

# Mengumpulkan Peserta

Time limit: 1 s

Memory limit: 256 MB

## Deskripsi

Perlombaan OSN akan segera dimulai! Di dalam ruang lomba telah disediakan kursi yang disusun membentuk persegi panjang berisi  $R$  baris (dinomori 1 sampai  $R$ )  $\times$   $C$  kolom (dinomori 1 sampai  $C$ ). Setiap kursi mungkin dalam keadaan kosong atau ditempati oleh seorang peserta OSN.

Keadaan seluruh kursi di dalam ruang lomba pada saat ini dapat direpresentasikan dalam sebuah matriks  $S$ . Apabila kursi pada baris ke- $i$  dan kolom ke- $j$  pada saat ini dalam keadaan kosong, maka  $S[i][j]$  berisi sebuah karakter  $0$ . Sebaliknya, apabila kursi pada baris ke- $i$  dan kolom ke- $j$  pada saat ini ditempati seorang peserta, maka  $S[i][j]$  berisi sebuah karakter  $1$ .

Pak Dengklek ingin memberikan pengarahan kepada para peserta OSN. Melihat para peserta OSN yang duduknya tersebar, Pak Dengklek akan mengumpulkan mereka terlebih dahulu sebelum memulai pengarahan. Pak Dengklek ingin agar seluruh peserta terkumpul dalam satu persegi panjang. Dengan kata lain, seluruh peserta duduk di antara baris ke  $r_1$  sampai  $r_2$  dan kolom ke  $c_1$  sampai  $c_2$  (untuk suatu nilai  $1 \leq r_1 \leq r_2 \leq R$  dan  $1 \leq c_1 \leq c_2 \leq C$ ), dan tidak ada kursi kosong yang berada di dalam rentang baris dan kolom tersebut.

Untuk melakukannya, Pak Dengklek mungkin saja harus memindahkan beberapa peserta ke tempat duduk lain yang mana saja selama masih kosong. Pak Dengklek ingin memindahkan sesedikit mungkin peserta karena setiap pemindahan mengganggu peserta lainnya. Pak Dengklek penasaran akan banyaknya pemindahan minimum yang harus dilakukan agar seluruh peserta terkumpul dalam satu persegi panjang, atau jika hal tersebut tidak mungkin.

## Format Masukan

Masukan diberikan dalam format berikut:

```
R C
S[1][1]S[1][2]...S[1][C]
S[2][1]S[2][2]...S[2][C]
.           . .
.           . .
.           ..
S[R][1]S[R][2]...S[R][C]
```

## Format Keluaran

Sebuah baris yang berisi banyaknya pemindahan minimum yang harus dilakukan agar seluruh peserta terkumpul dalam satu persegi panjang, atau  $-1$  jika hal tersebut tidak mungkin.

### Contoh Masukan 1

```
4 3
000
110
110
000
```

### Contoh Keluaran 1

```
0
```

### Penjelasan Contoh 1

Pada contoh ini, seluruh peserta telah terkumpul dalam satu persegi panjang, sehingga tidak ada pemindahan yang perlu dilakukan.

### Contoh Masukan 2

```
3 4
0010
1100
0100
```

### Contoh Keluaran 2

```
1
```

### Penjelasan Contoh 2

Pada contoh ini, Pak Dengklek dapat memindahkan peserta yang sedang duduk di baris pertama dan kolom ketiga ke tempat duduk pada baris ketiga dan kolom pertama. Setelah pemindahan itu, seluruh peserta telah terkumpul dalam satu persegi panjang, sehingga hanya diperlukan 1 pemindahan.

### Contoh Masukan 3

```
3 3
111
101
```

### Contoh Keluaran 3

-1

### Penjelasan Contoh 3

Pada contoh ini, tidak ada cara untuk mengumpulkan seluruh peserta dalam satu persegi panjang.

### Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq R, C$ .
- $1 \leq R \times C \leq 100.000$ .
- S hanya mengandung karakter `0` atau `1`.
- Terdapat setidaknya satu karakter `1` pada S.

### Subsoal 1 (5 poin)

Hanya berisi kasus uji berikut:

```
10 10
000000001
0010000100
0110111000
0111111000
0111111001
0101111000
0110111010
0110101000
0111111000
0000000000
```

### Subsoal 2 (7 poin)

Hanya berisi kasus uji berikut:

```
10 10
0101010101
1010101010
0101010101
```

1010101010

0101010101

1010101010

0101010101

1010101010

0101010101

1010101010

### **Subsoal 3 (12 poin)**

- $R = 1$ .
- $R \times C \leq 2.000$ .

