

# Jembatan Layang

## Deskripsi

Di kota Padang, terdapat sebuah kampus terkenal bernama Universitas Dengklek. Seluruh mahasiswa kampus tersebut tinggal pada sebuah gedung asrama setinggi  $N$  lantai. Tahun ini, pihak kampus selesai membangun gedung asrama baru yang juga setinggi  $N$  lantai.

Terdapat  $K$  buah jembatan layang di antara kedua gedung tersebut. Jembatan ke- $i$  terletak pada ketinggian lantai  $H[i]$ . Uniknya, setiap lantai dari masing-masing gedung hanya terhubung pada tepat satu jembatan melalui sebuah terowongan khusus. Lantai  $i$  pada gedung lama hanya terhubung pada jembatan ke- $A[i]$ , dan lantai  $i$  pada gedung baru hanya terhubung pada jembatan ke- $B[i]$ .

Sekarang, pihak kampus akan merelokasi seluruh mahasiswa ke gedung baru. Setiap lantai di gedung lama akan direlokasi ke sebuah lantai di gedung baru. Tidak ada dua lantai di gedung lama yang direlokasi ke lantai yang sama di gedung baru. Dengan kata lain, pihak kampus ingin mencari sebuah permutasi  $P$  dari  $\{1, 2, \dots, N\}$  yang menyatakan bahwa penghuni lantai  $i$  di gedung lama akan direlokasi ke lantai  $P[i]$  di gedung baru.

Untuk alasan keamanan dalam proses relokasi, pihak kampus menerapkan aturan berikut untuk setiap relokasi lantai:

- Penghuni lantai di gedung lama akan menuju ke satu-satunya jembatan yang terhubung, kemudian menyusuri jembatan menuju gedung baru, lalu menuju ke salah satu lantai di gedung baru yang terhubung pada jembatan tersebut.
- Lantai di gedung lama dan lantai di gedung baru tidak boleh keduanya sekaligus lebih tinggi atau sekaligus lebih rendah daripada jembatan yang menghubungkannya.

Atau dengan kata lain, secara formal, untuk setiap  $i$ , seluruh syarat berikut harus dipenuhi:

- $A[i] = B[P[i]]$
- Jika  $i > H[A[i]]$ , maka harus berlaku  $P[i] \leq H[A[i]]$ .
- Jika  $i < H[A[i]]$ , maka harus berlaku  $P[i] \geq H[A[i]]$ .

Sekarang pihak kampus penasaran, sebenarnya ada berapa permutasi berbeda yang mungkin? Karena hasilnya bisa sangat besar, mereka hanya penasaran pada hasilnya modulo  $1.000.000.007$ .

## Format Masukan

Baris pertama akan berisi "label kasus uji". Label kasus uji adalah sebuah string yang dijelaskan sebagai berikut:

- Panjang string tersebut adalah banyaknya subsoal ditambah satu.
- Karakter ke- $0$  (indeks dimulai dari  $0$ ) akan berisi  $0$  jika kasus uji tersebut merupakan contoh kasus uji, atau berisi  $'.'$  (titik) jika bukan.
- Untuk setiap nilai  $i$  di antara  $1$  hingga banyaknya subsoal, berlaku:
  - jika kasus uji tersebut memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi  $i$ , atau
  - jika kasus uji tersebut tidak memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi karakter  $'.'$  (titik).

Sebagai contoh, apabila label sebuah kasus uji sebuah soal adalah 0. .345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Baris-baris berikutnya diberikan dalam format berikut:

```
N K
H[1] H[2] .. H[K]
A[1] A[2] A[3] .. A[N]
B[1] B[2] A[3] .. B[N]
```

## Format Keluaran

Sebuah baris berisi banyaknya permutasi yang memenuhi syarat, modulo 1.000.000.007.

