

# 대회 안내

## 1. 대회 관련

- 각 문제를 해결할 때마다 ({대회 후 경과시간}+{해당 문제 오답 횟수 x20분})의 패널티가 가산됩니다.
- 순위는 문제를 많이 푼 순 > 패널티가 적은 순 > 마지막 정답이 빠른 순으로 정해집니다.
- 문의하기기능을 사용해 문제와 직접적으로 관련된 사항에 대해 질문할 수 있습니다.

## 2. 제출관련

- 제출시에 문제 번호와 사용 언어를 정확히 선택해주세요. 참가자의 실수에 대해 책임지지 않습니다.
- 제출한 코드는 채점 현황에서 확인할 수 있습니다.

## 3. 채점 관련

- 표준 입출력 이외의 파일을 읽거나 쓰려고 시도하는 경우 런타임 에러로 처리하여 오답으로 인정합니다.
- 병렬연산이나 스레드를 이용하면 안됩니다. 모든 프로그램은 단일 스레드 프로그램이어야 합니다.
- 시스템에 영향을 주는 기능(종료, sleep 등)을 사용하면 무조건 오답처리 합니다.
- 채점 입력 데이터 모두에 대해 올바른 답을 출력하여야 합니다. 부분점수는 없습니다.
- 예제 데이터는 하나의 예제 일 뿐입니다. 실제 채점은 수 많은 비공개 데이터들로 이루어집니다.
- 각 채점 결과의 의미는 아래와 같습니다. 감독관은 오답의 이유를 절대 알려드리지 않습니다.

채점 결과 메시지	의미
대기중	현재 다른 답안을 채점하는 중
컴파일중	채점을 위해 소스코드를 컴파일 하는 중
채점중	채점 서버에서 해당 답안을 채점하는 중
정답	제출한 답안이 정답으로 인정 된 경우
오답	제출한 답안이 틀린 데이터를 출력한 경우
시간 초과	제출한 프로그램이 제한된 시간내에 종료되지 않은 경우
메모리 초과	제출한 프로그램이 제한된 메모리를 초과하여 사용한 경우
컴파일 에러	제출한 코드에 문제가 있어서 컴파일 되지 않는 경우 <ul style="list-style-type: none"> <li>- 단 경고(Warning)은 무시합니다.</li> <li>- 채점 현황에서 소스코드를 확인하면 컴파일 결과를 알려드립니다.</li> </ul>
런타임 에러	제출한 프로그램이 실행도중에 문제를 일으킨 경우 <ul style="list-style-type: none"> <li>- Segmentation Faults</li> <li>- 병렬 연산, 멀티 쓰레딩, 시스템 콜 사용 등</li> <li>- 파일 입출력 시도</li> </ul>

## 4. C/C++ 관련

- 모든 입출력은 stdin, stdout을 통한 표준 입출력만 사용됩니다.
- GCC 컴파일러를 통해 채점됩니다. 비표준 MS C/C++ 라이브러리를 사용할 경우 컴파일 에러가 발생할 수 있습니다.
- C++의 cin/cout은 scanf/printf에 비하여 속도가 현저히 느립니다. 채점 시간에 반영되므로 후자의 이용을 권장합니다.

## 5. Java 관련

- 답안의 클래스명은 Main.java이어야 합니다. 또한, 소스코드에 Package 선언문을 절대로 넣지 마세요. 오답 처리됩니다.
- 모든 입출력은 System.in, System.out을 통한 표준 입출력만 사용됩니다.
- Java의 특성상 문자열 처리 시 StringBuilder를 활용하는 편이 속도면에서 매우 유리합니다.
- System.in 입력 시 Scanner보다 BufferedReader와 InputStreamReader를 활용하는 편이 훨씬 속도가 빠릅니다.
- Java는 언어와 구조의 특성상, 표기된 시간 제한이나 메모리 제한을 조금 넘겨도 인정합니다. (인정범위 : +1초, +128MB)

C/C++ 레퍼런스 사이트 : <http://en.cppreference.com/w/>

Python 레퍼런스 사이트 : <http://python.org/>

JAVA 레퍼런스 사이트 : <http://docs.oracle.com/javase/7/docs/>

# Problem A

## 트위스위트 페이트

시간 제한 : 1 Second

라이엇은 트위스위트 페이트(이하 트페)를 패치시켰다. 패치노트는 다음과 같다.