### **Subsoal 4 (16 poin)**

- $R = 1$ .

### **Subsoal 5 (15 poin)**

- $R \times C \leq 200$ .

### **Subsoal 6 (22 poin)**

- $R \times C \leq 2.000$ .

### **Subsoal 7 (23 poin)**

- Tidak ada batasan tambahan.

# Pertahanan Manado

Time limit: 1 s

Memory limit: 256 MB

## Deskripsi

Kota Manado sedang diserang wabah penyakit hebat! Pak Dengklek, sebagai tabib kota Manado, menyimpulkan bahwa warga Manado membutuhkan ramuan penyembuh yang dapat diracik dari  $N$  jenis obat, dinomori 1 sampai  $N$ .

Toko obat terbesar di Manado menjual setiap jenis obat dengan harga per botol masing-masing seharga  $S$  rupiah, untuk setiap jenisnya. Di samping itu, toko tersebut menjual  $M$  paket obat, dinomori 1 sampai  $M$ . Paket obat ke- $i$  terdiri dari obat jenis ke- $A[i]$  hingga ke- $B[i]$ , masing-masing satu botol. Paket obat ke- $i$  dijual seharga  $P[i]$  rupiah.

Bantulah Pak Dengklek untuk menentukan, berapa harga termurah untuk membeli minimum satu botol untuk masing-masing dari  $N$  jenis obat tersebut, untuk menyelamatkan warga Manado dari wabah penyakit tersebut.

## Format Masukan

Masukan diberikan dalam format berikut:

```
N M S
A[1] B[1] P[1]
A[2] B[2] P[2]
.
.
.
A[M] B[M] P[M]
```

## Format Keluaran

Sebuah baris yang berisi harga termurah untuk membeli minimum satu botol untuk masing-masing dari  $N$  jenis obat tersebut.

## Contoh Masukan 1

```
5 3 3
1 2 2
4 5 1
2 4 10
```

## Contoh Keluaran 1

6

### Penjelasan Contoh 1

Pada contoh ini, Pak Dengklek dapat membeli paket obat pertama (dengan harga 2 rupiah) dan kedua (dengan harga 1 rupiah), serta membeli jenis obat ketiga secara terpisah (dengan harga 3 rupiah). Sehingga, Pak Dengklek menghabiskan 6 rupiah dan mengumpulkan satu botol untuk masing-masing N jenis obat. Tidak ada cara agar Pak Dengklek menghabiskan kurang dari 6 rupiah.

## Contoh Masukan 2

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

## Contoh Keluaran 2

5

### Penjelasan Contoh 2

Pada contoh ini, Pak Dengklek dapat membeli seluruh paket obat dan menghabiskan 5 rupiah.

### Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq N \leq 200.000$ .
- $1 \leq M \leq 200.000$ .
- $1 \leq S \leq 1.000.000$ .
- $1 \leq A[i] \leq B[i] \leq N$ .
- $1 \leq P[i] \leq 1.000.000$ .

#### Subsoal 1 (5 poin)

Hanya berisi kasus uji berikut:

```
1001 1 999
1 1001 1000000
```

#### Subsoal 2 (9 poin)

Hanya berisi kasus uji berikut:

16 7 10

1 6 17

2 4 8

5 10 30

7 9 22

11 15 47

12 14 28

15 16 18

### **Subsoal 3 (13 poin)**

- $N \leq 200$ .
- $M \leq 16$ .

### **Subsoal 4 (13 poin)**

- $N \leq 200$ .
- $M \leq 200$ .

### **Subsoal 5 (17 poin)**

- $N \leq 2.000$ .
- $M \leq 2.000$ .

### **Subsoal 6 (18 poin)**

- $P[i] = S$ .

### **Subsoal 7 (25 poin)**

- Tidak ada batasan tambahan.

# Rekreasi OSN

Time limit: 0 s

Memory limit: 0 MB

## Deskripsi

Setelah kedua hari kompetisi, peserta OSN akan mengikuti kegiatan rekreasi. Tempat rekreasi dapat direpresentasikan sebagai petak-petak yang memiliki R baris (dinomori 1 sampai R dari utara ke selatan) dan C kolom (dinomori 1 sampai C dari barat ke timur). Petak yang terletak pada baris ke-i dan kolom ke-j dinotasikan sebagai petak (i, j). Setiap petak mungkin saja kosong atau berisi penghalang.