## Contoh Masukan 1

```
0...45.7
6 1
3
1 1 1 1 1 1
1 1 1 1 1 1
```

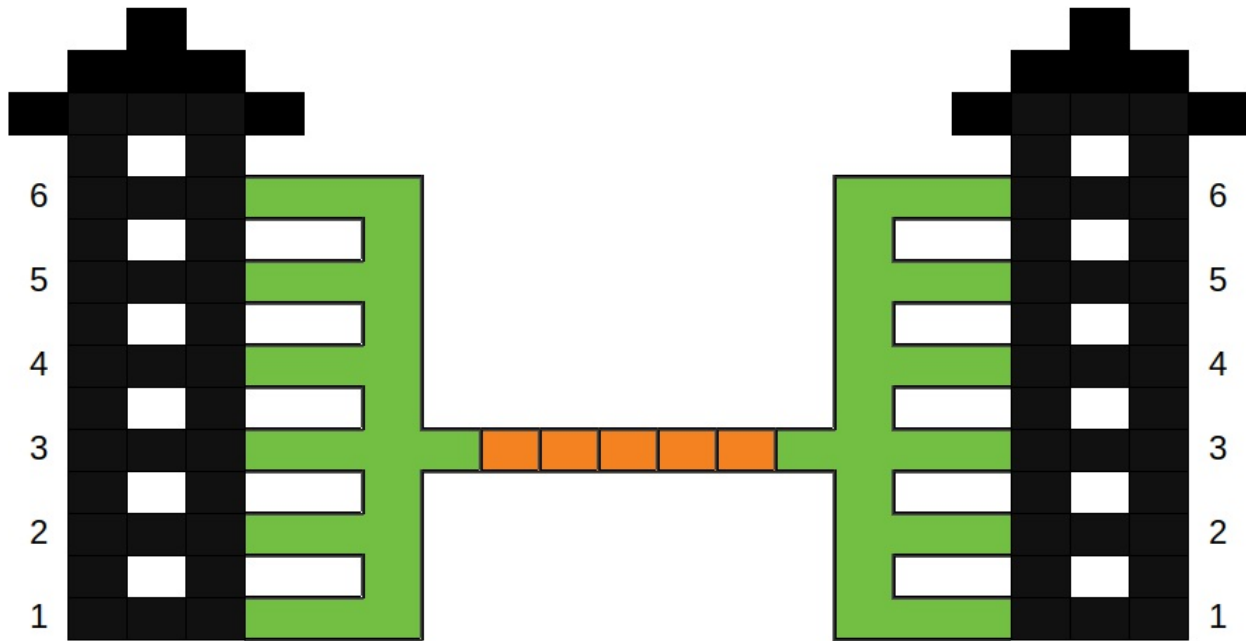
## Contoh Keluaran 1

```
36
```

## Penjelasan Contoh 1

Terdapat 6 lantai, dan 1 jembatan layang yang terletak pada ketinggian lantai 3. Setiap lantai di gedung lama maupun gedung baru, terhubung pada jembatan.

Contoh ini diilustrasikan oleh gambar berikut. Gedung sebelah kiri adalah gedung lama, sedangkan yang sebelah kanan adalah gedung baru.



Menurut aturan yang diberikan, maka:

- Lantai 1 di gedung lama hanya bisa direlokasi ke lantai 3, 4, 5, atau 6 di gedung baru.
- Lantai 2 di gedung lama hanya bisa direlokasi ke lantai 3, 4, 5, atau 6 di gedung baru.
- Lantai 3 di gedung lama bisa direlokasi ke seluruh lantai di gedung baru.
- Lantai 4 di gedung lama hanya bisa direlokasi ke lantai 1, 2, atau 3 di gedung baru.
- Lantai 5 di gedung lama hanya bisa direlokasi ke lantai 1, 2, atau 3 di gedung baru.
- Lantai 6 di gedung lama hanya bisa direlokasi ke lantai 1, 2, atau 3 di gedung baru.

Perhatikan bahwa lantai-lantai 4, 5, dan 6 di gedung lama pasti direlokasi ke lantai-lantai 1, 2, dan 3 di gedung baru. Terdapat  $3! = 6$  cara. Sisanya, lantai-lantai 1, 2, 3 di gedung lama bisa direlokasi ke lantai-lantai 4, 5, 6 di gedung baru. Terdapat  $3! = 6$  cara. Dengan demikian, terdapat total  $6 \times 6 = 36$  permutasi yang mungkin.