이번 패치에서는 트페를 조금 더 도박적인 챔피언으로 바꾸기로 결정했습니다. 그래서 W스킬인 카드를 뽑는 마나 코스트를 증가시키고 데미지 계산 방식을 바꿨습니다.

Red Card : 대상에게 1의 데미지를 입히고 주변 미니언에게 초당 레벨 x (0.1AP)의 데미지를 입힙니다.

Gold Card : 지금까지 입힌 데미지의 2배를 입힙니다.

Blue Card : 3의 데미지를 입히고 이동속도를 스킬 레벨당(30%, 40%, 50%, 60%, 70%)만큼 감소시킵니다.

트페 장인 종훈이는 패치된 트페를 정글로 플레이 해보았다. 변태같은 종훈이는 항상 모든 정글 몬스터를 정확히 몬스터의 체력만큼의 데미지를 주어서, 그것도 최소 스킬 사용으로 잡을 수 있는 사용 횟수를 다 외우고 있다. 예를 들어 몬스터의 체력이 12라고 하면 Blue, Blue, Gold Card순서로 사용하면 3회로 정확한 데미지이면서 최소 횟수로 잡을 수 있다.

이 모습을 본 세영이는 정말 그걸 다 외우고 있는지 궁금해서 이런 질문을 준비한다.

“체력 N이하의 모든 몬스터들을 잡을 때, 스킬 사용이 가장 많을 때의 횟수와, 그럴 경우에 가장 체력이 적은 몬스터의 체력은?”

종훈이의 대답은?

예를들면,

세영 : 체력 12이하의 모든 몬스터들을 잡을 때, 스킬 사용이 가장 많을 때의 횟수와, 그럴 경우에 가장 체력이 적은 몬스터의 체력은?

종훈 : (스킬 사용이 가장 많을 때는 10, 11일때 4회잖아? 그렇다면!) 4 10

### 입력

첫 번째 줄에 테스트 케이스 T가 주어진다. 각 케이스의 첫번째 줄에는 세영이의 질문에 등장하는 체력  $N(1 \leq N \leq 10000)$ 이 주어진다.

### 출력

종훈이의 대답을 출력하시오.

#### 예제 입력

#### 예제 출력

2	4 10
12	17 4437
5555	

# Problem B

## 자물쇠

시간 제한 : 1 Second

위대한 모험가 인디아나 일신은 고대의 안산 문명을 탐험하다가 에리카 신을 모시는 신전인 모모타니 신전에 당도하게 된다. 모모타니 신전에 들어가기 위해서는 4개의 번호로 이루어진 거대한 자물쇠를 풀어야 한다고 한다. 인디아나 일신은 안산 문명에 대해 조예가 깊었기 때문에, 이미 자물쇠의 비밀번호를 이미 알고 있었다. 하지만, 에리카 신전의 자물쇠는 너무 오래되어 심하게 돌리면 부서져 영원히 신전에는 들어갈 수 없게 된다. 따라서, 자물쇠는 최소한의 움직임만을 취해야 한다. 또한, 자물쇠의 철이 녹슬어 전체를 같은 방향으로 1칸 움직이거나 한 칸의 다이얼을 최대 3칸 움직이는 것만이 가능하다고 한다.

전설에 따르면 모모타니 신전에는 에리카 신이 남긴 “그것”이 있다고 한다. 인디아나 일신은 “그것”을 박물관으로 보내고 싶어 한다. 인디아나 일신을 도와 에리카 신전의 자물쇠를 열기 위해 필요한 움직임의 횟수를 출력하는 프로그램을 작성하시오. (단, 9에서 위로 한칸 돌리면 0이 되고, 0에서 아래로 한칸 돌리면 9가 된다.)