Keadaan seluruh petak di dalam tempat rekreasi ini dapat direpresentasikan dalam sebuah matriks S. Apabila petak (i, j) pada awalnya kosong, maka  $S[i][j]$  berisi sebuah karakter `.`. Sebaliknya, apabila petak (i, j) pada awalnya berisi penghalang, maka  $S[i][j]$  berisi sebuah karakter `#`. Pintu masuk tempat rekreasi ini terletak pada petak (1, 1) dan pintu keluar tempat rekreasi ini terletak pada petak (R, C). Seluruh peserta pada awalnya berada pada petak (1, 1). Dalam setiap langkah, peserta dapat berpindah satu petak ke arah selatan atau timur, sehingga mendekati pintu keluar. Peserta hanya dapat berpindah ke petak yang kosong.

Panitia OSN telah menyiapkan rute rekreasi yang sebaiknya dilewati oleh peserta. Rute ini direpresentasikan dalam sebuah string M yang berisi  $R + C - 2$  karakter. Apabila langkah ke-i pada rute ini merupakan satu petak ke arah timur, maka  $M[i]$  berisi sebuah karakter `>`. Sebaliknya, apabila langkah ke-i pada rute ini merupakan satu petak ke arah selatan, maka  $M[i]$  berisi sebuah karakter `v`. Rute ini dijamin hanya melewati petak kosong.

Celakanya, mungkin saja terdapat rute lain dari pintu masuk ke pintu keluar. Hal ini merepotkan panitia untuk mengawasi seluruh peserta karena peserta mungkin saja mengambil rute lain tersebut. Karenanya, panitia ingin meletakkan penghalang di beberapa petak kosong yang tidak dilewati oleh rute yang disiapkan oleh panitia agar terdapat hanya satu rute dari pintu masuk ke pintu keluar. Dengan kata lain, seluruh rute dari pintu masuk ke pintu keluar melewati himpunan petak yang persis sama dengan rute yang disiapkan oleh panitia.

Karena meletakkan penghalang di petak kosong memakan waktu, panitia ingin meletakkan sesedikit mungkin penghalang. Tentukan petak mana saja yang harus diletakkan penghalang. Apabila terdapat lebih dari satu solusi, Anda dapat mengeluarkan yang mana saja. Nilai Anda akan bergantung pada banyaknya penghalang yang Anda letakkan, menggunakan rumus yang akan dijelaskan pada bagian Penilaian.

## Informasi Tipe Soal

Soal ini bertipe "output-only". Untuk setiap kasus uji, Anda menuliskan keluaran program ke dalam sebuah berkas keluaran.

Masukan untuk soal ini dapat diunduh di [sini](#). Di dalam berkas .zip tersebut terdapat 2 + 10 masukan untuk diselesaikan: `rekreasi_sample_1.in`, `rekreasi_sample_2.in`, `rekreasi_1.in`, `rekreasi_2.in`, ..., `rekreasi_10.in`. Masukan contoh (`rekreasi_sample_1.in` dan `rekreasi_sample_2.in`) tidak termasuk dalam penilaian peserta.

Untuk setiap berkas masukan yang diselesaikan (Anda tidak harus menyelesaikan semua masukan), buatlah berkas keluaran dengan nama `rekreasi_sample_X.out` (untuk masukan contoh)



atau rekreasi\_X.out, dengan X adalah nomor kasus uji. Setelah itu, kompres semua berkas keluaran dalam sebuah berkas .zip, lalu kumpulkan.

## Format Masukan

Masukan diberikan dalam format berikut:

```
R C
M[1]M[2]...M[R + C - 2]
S[1][1]S[1][2]...S[1][C]
S[2][1]S[2][2]...S[2][C]
.           . .
.           . .
.           ..
S[R][1]S[R][2]...S[R][C]
```

## Format Keluaran

R baris: baris ke-i berisi C karakter. Karakter ke-j pada baris ke-i adalah:

- **.** jika petak (i, j) pada awalnya kosong dan Anda membiarkannya kosong.
- **x** jika petak (i, j) pada awalnya kosong dan Anda meletakkan penghalang pada petak tersebut.
- **#** jika petak (i, j) pada awalnya berisi penghalang.