## Contoh Masukan 2

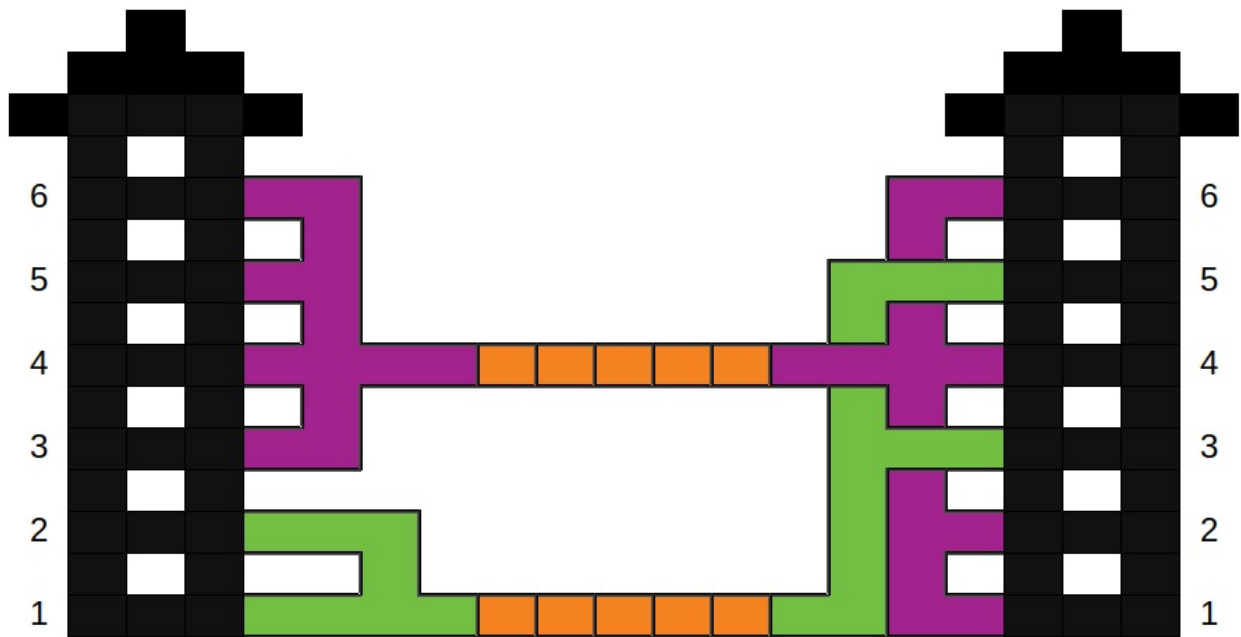
```
0...4..7
6 2
1 4
1 1 2 2 2 2
2 2 1 2 1 2
```

## Contoh Keluaran 2

```
0
```

## Penjelasan Contoh 2

Contoh ini diilustrasikan oleh gambar berikut.



Perhatikan bahwa lantai 2 di gedung lama terhubung pada jembatan yang berada pada ketinggian lantai 1 (lantai lebih tinggi daripada jembatan), namun hanya lantai 3 dan 5 di gedung baru yang terhubung pada jembatan tersebut (yang juga lebih tinggi daripada jembatan). Pada kasus ini, lantai 2 tidak dapat direlokasi ke manapun tanpa melanggar aturan yang diberikan, sehingga tidak ada permutasi yang mungkin.

### Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq K \leq N \leq 100.000$
- $1 \leq H[i] \leq N$
- Nilai-nilai  $H[i]$  berbeda-beda.
- $1 \leq A[i], B[i] \leq K$
- $A[i]$  mengandung setiap bilangan dari 1 hingga  $K$ .
- $B[i]$  mengandung setiap bilangan dari 1 hingga  $K$ .
- Untuk setiap jembatan layang, banyaknya lantai di gedung lama yang terhubung, sama dengan banyaknya lantai di gedung baru yang terhubung.

#### Subsoal 1 (6 poin):

Hanya berisi kasus uji berikut:

```
.1..45.7
5 1
3
1 1 1 1 1
1 1 1 1 1
```

#### Subsoal 2 (8 poin):

Hanya berisi kasus uji berikut:

..2.4..7

9 4

2 3 5 8

2 2 1 3 3 2 3 4 4

2 1 3 2 3 3 2 4 4

**Subsoal 3 (10 poin):**

- $K = N$

**Subsoal 4 (15 poin):**

- $N \leq 10$

**Subsoal 5 (23 poin):**

- $K = 1$

**Subsoal 6 (18 poin):**

- Untuk setiap  $i$ ,  $H[A[i]] \neq i$
- Untuk setiap  $i$ ,  $H[B[i]] \neq i$

**Subsoal 7 (20 poin):**

- Tidak ada batasan tambahan.

# Pertahanan Padang

## Deskripsi

Kota Padang terkenal dengan "sajak biner"-nya. Sebuah sajak biner adalah sebuah string  $S$  yang memenuhi seluruh syarat berikut:

- $S$  yang hanya terdiri atas karakter '0' atau '1'.
- $S$  terdiri atas setidaknya 1 karakter.
- $S[0] = '1'$ . (Indeks string dimulai dari 0.)

