로 움직였을 때, 보드 상태(패배 시 몇 번 움직였는지 출력)



예제 입력 1 복사

```
4 5
I....
.....
.R.R.
.....
6
```

맵크기 r c

맵 .:빈칸, R: 미친 아두이노, I: 종수

종수 이동방향 순서

예제 출력 1 복사

```
.I...
.RR..
.....
.....
```

이동 다 끝내면
맵 상태 출력

예제 입력 3 복사

```
12 8
...I....
........
........
........
........
RR......
......RR
R......R
........
........
........
...R....
66445394444162
```

예제 출력 3 복사

```
kraj 11
```

중간에 끝나면 이렇게
“kraj 이동횟수” 출력

8972: 미친 아두이노


```java
// 초기 보드 입력 및 위치 파싱
for (int i = 0; i < r; i++) {
    board[i] = br.readLine().toCharArray();
    for (int j = 0; j < c; j++) {
        if (board[i][j] == 'I') {
            player = new Coord(i, j);
        } else if (board[i][j] == 'R') {
            enemies.add(new Coord(i, j));
        }
    }
}

String command = br.readLine();
for (int i = 0; i < command.length(); i++) {
    int dir = command.charAt(i) - '0'; // 1 ~ 9

    // 1. 종수 이동
    if (!movePlayer(dir)) {
        System.out.println("kraj " + (i + 1));
        return;
    }

    // 2. 미친 아두이노 이동
    if (!moveEnemies()) {
        System.out.println("kraj " + (i + 1));
        return;
    }
}
```

입력 받으면서 적의 좌표를 리스트에 저장해주기(매번 적 위치를 배열 돌면서 찾을 수 없으니까)

종수 이동

적 이동

```java
private static boolean movePlayer(int dir) {  1 usage
    int nx = player.x + dx[dir];
    int ny = player.y + dy[dir];

    // 이동한 칸에 미친 아두이노가 있으면 패배
    if (board[nx][ny] == 'R') {
        return false;
    }

    // 기존 위치 비우고 새 위치로 이동
    board[player.x][player.y] = '.';
    player.x = nx;
    player.y = ny;
    board[player.x][player.y] = 'I';

    return true;
}
```

종수 이동은 걍 해주면 됨
이동하는 칸에 적이 있으면 그냥 이동 실패 false 반환

8972: 미친 아두이노


```java
String command = br.readLine();
for (int i = 0; i < command.length(); i++) {
    int dir = command.charAt(i) - '0'; // 1 ~ 9

    // 1. 종수 이동
    if (!movePlayer(dir)) {
        System.out.println("kraj " + (i + 1));
        return;
    }

    // 2. 미친 아두이노 이동
    if (!moveEnemies()) {
        System.out.println("kraj " + (i + 1));
        return;
    }
}
```

종수 이동

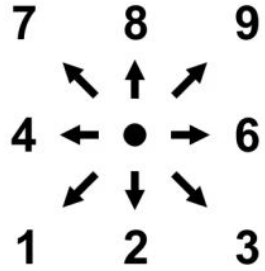
적 이동

```java
// 1. 각 미친 아두이노의 다음 위치 정하기
for (Coord enemy : enemies) {
    // 기존 자리 비우기 (새로 다시 채울 예정)
    board[enemy.x][enemy.y] = '.';

    int bestX = enemy.x;
    int bestY = enemy.y;
    int minDist = Integer.MAX_VALUE; // 종수랑 가장 가까운 거리의 좌표로 이동해야 함

    // 1~9 방향 중, 5(제자리)는 제외
    for (int dir = 1; dir <= 9; dir++) {
        if (dir == 5) continue;

        // 이동할 좌표
        int nx = enemy.x + dx[dir];
        int ny = enemy.y + dy[dir];

        // 배열 범위 밖
        if (nx < 0 || nx >= r || ny < 0 || ny >= c) continue;

        // 거리 계산(지문에서 주어진대로 계산함)
        int dist = Math.abs(nx - player.x) + Math.abs(ny - player.y);
        // 최단거리면 이동할 좌표 및 최단거리 갱신
        if (dist < minDist) {
            minDist = dist;
            bestX = nx;
            bestY = ny;
        }
    }
```