## Contoh Masukan 1

```
5 5
>>vvv>>v
....#
.#..#
.#.#.
.....
...#.
```

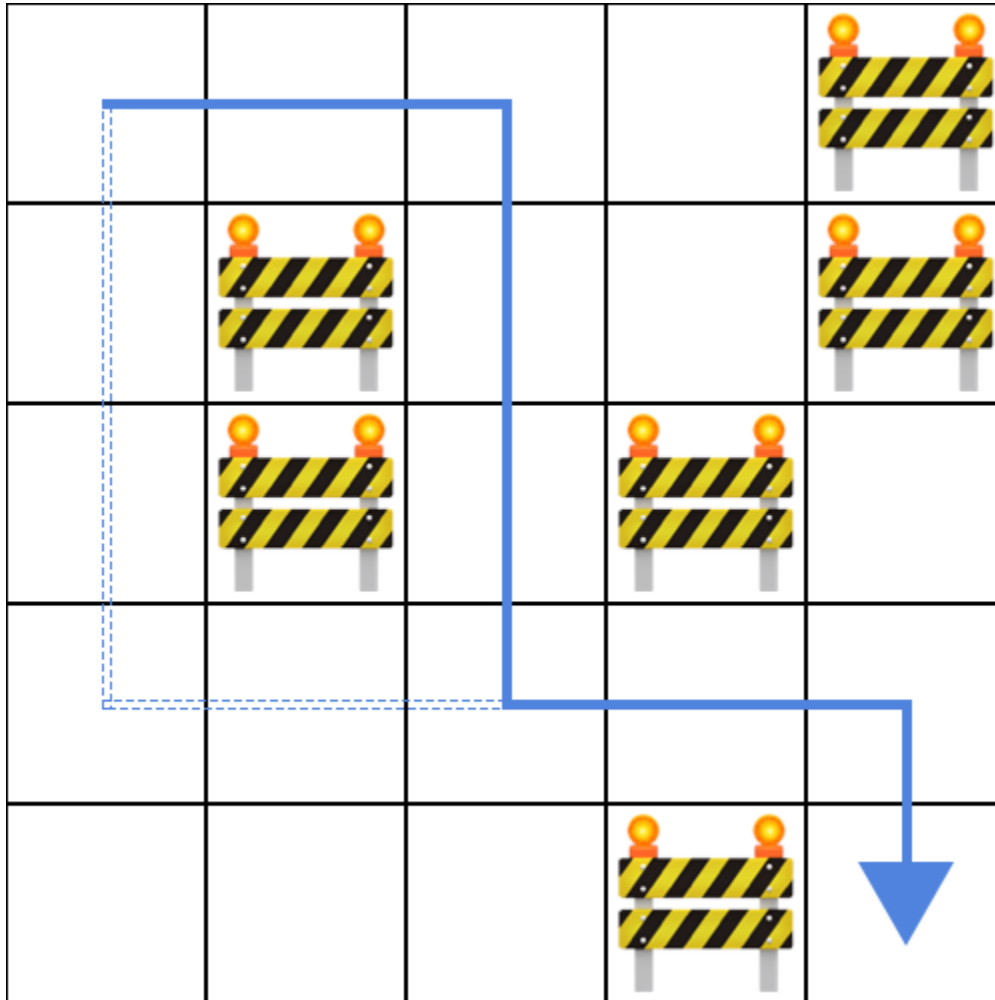
## Contoh Keluaran 1

```
....#
x#..#
.#.#.
.....
```

...#.

## Penjelasan Contoh 1

Contoh ini dapat diilustrasikan dengan gambar berikut, dengan garis padat merepresentasikan rute yang disiapkan oleh panitia, sedangkan garis putus-putus merepresentasikan rute alternatif.



Karenanya, setidaknya satu petak kosong harus diletakkan penghalang. Keluaran berikut juga merupakan keluaran yang benar.

....#

.#..#

.#.#.

.x...

...#.

## Contoh Masukan 2

```

7 6
>>>>vvvvvv
.....
.####.
...#.
###.#.
...#.
.####.
.....

```

## Contoh Keluaran 2

```

.....
.####.
...#.
###.#.
...#.
.####.
.....

```

## Penjelasan Contoh 2

Karena peserta dapat berpindah satu petak hanya ke arah selatan atau timur, peserta tidak dapat melewati jalan yang berbelok-belok pada contoh ini. Sehingga, tidak terdapat rute alternatif dan panitia tidak perlu meletakkan penghalang di petak kosong manapun.