Sebuah "sajak biner bergantian" adalah sebuah string  $S$  yang memenuhi seluruh syarat berikut:

- $S$  adalah sajak biner.
- $S[i]$  dan  $S[i-1]$  berbeda, untuk  $i > 0$ .

Sebagai contoh, "1", "10", "101", dan "1010" adalah sajak-sajak biner bergantian, sedangkan "", "0", "01", "010", dan "100" bukan.

Suatu hari, kota Padang tengah diserang sesosok makhluk jahat! Pak Dengklek, sang penguasa kota Padang, sedang berunding dengan makhluk jahat tersebut agar ia segera pergi. Rupanya, makhluk jahat tersebut hanya ingin dibuatkan sebuah sajak biner  $S$  yang memenuhi seluruh berikut:

- $S$  mengandung tepat  $A$  buah subsekuens yang berupa sajak biner bergantian dengan panjang ganjil.
- $S$  mengandung tepat  $B$  buah subsekuens yang berupa sajak biner bergantian dengan panjang genap.

Sebuah subsekuens dari  $S$  dihasilkan dengan membuang nol atau lebih karakter dari  $S$ . Sebagai contoh, sajak biner "1101" memiliki tepat 7 buah subsekuens sajak biner bergantian: "1" (3 buah), "10" (2 buah), dan "101" (2 buah).

Karena mungkin saja terdapat banyak sajak biner yang memenuhi syarat, makhluk jahat tersebut ingin sajak biner yang terkecil secara leksikografis (lihat bagian Catatan).

Karena sajak biner  $S$  bisa panjang sekali, makhluk jahat tersebut memberikan keringanan dengan hanya akan memberikan  $Q$  buah pertanyaan. Pertanyaan ke- $i$  berupa: apakah karakter-karakter  $S$  dari indeks  $L[i]$  sampai dengan  $R[i]$ , inklusif?

Bantulah Pak Dengklek untuk menjawab pertanyaan-pertanyaan tersebut!

## Catatan

Urutan leksikografi didefinisikan sebagai berikut: string  $U$  dikatakan lebih kecil secara leksikografis daripada string  $V$  apabila  $U$  merupakan prefiks dari  $V$ , atau pada indeks terkecil  $i$  yang mana  $U[i]$  berbeda dengan  $V[i]$ , berlaku  $U[i] < V[i]$ .

## Format Masukan

Baris pertama akan berisi "label kasus uji". Label kasus uji adalah sebuah string yang dijelaskan sebagai berikut:

- Panjang string tersebut adalah banyaknya subsoal ditambah satu.

- Karakter ke-0 (indeks dimulai dari 0) akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, atau berisi '.' (titik) jika bukan.
- Untuk setiap nilai i di antara 1 hingga banyaknya subsoal, berlaku:
  - jika kasus uji tersebut memenuhi batasan subsoal ke-i, maka karakter ke-i berisi i, atau
  - jika kasus uji tersebut tidak memenuhi batasan subsoal ke-i, maka karakter ke-i berisi karakter '.' (titik).

Sebagai contoh, apabila label sebuah kasus uji sebuah soal adalah 0. .345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Baris-baris berikutnya diberikan dalam format berikut:

```
A B Q
L[1] R[1]
L[2] R[2]
.
.
L[Q] R[Q]
```

## Format Keluaran

Apabila tidak ada sajak biner S yang memenuhi syarat, keluarkan sebuah baris berisi:

```
TIDAK MUNGKIN
```

Apabila ada, keluarkan:

```
MUNGKIN
T[1]
T[2]
.
.
T[Q]
```

dengan T[i] adalah jawaban dari pertanyaan ke-i, apabila L[i] dan R[i] adalah indeks yang terdapat pada S, atau DI LUAR BATAS apabila tidak.

## Contoh Masukan 1

```
0. . . . 56.8
5 2 4
0 3
1 2
3 3
0 4
```

## Contoh Keluaran 1

```
MUNGKIN
1101
10
1
DI LUAR BATAS
```

## Penjelasan Contoh 1

Sajak biner terkecil secara leksikografis yang memenuhi syarat adalah  $S = "1101"$ :

- Terdapat  $A = 5$  subsekuens berupa sajak biner bergantian dengan panjang ganjil: "1" (3 buah) dan "101" (2 buah).
- Terdapat  $B = 2$  subsekuens berupa sajak biner bergantian dengan panjang genap: "10" (2 buah).