### 입력

첫 번째 줄에 테스트 케이스의 수 T가 주어진다. 각 테스트 케이스에는 0000부터 9999사이의 숫자 두 개가 공백 한 칸을 사이에 두고 주어진다. 첫 번째 수는 현재의 자물쇠 번호, 두 번째 수는 자물쇠를 풀기 위한 번호이다.

### 출력

각각의 케이스마다 한 줄에 자물쇠를 조건에 따라 돌려서 열 수 있는 최소 횟수를 출력한다.

#### 예제 입력

2	4
0000 1234	6
5913 2578	

#### 예제 출력

# Problem C

## 막타는 나의 것

시간 제한 : 1 Second

민건이는 요새 니리지라는 RPG게임을 통해 용돈 벌이를 하고 있다. 피눈물 나는 노력 끝에, 민건이는 1등급 보스 몬스터인 붉은 해서기를 죽일 기회를 얻게 되었다. 안타깝게도, 보스 몬스터의 체력을 깎기 위해 모든 템과 자신의 체력을 많이 소모한 민건이는 평타 밖에 칠 수 없는 상태가 되었다. 당장이라도 해서기의 목을 치고 싶지만 안타깝게도 막타를 뺏어 아이템과 경험치를 얻으려는 유저 수형이와 대치하게 된다. 수형이도 보스 몬스터를 잡으러 오기 위해 체력과 아이템을 쓰며 왔기 때문에 평타 밖에는 칠 수 없다.

서로 눈치 싸움을 하며 막타를 치기 위한 공격을 하기 때문에, 민건이가 보스 몬스터에게 A의 피해를 입히면 수형이는 다음 턴에  $1 \times A \sim 2 \times A$ 만큼의 피해를 입힐 수 있다.

민건이 먼저 공격하고 서로 차례를 번갈아가며 최선을 다해서 막타를 치기 위해 공격한다고 할 때, 민건이가 이기기 위해 처음 턴에 보스 몬스터에게 가해야할 최소 피해가 얼마인지 구하는 프로그램을 작성하시오.

### 입력

첫 번째 줄에 테스트 케이스 T가 주어진다. 각 케이스의 첫번째 줄에는 보스 몬스터의 체력 N이 주어진다. ( $1 \leq N \leq 2^{31}-1$ )

### 출력

각각의 케이스마다 한 줄에 민건이가 승리하기 위해 첫번째 턴에서 입힐 피해의 최소값을 출력한다.

#### 예제 입력

#### 예제 출력

2	1
4	2
7	

# Problem D

## 60간지

시간 제한 : 1 Second

수형이는 2017년 일출여행계획을 짜기위해 내년 신정의 요일이 연휴를 만드는지 알아보려고 달력을 보던 중 각 연도별 1월 1일이 무슨 요일인지 궁금해졌다. 그리고 달력에 병신년이라고 적혀있는 것을 보고 다른 단어도 많은데 왜 하필 병신년인지 알아보던 중 '60간지'라는 상식을 깨우쳤다.

'60간지'는 하늘을 상징하는 10가지, 땅을 상징하는 12가지 동물을 조합하여 60가지의 경우로 나타낸 것이다. 10 간에는 갑, 을, 병, 정, 무, 기, 경, 신, 임, 계가 있으며, 12지에는 자, 축, 인, 묘, 진, 사, 오, 미, 신, 유, 술, 해가 있다. 10 간과 12지를 차례대로 붙인 것을 육십갑자라 부른다. 예를 들면, 갑자 -> 을축 -> 병인 -> 정묘 순서인 것이다.

수형이의 호기심을 해결하기 위해 주어진 연도가 60간지 중 어느 것이며, 그 해의 1월 1일은 무슨 요일인지 알아주자. (단, 윤년은 1년이 366일이다. 윤년은 4의 배수이면서 100의 배수가 아닌 해거나, 400의 배수인 해이다.)