## Penilaian

Keluaran Anda akan mendapatkan nilai pada sebuah kasus uji jika keluaran Anda mengikuti format keluaran yang disebutkan sebelumnya dan memenuhi seluruh syarat berikut:

- Jika  $S[i][j] = .$ , maka karakter ke- $j$  pada baris ke- $i$  dalam keluaran Anda adalah  $.$  atau  $x$ .
- Jika  $S[i][j] = \#$ , maka karakter ke- $j$  pada baris ke- $i$  dalam keluaran Anda adalah  $\#$ .
- Anda tidak meletakkan penghalang pada petak yang dilewati oleh rute yang disiapkan oleh panitia.
- Seluruh rute dari pintu masuk ke pintu keluar melewati himpunan petak yang persis sama dengan rute yang disiapkan oleh panitia.

Jika  $P$  adalah banyaknya penghalang tambahan yang Anda letakkan, dan  $Q$  adalah banyaknya penghalang tambahan minimum yang perlu diletakkan pada solusi optimal, maka nilai yang Anda dapatkan adalah:

| Kondisi                                       | Poin   |
|---|--|
| $P = Q$                                       | 10   |
| $Q < P \leq (R + C - 2)$                      | $\left[ 2 + 8 \times \left( \frac{(R + C - 2) - P}{(R + C - 2) - Q} \right)^4 \right]$ |
| $(R + C - 2) < P \leq (2 \times (R + C - 2))$ | 2  |
| $(2 \times (R + C - 2)) < P$                  | 1  |

## Kasus Uji

Untuk setiap kasus uji:

- $5 \leq R, C \leq 1.000$ .
- M hanya mengandung karakter  $>$  atau  $v$ .
- S hanya mengandung karakter  $.$  atau  $\#$ .
- Rute yang disiapkan oleh panitia dijamin hanya melewati petak kosong.

### rekreasi\_1.in

- $R = 6$ .
- $C = 6$ .
- S hanya mengandung karakter  $.$ .

### rekreasi\_2.in

- $R = 5$ .
- $C = 7$ .
- $M[i] = >$ , untuk  $1 \leq i < C$ .
- $M[i] = v$ , untuk  $C \leq i \leq R + C - 2$ .

### rekreasi\_3.in

- $R = 7$ .
- $C = 5$ .

### rekreasi\_4.in

- $R = 100$ .
- $C = 16$ .

### rekreasi\_5.in

- $R = 677$ .
- $C = 751$ .
- $M[i] = >$ , untuk  $1 \leq i < C$ .
- $M[i] = v$ , untuk  $C \leq i \leq R + C - 2$ .
- S hanya mengandung karakter  $.$ .

### rekreasi\_6.in

- $R = 93$ .
- $C = 78$ .

- S hanya mengandung karakter `>`.

### rekreasi\_7.in

- $R = 1.000$ .
- $C = 1.000$ .
- S hanya mengandung karakter `>`.

### rekreasi\_8.in

- $R = 100$ .
- $C = 99$ .
- $M[i] = >$ , untuk  $1 \leq i < C$ .
- $M[i] = v$ , untuk  $C \leq i \leq R + C - 2$ .

### rekreasi\_9.in

- $R = 853$ .
- $C = 814$ .
- $M[i] = >$ , untuk  $1 \leq i < C$ .
- $M[i] = v$ , untuk  $C \leq i \leq R + C - 2$ .

### rekreasi\_10.in

- $R = 1.000$ .
- $C = 1.000$ .
- $M[i] = >$ , untuk  $1 \leq i < C$ .
- $M[i] = v$ , untuk  $C \leq i \leq R + C - 2$ .

# Tempat Wisata

Time limit: 1 s

Memory limit: 256 MB

## Deskripsi

Terdapat  $M$  tempat wisata di kota Manado, dinomori 1 sampai  $M$ . Pada hari ini, terdapat  $N$  turis di kota Manado, dinomori 1 sampai  $N$ . Karena banyaknya tempat wisata, setiap turis mungkin hanya dapat mengunjungi beberapa tempat wisata saja hari ini. Turis ke- $i$  mengunjungi tempat wisata ke- $A[i]$ , ke- $(A[i] + 1)$ , ke- $(A[i] + 2)$ , ..., dan ke- $B[i]$ .

Seorang turis akan berkenalan dengan turis lain jika kedua turis ini mengunjungi setidaknya satu tempat wisata yang sama. Turis-turis ini ingin mengetahui berapa banyak turis yang akan mereka kenal, tidak termasuk diri mereka sendiri.