## Contoh Masukan 2

```
0...56.8
3 2 4
0 3
1 2
3 3
0 4
```

## Contoh Keluaran 2

```
TIDAK MUNGKIN
```

## Subsoal

Untuk semua subsoal, berlaku:

- $0 \leq A, B \leq 10^{15}$
- $A + B > 0$
- $0 \leq Q \leq 20.000$
- $0 \leq L[i] \leq R[i] < 10^{16}$
- $R[i] - L[i] \leq 50$

### Subsoal 1 (10 poin):

Hanya berisi kasus uji berikut:

```
.1..456.8
3 9 1
1 4
```

### Subsoal 2 (10 poin):

Hanya berisi kasus uji berikut:



..2..56.8

25 3 4

0 7

0 8

0 9

0 10

**Subsoal 3 (10 poin):**

- $A + B \leq 8$
- $Q = 1$

**Subsoal 4 (11 poin):**

- $A + B \leq 18$
- $Q = 1$

**Subsoal 5 (15 poin):**

- $A, B \leq 2.000$

**Subsoal 6 (20 poin):**

- $A, B \leq 1.000.000$

**Subsoal 7 (14 poin):**

- $Q = 0$

**Subsoal 8 (10 poin):**

- Tidak ada batasan tambahan.

# Ojek Daring

## Deskripsi

Terdapat  $V$  kota di Sumatra Barat, dinomori 1 hingga  $V$ . Terdapat  $E$  buah ruas jalan dua arah yang menghubungkan kota-kota, dinomori 1 hingga  $E$ . Ruas jalan ke- $i$  menghubungkan kota  $X[i]$  dengan  $Y[i]$ , dan memiliki jarak  $K[i]$  km. Setiap pasang kota terhubung oleh paling banyak satu ruas jalan. Tidak ada ruas jalan yang menghubungkan sebuah kota dengan dirinya sendiri.

Hanya terdapat dua moda transportasi: ojek pangkalan (lokal) dan ojek daring (*online*). Keduanya dapat Anda pesan melalui telepon genggam Anda, yang akan menghampiri Anda di manapun Anda berada.

Terdapat beberapa aturan pemesanan ojek, bergantung pada lokasi Anda sekarang dan jenis ojek, sebagaimana dijelaskan berikut:

- Jika Anda sedang berada tepat pada suatu kota (yakni, tepat pada suatu ujung dari suatu ruas jalan), maka Anda dapat memesan dan mulai menaiki ojek pangkalan maupun ojek daring.
- Jika Anda sedang berada pada bagian manapun dari suatu ruas jalan yang memiliki jarak berupa bilangan bulat dari kedua ujungnya (selanjutnya akan disebut "bagian bulat"), selain pada kota sebagaimana dijelaskan di atas, maka Anda:
  - dapat memesan dan mulai menaiki ojek pangkalan.
  - dapat memesan dan mulai menaiki ojek daring, hanya apabila ruas jalan tersebut TIDAK dikuasai oleh ojek pangkalan. Ruas jalan ke- $i$  dikuasai oleh ojek pangkalan apabila  $Q[i] = 1$ .

Setelah Anda memesan ojek dan menaikinya, untuk kedua jenis ojek, Anda boleh melewati kota dan jalan manapun, kemudian dapat turun di kota manapun maupun bagian bulat manapun dari ruas jalan manapun.

Untuk ojek pangkalan, tarif sekali naik adalah  $C_p$  (yakni, jauh-dekat harga sama), dan jarak maksimum dalam satu perjalanan adalah  $M_p$ .

Untuk ojek daring, tarif sekali naik adalah  $C_d \times$  (jarak yang ditempuh), dan jarak maksimum dalam satu perjalanan adalah  $M_d$ .

Anda ingin pergi dari kota A menuju kota B, dengan hanya menaiki ojek (tanpa jalan kaki). Tentukan tarif termurah yang bisa Anda capai!

## Format Masukan

Baris pertama akan berisi "label kasus uji". Label kasus uji adalah sebuah string yang dijelaskan sebagai berikut:

- Panjang string tersebut adalah banyaknya subsoal ditambah satu.
- Karakter ke-0 (indeks dimulai dari 0) akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, atau berisi '.' (titik) jika bukan.
- Untuk setiap nilai  $i$  di antara 1 hingga banyaknya subsoal, berlaku:
  - jika kasus uji tersebut memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi  $i$ , atau

- o jika kasus uji tersebut tidak memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi karakter '.' (titik).

