

## Triangolo (Discesa): computa il rango di una soluzione ottima data

Ti viene dato un triangolo di numeri interi tutti presi dall'intervallo  $[0,9]$ :

```
    4
   1 5
  7 2 3
 1 2 1 2
1 5 3 3 3
```

Consideriamo ammissibile un percorso che discenda dal vertice in alto (quì di valore 4) visitando precisamente uno dei valori per ciascuna riga senza mai compiere salti eccessivi, nè verso sinistra nè verso destra. In pratica, nel passare da una riga alla successiva, non essendo presente un elemento immediatamente sottostante, l'unica scelta consentita è quella di portarsi verso sinistra oppure verso destra, ma solo fino al primo elemento della riga successiva che si incontra in tale direzione. Dove  $n$  è il numero di righe del triangolo, siamo quindi chiamati a compiere  $n - 1$  scelte del tipo “verso destra” oppure “verso sinistra”. Un tale percorso trova quindi codifica in una stringa di lunghezza  $n - 1$  sull'alfabeto  $\{L, R\}$ . Una soluzione ottima è un percorso ammissibile di massimo valore, possono essercene anche in numero esponenziale in  $n$  (ve ne sono esattamente  $2^{n-1}$  quando tutti gli interi del triangolo sono uguali). Possiamo tuttavia ordinare tali soluzioni ottime secondo l'ordine lessicografico delle stringhe su  $\{L, R\}$  che le rappresentano. Ad esempio, la soluzione ottima con rappresentazione lessicograficamente minima sarà quella di rango zero. Dato il triangolo ed una soluzione ottima entro esso, si richiede di restituire il rango di tale soluzione ottima.

### Input

La prima riga contiene  $T$ , il numero di testcase da risolvere. Seguono  $T$  istanze del problema. Ogni istanza è composta nel seguente modo: la prima riga fornisce  $n$ , il numero di righe del triangolo. Seguono le  $n$  righe del triangolo allineate tutte a sinistra. L'ultima riga fornisce la codifica di un percorso ammissibile che è garantito essere un percorso di valore massimo. Tale codifica è una stringa di lunghezza  $n - 1$  sull'alfabeto  $\{L, R\}$  (sinistra/destra, specificando le scelte come a partire dal vertice in alto).

### Output

L'output atteso consta di tre righe per ogni testcase: la prima riga deve contenere il massimo valore di un percorso ammissibile, la seconda conterrà il numero di percorsi ottimi, la terza ritorna il rango della soluzione ottima ricevuta in input.

## Esempio

### Input

```
3
1
7

5
4
1 5
7 2 3
1 2 1 2
1 5 3 3 3
LLLL
4
1
1 1
1 1 1
1 1 1 1
LRR
```

Spiegazione: tre testcase. Nel primo  $n = 1$  e la stringa vuota codifica l'unica soluzione ottima (che avrà  $rank = 0$ ), il triangolo assegnato ha una sola riga e consta di un singolo valore intero, il 7. Il secondo triangolo ha 5 righe. Il terzo ha 4 righe e tutti i 10 valori che lo compongono sono pari ad 1 (le soluzioni ottime sono 8 e la soluzione ottima proposta (il percorso codificato dalle scelte "LRR") ha rango 3, essendo il quarto secondo l'ordinamento lessicografico).

### Output

```
7
7
1
0
19
1
0
4
8
3
```

Spiegazione: nel primo testcase, la soluzione ottima ha valore 7 ed è unica; essa trova codifica nella stringa vuota. Nel secondo testcase la soluzione ottima è di nuovo unica (un percorso di valore 19. Nel terzo testcase tutti i  $2^3 = 8$  percorsi ammissibili hanno lo stesso valore (4); di queste 8 soluzioni ottime, le prime 4 iniziano in  $L$  e le altre 4 iniziano in  $R$ .

## Assunzioni

Per il subtasking sono previste le seguenti **size**, dove il default è **big** che include anche i testcase **medium**, **small** e **tiny**:

- **tiny**:  $n \leq 7$
- **small**:  $n \leq 10$
- **medium**:  $n \leq 28$
- **big**:  $n \leq 40$

Il tempo limite per testcase è di 1 secondo.