## Format Masukan

Masukan diberikan dalam format berikut:

```
N M
A[1] B[1]
A[2] B[2]
.
.
.
A[N] B[N]
```

## Format Keluaran

$N$  baris: baris ke- $i$  berisi banyaknya turis yang akan dikenal oleh turis ke- $i$ .

## Contoh Masukan 1

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

## Contoh Keluaran 1

2  
1  
1  
0

## Penjelasan Contoh 1

Pada contoh ini, turis pertama akan berkenalan dengan turis kedua dan ketiga.

## Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq N \leq 200.000$ .
- $1 \leq M \leq 10^9$ .
- $1 \leq A[i] \leq B[i] \leq M$ .

### Subsoal 1 (6 poin)

Hanya berisi kasus uji berikut:

8 11  
1 7  
4 11  
2 4  
9 10  
7 11  
2 4  
2 8  
8 8

### Subsoal 2 (8 poin)

Hanya berisi kasus uji berikut:

20 20  
1 20  
2 20  
3 18  
4 20  
5 20  
6 18

7 14  
8 16  
9 18  
10 20  
11 11  
12 12  
13 13  
14 14  
15 15  
16 16  
17 17  
18 18  
19 19  
20 20

**Subsoal 3 (13 poin)**

- $N \leq 200$ .
- $M \leq 400$ .

**Subsoal 4 (14 poin)**

- $N \leq 2.000$ .

**Subsoal 5 (22 poin)**

- $M \leq 400.000$ .

**Subsoal 6 (21 poin)**

- Tidak ada tempat wisata yang dikunjungi lebih dari dua turis.

**Subsoal 7 (16 poin)**

- Tidak ada batasan tambahan.



# Mencari Emas

Time limit: 1 s

Memory limit: 256 MB

## Deskripsi

Kota Manado dapat direpresentasikan sebagai petak-petak yang memiliki R baris (dinomori 1 sampai R) dan C kolom (dinomori 1 sampai C). Petak yang terletak pada baris ke-i dan kolom ke-j dinotasikan sebagai petak  $(i, j)$ .

Terdapat dua potongan emas di kota Manado, terletak pada petak  $(A_i, A_i)$  dan petak  $(B_i, B_i)$ . Kedua potongan emas mungkin saja terletak pada petak yang sama.

Pak Dengklek ingin mencari letak kedua potongan emas tersebut. Karenanya, Pak Dengklek membeli mesin pendeteksi emas. Untuk menggunakannya, Pak Dengklek dapat berdiri pada salah satu petak dan menekan tombol pada mesin pendeteksi emas tersebut. Jika Pak Dengklek berada pada petak  $(D_i, D_i)$  dan menekan tombol pada mesin pendeteksi emas, maka mesin tersebut akan memberikan jarak *Manhattan* ke potongan emas yang lebih dekat, yaitu nilai yang lebih kecil di antara  $|D_i - A_i| + |D_i - A_i|$  atau  $|D_i - B_i| + |D_i - B_i|$ .

Bantulah Pak Dengklek untuk mencari letak kedua potongan emas tersebut dengan menekan tombol pada mesin pendeteksi emas tidak lebih dari Q kali.

## Informasi Tipe Soal

Soal ini bertipe "interaktif". Pada soal ini Anda akan berinteraksi dengan program pengujian melalui standar masukan (stdin) dan standar keluaran (stdout). Perhatikan format interaksi di bawah ini dengan saksama.

## Format Interaksi

Pada awalnya, nilai R, C, dan Q akan diberikan dari standar masukan dalam format berikut:

```
R C Q
```

Kemudian, sebanyak tidak lebih dari Q kali, Anda dapat menghasilkan keluaran ke standar keluaran dalam format berikut:

```
? Di Dj
```

Ini berarti Pak Dengklek akan berdiri pada petak  $(D_i, D_j)$  dan menekan tombol pada mesin pendeteksi emas. Nilai  $D_i$  dan  $D_j$  harus memenuhi  $1 \leq D_i \leq R$  dan  $1 \leq D_j \leq C$ . Nilai yang diberitahukan oleh mesin pendeteksi emas (dinotasikan dengan nilai K) akan diberikan dari standar masukan dalam format berikut:

```
K
```

Pada akhirnya, jawaban Anda harus dikeluarkan ke standar keluaran dalam format berikut:

```
! Ai Aj Bi Bj
```

Ini berarti Anda menebak bahwa kedua potongan emas berada pada petak  $(A'_i, A'_j)$  dan petak  $(B'_i, B'_j)$ . Jawaban Anda akan dianggap benar jika setidaknya salah satu dari pernyataan berikut terpenuhi:

- $(A_i, A_i) = (A'_i, A'_j)$  dan  $(B_i, B_i) = (B'_i, B'_j)$ , atau
- $(A_i, A_i) = (B'_i, B'_j)$  dan  $(B_i, B_i) = (A'_i, A'_j)$ .