### 입력

첫 번째 줄에 테스트 케이스의 수를 의미하는 자연수 T가 주어진다. 그 다음에는 T개의 테스트 케이스가 주어진다. 각 테스트 케이스는 자연수 하나로 이루어져있다. 자연수는 1보다 크거나 같고, 4000보다 작거나 같다.(단, 서기 1년 1월 1일은 월요일이다. 그리고 서기 4년이 갑자년이다. 현재 날짜와 다름에 유의하라.)

### 출력

각각의 케이스마다 한 줄에 60간지 중 어느 것 인지와 해당 연도의 1월 1일이 무슨 요일인지 공백을 통해 구분하여 출력한다. (10간은 영문 대문자 순서로, 12지는 영문 소문자 순서로 바뀌서 출력해야 한다.)

#### 예제 입력

2 2000 1919	Ge Saturday Fh Wednesday
-------------------	-----------------------------

#### 예제 출력

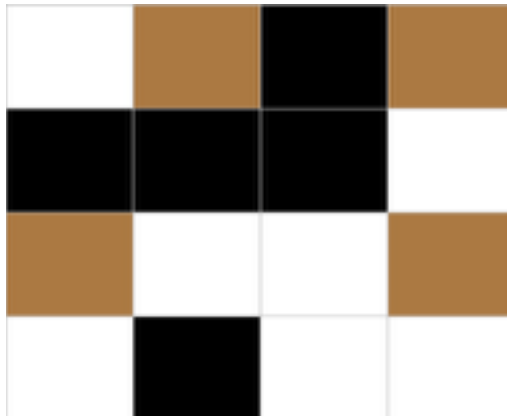
- \* 월요일 Monday 화요일 Tuesday 수요일 Wednesday 목요일 Thursday 금요일 Friday
- \* 토요일 Saturday 일요일 Sunday

# Problem E

## 메이즈러너

시간 제한 : 1 Second

제 1회 HEPC에 참가하지 않은 프로불참러 조세호우는 안재호우 결혼식에 가던 도중 길을 잃고 헤매다 미로에 갇히게 되었다. 얼른 결혼식을 가야하는 조세호우는 미로 속에서 자신과 같은 상황에 처한 사람들을 만나게 되고, 그들은 미로에서 탈출하기 위해 미로의 모양을 파악하기 시작했다. 미로에는 구덩이가 존재하며, 그 구덩이를 통해 다른 구덩이가 있는 곳으로 갈 수 있다는 사실을 알아냈다. 구덩이는 입구와 출구가 정해져있으며 출구로 들어가서 입구로 나갈 수 없다. 그리고 구덩이를 들어가는 것보다 지나쳐가는 것이 더 빠르다면 그냥 지나쳐 갈 수 있다. 조세호는 결혼식이 끝나기전에 얼른 탈출하여 서른세바퀴 녹화에 가야하기 때문에 미로를 최단시간내에 탈출하고 싶어한다. 조세호가 미로를 성공적으로 탈출할 수 있는지 알아내보자. 미로의 출발점은 항상 왼쪽 맨 위, 도착점은 오른쪽 맨 아래이다.



위의 예제(4 \* 4)에서 검은칸은 벽(숫자 1), 흰칸은 길(숫자 0), 갈색칸은 구덩이(숫자 2)이다. (0, 1)의 구덩이는 (0, 3)의 구덩이와 연결되어 있고, (2, 3)의 구덩이는 (2, 0)의 구덩이와 연결되어 있다. 왼쪽 맨 위의 좌표를 (0, 0)으로 설정하면 (0, 0) - (0, 1) - (0, 3) - (1, 3) - (2, 3) - (3, 3) 순서로 최단의 시간(5초)으로 탈출하게 된다. 미로는 한칸 움직일때마다 1초의 시간이 지난다고 가정한다.