Sebagai contoh, apabila label sebuah kasus uji sebuah soal adalah 0. .345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Baris-baris berikutnya diberikan dalam format berikut:

```
V E
C_d M_d
C_p M_p
A B
X[1] Y[1] K[1] Q[1]
X[2] Y[2] K[2] Q[2]
.
.
X[E] Y[E] K[E] Q[E]
```

## Format Keluaran

Sebuah baris berisi tarif termurah untuk pergi dari kota A ke kota B.

### Contoh Masukan 1

```
0.....67
6 6
2 6
4 2
5 2
1 2 14 1
1 3 4 0
3 4 8 0
4 2 12 0
5 6 1 0
6 1 1 0
```

### Contoh Keluaran 1

```
32
```

### Penjelasan Contoh 1

Cara termurah adalah sebagai berikut:

1. Sekali naik ojek pangkalan dari kota 5 menuju kota 6 lalu kota 1 (total sejauh 2 km). Tarif = 4.
2. Sekali naik ojek daring menuju kota 2 sejauh 6 km. Tarif =  $2 \times 6 = 12$ .
3. Empat kali naik ojek pangkalan menuju kota 2 masing-masing sejauh 2 km. Tarif =  $4 \times 4 = 16$ .

Total tarif adalah  $4 + 12 + 16 = 32$ .

### Contoh Masukan 2

```
0....567
7 9
1 2
2 1
1 7
1 2 1 1
2 3 1 1
3 1 1 1
3 4 1 1
4 5 1 1
5 3 1 1
5 6 1 1
6 7 1 1
7 5 1 1
```

### Contoh Keluaran 2

```
3
```

### Penjelasan Contoh 2

Cara termurah adalah sebagai berikut:

1. Sekali naik ojek daring dari kota 1 menuju kota 3 lalu kota 5 (total sejauh 2 km). Tarif =  $1 \times 2 = 2$ .
2. Sekali naik ojek daring menuju kota 7 (sejauh 1 km). Tarif =  $1 \times 1 = 1$ .

Total tarif adalah  $2 + 1 = 3$ .

### Contoh Masukan 3

```
0....5.7
2 1
100 3
1 100
1 2
1 2 6 1
```

### Contoh Keluaran 3

```
1
```

### Penjelasan Contoh 3

Cara termurah adalah dengan sekali naik ojek pangkalan dari kota 1 menuju kota 2 (sejauh 6 km). Tarif = 1.

### Subsoal

Untuk semua subsoal, berlaku:

- $2 \leq V \leq 100$
- $V - 1 \leq E \leq V \times (V - 1) / 2$
- $1 \leq C_p, C_d \leq 10^8$
- $1 \leq M_p, M_d \leq 100$
- $1 \leq A, B, X[i], Y[i] \leq V$
- $A \neq B$
- Untuk setiap  $i$ ,  $X[i] \neq Y[i]$
- $1 \leq K[i] \leq 10^8$
- $0 \leq Q[i] \leq 1$
- Setiap pasang kota terhubung oleh paling banyak satu ruas jalan.
- Dijamin selalu terdapat cara untuk pergi dari setiap kota ke semua kota lainnya, melalui kota-kota dan jalan-jalan.

**Subsoal 1 (8 poin):**

Hanya berisi kasus uji berikut:

```
.1....67
5 4
3 3
10 4
1 5
1 2 1 0
2 3 100 0
3 4 100000 1
4 5 100000000 0
```

**Subsoal 2 (10 poin):**

Hanya berisi kasus uji berikut:

```
..2.4567
10 15
3 1
5 1
1 10
1 4 4 1
1 2 1 0
2 3 2 0
2 4 2 0
3 4 2 0
3 5 2 0
4 5 2 0
3 8 9 0
5 6 1 0
6 7 1 0
7 5 1 0
6 8 3 1
8 9 3 0
7 9 3 0
9 10 4 0
```

**Subsoal 3 (11 poin):**

- $C_d = C_p = 1$
- $M_d = M_p = 1$
- $K[i] = 1$

**Subsoal 4 (13 poin):**

- $M_d = M_p = 1$

**Subsoal 5 (16 poin):**

- $K[i] \leq 10$

**Subsoal 6 (18 poin):**

- $M_d, M_p \leq 10$

**Subsoal 7 (24 poin):**

- Tidak ada batasan tambahan.