### Contoh Interaksi 1

Pada contoh berikut,  $R = 4, C = 5, Q = 5, A_i = 2, A_j = 2, B_i = 3, B_j = 5$ .

| Standar masukan | Standar keluaran |
|-----------------|------------------|
| 4 5 5           |                  |
|                 | ? 2 3            |
| 1               |                  |
|                 | ? 1 3            |
| 2               |                  |
|                 | ? 2 4            |
| 2               |                  |
|                 | ? 3 5            |
| 0               |                  |
|                 | ! 3 5 2 2        |

### Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq R, C \leq 10^9$ .
- $1 \leq A_i, B_i \leq R$ .

- $1 \leq A_i, B_i \leq C$ .

### Subsoal 1 (13 poin)

- $1 \leq R, C \leq 100$ .
- $A_i = B_i$ .
- $A_i = B_i$ .
- $Q = 100$ .

### Subsoal 2 (21 poin)

- $A_i = B_i$ .
- $A_i = B_i$ .
- $Q = 5$ .

### Subsoal 3 (8 poin)

- $1 \leq R, C \leq 10$ .
- $Q = 100$ .

### Subsoal 4 (20 poin)

- $1 \leq R, C \leq 50$ .
- $Q = 100$ .

### Subsoal 5 (11 poin)

- $R = 1$ .
- $Q = 5$ .

### Subsoal 6 (27 poin)

- $Q = 5$ .

## Catatan

Untuk setiap kasus uji, notasikan  $P$  sebagai banyaknya penekanan tombol pada mesin pendeteksi emas. Jika Anda menebak letak kedua potongan emas dengan benar dan  $P \leq Q$ , maka Anda akan mendapatkan verdict **AC**. Jika  $P > Q$ , maka Anda akan mendapatkan verdict **WA** dan pesan " $P > Q$ " pada grader. Jika Anda tidak menebak letak kedua potongan emas dengan benar, atau nilai  $D_i$  dan  $D_j$  yang Anda keluarkan tidak memenuhi  $1 \leq D_i \leq R$  dan  $1 \leq D_j \leq C$ , maka Anda akan mendapatkan verdict **WA** dan tidak ada pesan pada grader.

## Peringatan

Selalu lakukan *flush* setiap kali setelah program Anda mengeluarkan keluaran.

- Pascal:

```
flush(output);
```

- C/C++:

- untuk library `stdio.h` (`cstdio`):

```
fflush(stdout);
```

- untuk library `iostream`:

```
std::cout << std::flush;
```

# Penyebaran Hoaks

Time limit: 2 s

Memory limit: 256 MB

## Deskripsi

Pak Dengklek baru saja mengembangkan situs sosial media. Terdapat  $N$  pengguna pada situs ini, dinomori 1 sampai  $N$ . Terdapat  $S$  jam dalam satu hari. Seluruh pengguna ini memiliki jadwal penggunaan yang tetap dari hari ke hari. Sehingga, setiap harinya pengguna ke- $i$  akan terhubung (*online*) pada situs ini sebanyak  $T[i]$  kali:

- dari awal jam ke- $A[i][1]$  sampai akhir jam ke- $B[i][1]$ ,
- dari awal jam ke- $A[i][2]$  sampai akhir jam ke- $B[i][2]$ ,
- ...
- dan dari awal jam ke- $A[i][T[i]]$  sampai akhir jam ke- $B[i][T[i]]$ ,

Pada setiap saat, seluruh pengguna pada situs ini senang berbagi berita kepada seluruh pengguna lainnya yang sedang terhubung. Celakanya, salah satu dari  $N$  pengguna ini memiliki hoaks pada awal hari pertama dan akan menyebarkannya. Karenanya, seluruh pengguna yang bertemu dengan pengguna ini juga akan memiliki hoaks pada akhir hari pertama. Dua pengguna dinyatakan bertemu jika kedua pengguna tersebut terhubung pada setidaknya satu jam yang sama.