### 입력

첫 번째 줄에 테스트 케이스의 수를 의미하는 자연수 T가 주어진다. 그 다음에는 T개의 테스트 케이스가 주어진다. 각 테스트 케이스 첫 줄에 미로의 크기 N, M ( $1 < N, M \leq 1000$ )이 주어지며, 두번째 줄 부터 N개의 미로정보가 M줄 주어진다. 맵이 N줄에 걸쳐서 주어진 다음 구덩이의 개수 H ( $\leq 1000$ )가 주어지며, 구덩이의 개수만큼 입구 좌표와 출구 좌표가 한 줄에 하나씩 4개의 숫자가 주어지게 된다. 미로의 입구와 출구는 0으로 되어있고, 구덩이의 개수(미로에 있는 2의 개수)가 미로의 25%를 초과하는 경우는 없다.

(0 : 길, 1 : 벽, 2 : 구덩이)

### 출력

각각의 케이스마다 한 줄에 정답에 해당하는 값을 출력한다. 이 값은 미로를 탈출 할 수 있으면 탈출할 수 있는 최소 횟수를 나타낸다. 만약 미로를 탈출할 수 없다면 'Proabsentee'를 출력한다.

### 예제 입력

```
1
4 4
0 2 1 2
1 1 1 0
2 0 0 2
0 1 0 0
2
0 1 0 3
2 3 2 0
```

### 예제 출력

```
5
```



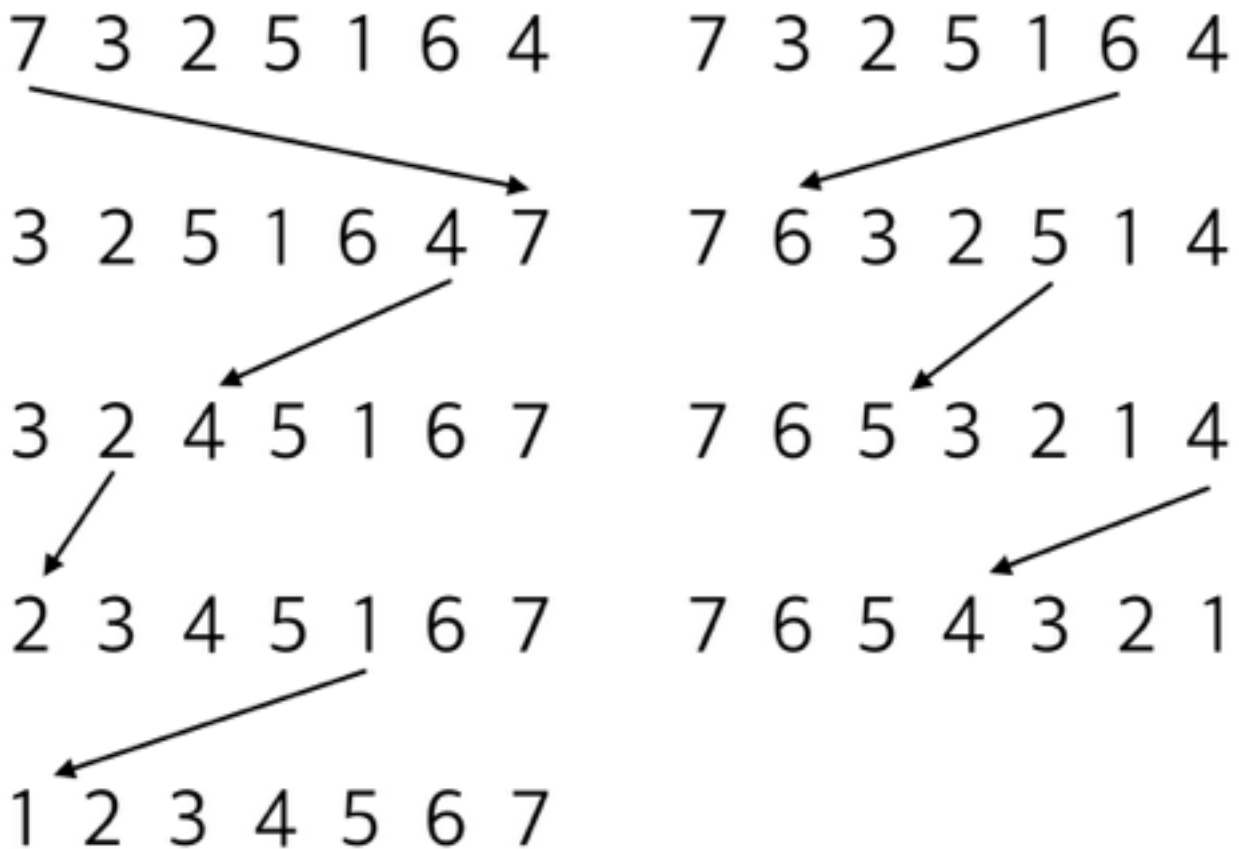
# Problem F

## 나폴레옹의 모자란 군사들

시간 제한 : 1 Second

나폴레옹의 군사들은 원정을 가던 중 숲 속이 흔들리는 수상쩍음을 발견했다. 순간적으로 그들은 전투 태세를 갖추었지만, 숲 속에서 나온 것은 사슴이었고 깜짝 놀란 군사들은 혼비백산하여 대열이 흐트러지고 말았다. 그런데 이 상황이 빨리 복구되지 않는 것을 나폴레옹이 발견하고는 매우 심기가 불편해졌다. 