9개 이동 좌표마다 종수랑 거리 계산해주고 제일 가까운 좌표로 갱신하기

```java
    // 이동 결과가 종수 위치면
    if (bestX == player.x && bestY == player.y) {
        return false;
    }

    cnt[bestX][bestY]++; // 해당 칸으로 이동 예정인 미친 아두이노 수 증가
```

제일 가까운 좌표가 종수 좌표면 게임 종료
아니면 그 좌표 적 개수 증가



8972: 미친 아두이노


```java
String command = br.readLine();
for (int i = 0; i < command.length(); i++) {
    int dir = command.charAt(i) - '0'; // 1 ~ 9

    // 1. 종수 이동
    if (!movePlayer(dir)) {
        System.out.println("kraj " + (i + 1));
        return;
    }

    // 2. 미친 아두이노 이동
    if (!moveEnemies()) {
        System.out.println("kraj " + (i + 1));
        return;
    }
}
```

종수 이동

적 이동

```java
// 2. 폭발 처리 후 enemy 리스트 재구성
enemies.clear();

for (int x = 0; x < r; x++) {
    for (int y = 0; y < c; y++) {
        if (cnt[x][y] == 1) {
            enemies.add(new Coord(x, y));
        }
    }
}

// 보드에 반영
for (Coord e : enemies) {
    board[e.x][e.y] = 'R';
}

return true;
```

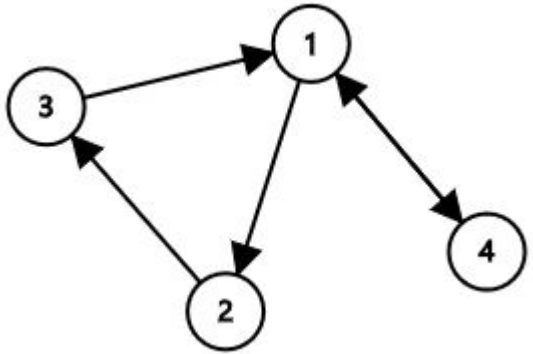
적 리스트 싹 다 초기화 해주고 적이 1명인 좌표만 적리스트에 넣어주기

적 리스트좌표 board에 갱신

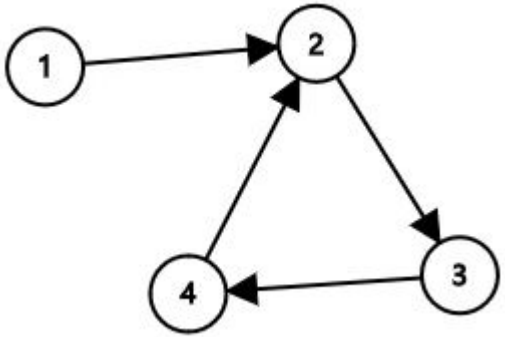
이동 성공

26146: 즉흥 여행(Easy)

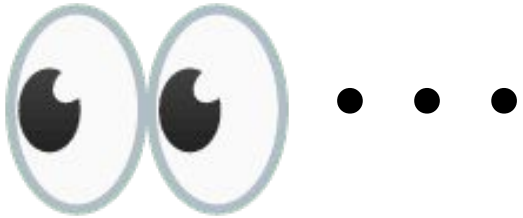
하나의 나라를 여러 번 방문할 수 있으며, 하나의 항공편을 여러 번 사용할 수 있다.
시작점을 어떻게 골라도 모든 나라를 방문할 수 있는 경로가 있다면 Yes를, 아니면 No를 출력



어디서 시작하든 모든 정점을
다 방문할 수 있음
Yes



2, 3, 4에서 시작했을 때 1로
가는 경로가 없으므로
No



제출 번호	아이디	문제	결과	메모리	시간	언어	코드 길이	제출한 시간
100255453	dalpiecel7	26146	맞았습니다!!	198892 KB	920 ms	Java 11	1939 B	1일 전
100255320	dalpiecel7	26146	틀렸습니다			Java 11	1825 B	1일 전
100254917	dalpiecel7	26146	메모리 초과			Java 11	1887 B	1일 전