Hoaks ini kemudian tersebar juga pada hari kedua. Sehingga, seluruh pengguna yang bertemu dengan pengguna yang memiliki hoaks pada akhir hari pertama juga akan memiliki hoaks pada akhir hari kedua. Hal ini berlanjut hari-hari berikutnya.

Terdapat  $Q$  skenario, dinomori 1 sampai  $Q$ . Pada skenario ke- $i$ , pengguna yang pada awal hari pertama memiliki hoaks adalah pengguna ke- $P[i]$ . Skenario yang berbeda mungkin saja menyebabkan penyebaran hoaks yang berbeda. Karenanya, untuk setiap skenario, Pak Dengklek penasaran akan banyaknya pengguna yang memiliki hoaks pada akhir hari ke- $N$ , dengan  $N$  adalah banyaknya peserta.

## Format Masukan

Masukan diberikan dalam format berikut:

```
N S Q
T[1] A[1][1] B[1][1] A[1][2] B[1][2] ... A[1][T[1]] B[1][T[1]]
T[2] A[2][1] B[2][1] A[2][2] B[2][2] ... A[2][T[2]] B[2][T[2]]
.
.
.
T[N] A[N][1] B[N][1] A[N][2] B[N][2] ... A[N][T[N]] B[N][T[N]]
P[1]
P[2]
.
```

•  
•  
P[Q]

## Format Keluaran

Q baris: baris ke-i berisi banyaknya pengguna yang memiliki hoaks pada akhir hari ke-N pada skenario ke-i.

## Contoh Masukan 1

```
5 10 2
2 2 4 7 9
1 1 3
1 1 1
1 9 10
1 5 6
1
5
```

## Contoh Keluaran 1

```
4
1
```

## Penjelasan Contoh 1

Pada skenario pertama, pengguna pertama bertemu dengan pengguna kedua (pada jam kedua sampai jam ketiga) dan pengguna keempat (pada jam kesembilan). Sehingga, pada akhir hari pertama, kedua pengguna tersebut akan memiliki hoaks. Pengguna kedua juga bertemu dengan pengguna ketiga (pada jam pertama). Sehingga, pada akhir hari kedua, pengguna ketiga juga akan memiliki hoaks. Tidak akan ada penyebaran hoaks pada hari ketiga, keempat, dan kelima, sehingga 4 pengguna akan memiliki hoaks pada akhir hari kelima.

Pada skenario kedua, pengguna kelima tidak bertemu dengan pengguna lainnya, sehingga pengguna lainnya tidak akan memiliki hoaks.

## Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq Q \leq N \leq 200.000$ .
- $1 \leq S \leq 10^9$ .
- $1 \leq T[i] \leq S$ .

- Jumlah  $T[i]$  untuk seluruh  $1 \leq i \leq N$  tidak melebihi 200.000.
- $1 \leq A[i][j] \leq B[i][j] \leq S$ .
- $B[i][j - 1] < A[i][j]$ , untuk  $2 \leq j \leq T[i]$ .
- $1 \leq P[i] \leq N$ .
- $P[i - 1] < P[i]$ , untuk  $2 \leq i \leq Q$ .

### Subsoal 1 (7 poin)

Hanya berisi kasus uji berikut:

```
5 10 2
2 1 3 9 10
1 5 6
1 5 6
1 4 4
2 4 4 7 9
1
2
```

### Subsoal 2 (8 poin)

Hanya berisi kasus uji berikut:

```
10 10 5
5 2 2 4 4 6 6 8 8 10 10
1 2 2
1 3 3
1 4 4
1 5 5
1 6 6
1 7 7
1 8 8
1 9 9
1 10 10
1
5
6
8
9
```

**Subsoal 3 (12 poin)**

- $N, S, Q \leq 50$ .
- Jumlah  $T[i]$  untuk seluruh  $1 \leq i \leq N$  tidak melebihi 50.

**Subsoal 4 (14 poin)**

- $N, Q \leq 50$ .
- Jumlah  $T[i]$  untuk seluruh  $1 \leq i \leq N$  tidak melebihi 50.

**Subsoal 5 (18 poin)**

- $N, Q \leq 200$ .
- Jumlah  $T[i]$  untuk seluruh  $1 \leq i \leq N$  tidak melebihi 200.

**Subsoal 6 (23 poin)**

- $N, Q \leq 2.000$ .
- Jumlah  $T[i]$  untuk seluruh  $1 \leq i \leq N$  tidak melebihi 2.000.

**Subsoal 7 (18 poin)**

- Tidak ada batasan tambahan.