군사들은 심기가 불편한 나폴레옹이 화를 내기 전에 빨리 대열을 원상태로 복구하고 전진하여야 한다. 그런데 이 모자란 군사들은 자신의 앞 사람만 바라보고 가기 때문에 전진하기 위해서는 그 방향에서부터 오름차순이 되어야 한다. 어느 방향으로 가야 더 빠르게 대열을 복구할까?

예를 들어, 7 명의 군사들이 있다고 가정하자. 그렇다면 군사들은 각각 1 부터 7 까지의 번호를 나누어 가지고 있다. 군사들의 현재 대열 상태가 7 3 2 5 1 6 4 라고 할 때, 아래의 그림을 통해 방향에 따라 어떻게 정렬하는지 살펴보자.



왼쪽 방향으로 나아가기 위해서 정렬하는 것은 4번의 움직임이 필요하고, 오른쪽 방향으로 나아가기 위해서는 3번의 움직임 밖에 필요하지 않다. 따라서 오른쪽 방향으로 나아가는 것이 더 빠른 복구를 할 수 있다. 그렇기 때문에 출력되는 것은 "RIGHT" 이다. 이 때, 좌우가 필요한 움직임의 수가 같을 경우 "IDGARA" 를 출력한다.

## 입력

첫 번째 줄에는 테스트 케이스의 수인 정수  $T$  ( $1 \leq T$ ) 가 입력된다. 나폴레옹 군사의 수  $N$ 을 입력 받는다. 그리고 각각의 군사들은 1 부터  $N$  까지의 번호를 서로 나누어 가지고 있다. 따라서 두 번째 줄에는 그 군사들의 현재 대열 상태로서 군사들의 번호들이 공백을 구분으로 랜덤하게 입력된다. ( $1 \leq N \leq 10,000$ )

## 출력

첫 번째 줄에 왼쪽으로 나아가는 것이 더 빠르다면 “LEFT” 를, 오른쪽으로 나아가는 것이 더 빠르다면 “RIGHT” 가, 좌우에 필요한 움직임의 수가 같은 경우 “IDGARA” 출력한다.