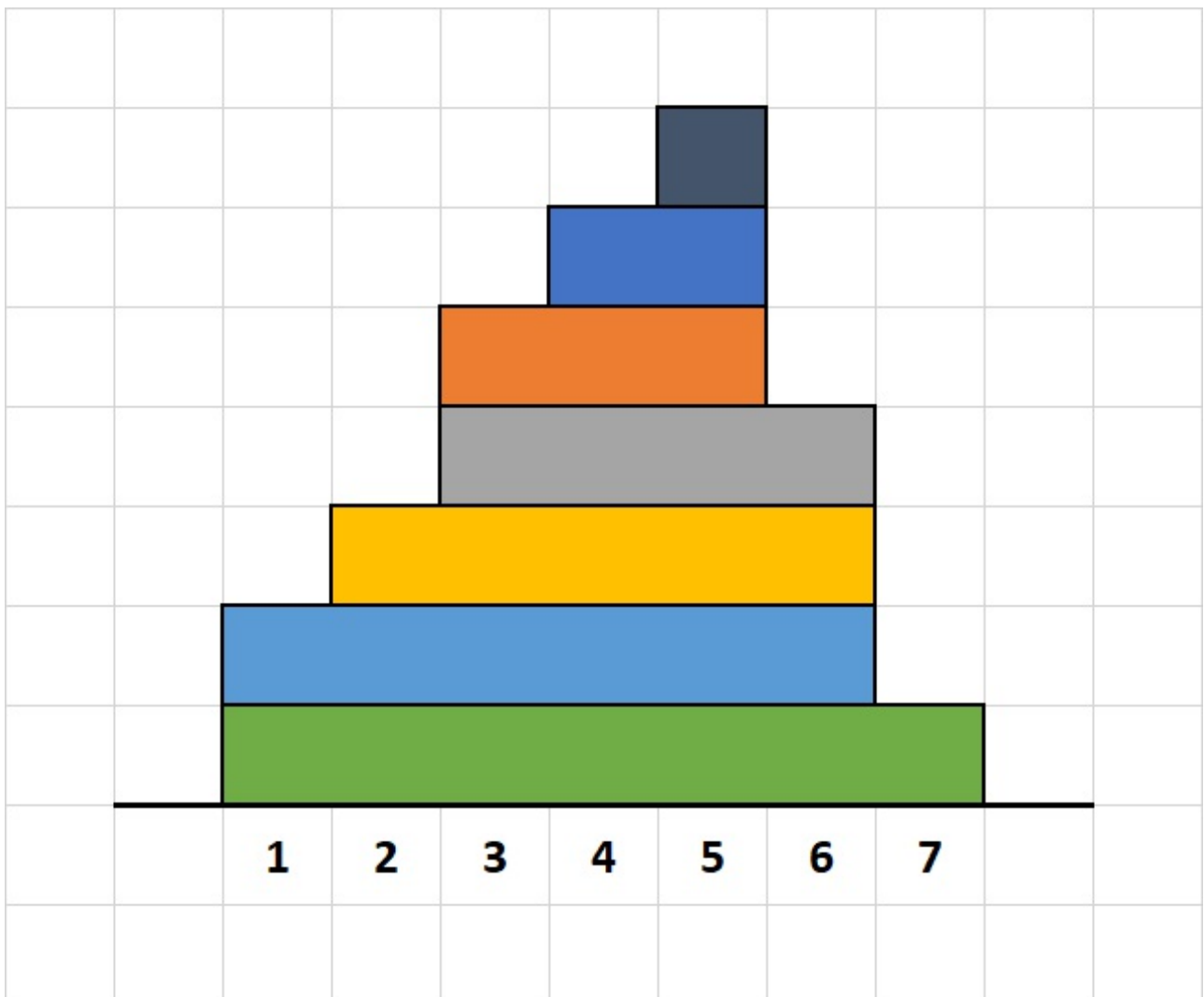
# Hiasan Atap

## Deskripsi

Pak Dengklek ingin membuat hiasan untuk atap rumahnya. Ia memiliki  $N$  buah balok berukuran  $1 \times N, 1 \times (N-1), \dots, 1 \times 1$ . Ia akan memasang balok-balok tersebut secara berurutan dari yang paling panjang ke yang paling pendek.

Pertama-tama, Pak Dengklek meletakkan balok terpanjang sedemikian sehingga balok tersebut berada pada kolom 1 hingga  $N$ . Kemudian, ia akan meletakkan setiap balok selanjutnya di atas tepat balok sebelumnya, dengan salah satu dari dua cara berikut: rata kiri atau rata kanan.