### 예제 입력

### 예제 출력

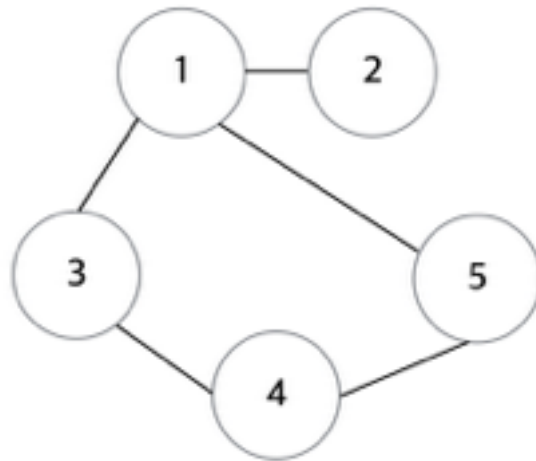
2	RIGHT
7	IDGARA
7 3 2 5 1 6 4	
7	
7 2 3 6 4 5 1	

# Problem G

## 나-는 중국집 사장

시간 제한 : 1 Second

나-는 중국집 사장이다. 그리고 배달도 내-가 한다. 근데 지금 주문이 너-무 많이 밀려 있다. 더군다나 주문 마-다-마-다 폴코스요리를 주문해서 배달해주고 다시 가게에 들러 새로 음식을 가져가서 배달을 또- 해야 한다. 심지어 우리 동네에는 있는 길도 있고 없는 길도 있고 하기 때문에 길을 잘 찾아보고 가야 빠르게 배달이 가능하다. 나-는 전.지.적 작가 시점이라서 어느 집에서 다른 집으로의 길이 있는지 없는지 알 수 있다. 빠른 길로만 다니면서 모-든 집에 배달을 완료 하려면 몇 번이나 그 길들을 왔다 갔다 해야할까?



나-의 중국집은 1에 존재한다. 따라서 1에서부터 다른 집까지의 거리를 계산하여 모두 더 해주면 된다. 물론 길이 없는 집은 배달가지 않는다. 따라서 그런 집은 지도에 그려주지도 않을 것이다. 우리 동네 지도가 예시와 같다면,

1 -> 2 -> 1 : 길 2번  
1 -> 3 -> 1 : 길 2번  
1 -> 3 -> 4 -> 3 -> 1 : 길 4번  
1 -> 5 -> 1 : 길 2번

이기 때문에  $2 + 2 + 4 + 2 = 10$  개의 길을 다녀야 배달이 완료된다.

### 입력

입력의 첫 줄에는 테스트 케이스의 수인 정수  $T$  ( $1 \leq T$ ) 가 입력되고 다음 줄 부터 각 케이스마다 집의 수  $N$  과 집과 집 사이에 길이 존재하는지 알려주는 순서쌍의 개수  $R$  이 입력된다. 그리고 다음  $R$  줄 동안 집과 집 사이에 길의 존재로 순서쌍이 입력된다. 순서쌍은 1 부터  $N$  이하의 수로만 이루어져 있다. ( $2 \leq N \leq 100, 1 < R < N(N+1)/2$ )

### 출력

$T$  줄에 걸쳐 각 상황에 대한 배달을 위한 길 횟단 횟수를 쓴다.

**예제 입력**

**예제 출력**

1 5 5 1 2 1 3 1 5 3 4 4 5	10
---	----

# Problem H

## PRODUCE IOI

시간 제한 : 1 Second

IOI의 데뷔 소식에 들뜬 민건이는 사람들이 IOI를 얼마나 좋아하는지 점수화 할 수 있는 테스트를 고안했다.  $N+1$ 개의 I와  $N$ 개의 O로 이루어져 있으면, I와 O가 교대로 나오는 문자열을  $P_N$ 이라고 한다.

$P_1$  IOI  
 $P_2$  IOIOI  
 $P_3$  IOIOIOI  
 $P_N$  IOIOL...OI (O가  $N$ 개)

임의로 생각한 I와 O로만 이루어진 문자열  $S$ 와 정수  $N$ 이 주어졌을 때,  $S$ 안에  $P_N$ 이 포함된 개수만큼 점수를 얻을 수 있다. 이 점수를 구하는 프로그램을 작성하시오.

### 입력

첫 번째 줄에 테스트 케이스의 수를 의미하는 자연수  $T$ 가 주어진다. 그 다음에는  $T$ 개의 테스트 케이스가 주어진다. 각 테스트 케이스는 첫째 줄에  $N$ , 둘째 줄에  $S$ 의 길이  $M$ , 셋째 줄에  $S$ 로 이루어져있다.. ( $1 \leq N \leq 1,000,000$ ,  $2N+1 \leq M \leq 1,000,000$ )

### 출력

몇 점을 얻었는지 출력한다.

### 예제 입력

```
2
1
13
OOIOIOIOIOIOI
2
13
OOIOIOIOIOIOI
```

### 예제 출력

```
4
2
```

# Problem I

## Spell Checker

시간 제한 : 1 Second

스펠체킹 프로그램(spell checker)이 잘못 쓰인 단어를 만나는 상황을 생각해보자. 아마도 잘못 쓴 그 단어는 사전에서 발견되지 않을 것이지만 그것과 가장 유사한 단어를 찾을 수는 있을 것이다. 이런 경우에 사전에서 그 단어와 가장 가깝다는 사실을 어떻게 알수있을까?

아마도 두 단어를 일렬로 정렬시켜놓고 일치하지 않는 문자의 갯수를 두 단어 사이의 "편집간격"으로 정의해서 편집간격이 가장 작은 숫자에 해당하는 단어를 찾는 식으로 스펠체킹 프로그램을 작성할 수 있을것이다. 편집간격은 두 단어 사이의 다양한 정렬방식 중에서 차이가 나는 문자의 갯수의 최소값으로 정의할 수 있다. 예를들면, "snowy" 라는 단어와 "sunny" 라는 단어는 다음의 2가지 방식으로 정렬시켜서 비교해볼 수 있다.

방법1	방법2
S * N O W Y	* S N O W * Y
S U N N * Y	S U N * * N Y

첫번째 방법으로 두 단어를 대비시킬 경우 모두 6개의 문자중에서 3개의 불일치가 발생하고 두번째 방법으로 두 단어를 대비시킬 경우 모두 7개의 문자중에서 5개의 불일치가 발생한다. 따라서 두 단어 사이의 편집간격은 가능한 최소값인 3 으로 결정된다.

일반적으로는 당연히 수많은 정렬방법들 중에서 불일치하는 문자의 갯수를 최소화 하는 경우를 찾아야 한다. 두 단어 사이의 편집간격을 구하는 프로그램을 작성하라.

### 입력

첫 번째 줄에 테스트 케이스의 수를 의미하는 자연수 T가 주어진다. 그 다음에는 T개의 테스트 케이스가 주어진다. 각 테스트 케이스는 문자열 2개로 이루어져있다. 두 문자열의 길이는 10이하이고, 길이가 0인 단어는 주어지지 않는다.

### 출력

각각의 케이스마다 한 줄에 편집간격을 출력한다.

#### 예제 입력

#### 예제 출력

1 snowy sunny	3
---------------------	---

# Problem J

## 우리는 별을 찍는 소년들

시간 제한 : 1 Second

우리는 우리의 대회를 크게 홍보하기 위해 글씨를 대문짝만하게 뽑기로 했다. 그런데 명색이 대회인데 글씨를 일일이 손으로 적을 순 업지 않지 않겠는가? 따라서 우리는 컴퓨터를 이용해서 글씨를 뽑기로 했다. 그리고 멀리서도 보이게 번쩍번쩍 빛나게 LED를 넣을 것이기 때문에 별 모양으로 크게크게 찍어낼 것이다. 자 이제 우리 모두 함께 대회명을 대문짝만하게 찍어보자.

### 입력

입력은 주어지지 않는다.

### 출력

HEPC를 아래와 같이 대문짝만하게 출력한다. 각 알파벳 사이에 한 칸의 공백이 존재한다.

#### 예제 입력

#### 예제 출력

	<pre>* * **** **** ** * * * * * **** **** * * * * * * * * **** * ****</pre>
--	---