Sebagai contoh, misalkan  $N = 7$ . Mulanya, Pak Dengklek akan meletakkan balok  $1 \times 7$  pada kolom 1 hingga 7. Misalkan untuk 6 balok berikutnya, Pak Dengklek secara berturut-turut meletakkan rata kiri, kanan, kanan, kiri, kanan, kanan. Maka, hasil akhir hiasan tersebut adalah sebagai berikut:



Karena Pak Dengklek memiliki sangat banyak balok, maka konfigurasi peletakan balok dinyatakan sebagai string  $S$ , yang disambung sebanyak  $K$  kali. Dengan demikian,  $N = |S| \times K + 1$ . Karakter ke- $i$  pada string akhir adalah cara peletakan balok ke- $(i+1)$ , dengan karakter 'A' menyatakan rata kiri dan karakter 'B' menyatakan rata kanan. Sebagai contoh, konfigurasi peletakan contoh di atas dapat dinyatakan dengan  $S = "ABB"$  dan  $K = 2$ .

Pak Dengklek sekarang tertarik untuk mengetahui ketinggian balok tertinggi pada suatu rentang kolom. Sebagai contoh, dari kolom 1 hingga 7, ketinggiannya adalah 7 (balok tertinggi berada pada kolom 5). Contoh lainnya, dari kolom 2 hingga 4, ketinggiannya adalah 6 (balok tertinggi berada pada kolom 4).

Pak Dengklek memiliki  $Q$  pertanyaan. Pertanyaan ke- $i$  menanyakan ketinggian balok tertinggi dari kolom  $L[i]$  hingga  $R[i]$ , inklusif. Bantulah ia untuk menjawab pertanyaan-pertanyaan tersebut!

## Format Masukan

Baris pertama akan berisi "label kasus uji". Label kasus uji adalah sebuah string yang dijelaskan sebagai berikut:

- Panjang string tersebut adalah banyaknya subsoal ditambah satu.
- Karakter ke-0 (indeks dimulai dari 0) akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, atau berisi '.' (titik) jika bukan.
- Untuk setiap nilai  $i$  di antara 1 hingga banyaknya subsoal, berlaku:
  - jika kasus uji tersebut memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi  $i$ , atau
  - jika kasus uji tersebut tidak memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi karakter '.' (titik).

Sebagai contoh, apabila label sebuah kasus uji sebuah soal adalah 0. .345, maka:

- Soal tersebut memiliki 5 buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke-3, ke-4, dan ke-5.

Baris-baris berikutnya diberikan dalam format berikut:

```
S
K Q
L[1] R[1]
L[2] R[2]
.
.
L[Q] R[Q]
```

## Format Keluaran

$Q$  buah baris, masing-masing berisi jawaban dari setiap pertanyaan.

## Contoh Masukan

```
0.....8
ABB
2 4
1 7
2 4
6 7
1 1
```

## Contoh Keluaran



7  
6  
4  
2

## Penjelasan Contoh

Contoh kasus uji ini sesuai dengan ilustrasi pada deskripsi soal.

## Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq |S| \leq 100.000$
- S hanya mengandung karakter 'A' atau 'B'.
- $1 \leq K \leq 10^9$
- $1 \leq Q \leq 100.000$
- $1 \leq L[i] \leq R[i] \leq N$

### Subsoal 1 (6 poin):

Hanya berisi kasus uji berikut:

```
.1.....8  
AABA  
3 6  
1 4  
5 9  
10 13  
1 2  
3 5  
10 12
```

### Subsoal 2 (8 poin):

Hanya berisi kasus uji berikut:

```
..2.....8  
AABABAABB  
11 7  
1 100  
1 1  
2 2  
3 3  
98 98  
99 99  
100 100
```

### Subsoal 3 (11 poin):

- $|S| \leq 1.000$
- $K = 1$

**Subsoal 4 (9 poin):**

- $K = 1$
- $Q = 1$

**Subsoal 5 (16 poin):**

- $Q = 1$

**Subsoal 6 (15 poin):**

- $|S| \leq 2$

**Subsoal 7 (19 poin):**

- Untuk setiap  $i$ ,  $L[i] = R[i]$

**Subsoal 8 (16 poin):**

- Tidak ada batasan tambahan.

# Permainan Pajel

## Deskripsi

Pak Dengklek membuat permainan soliter baru untuk Anda bernama Pajel. Pada permainan ini, terdapat sebuah papan berisi petak-petak berukuran  $N$  baris dan  $N$  kolom. Baris-baris dan kolom-kolom dinomori dari 1 sampai dengan  $N$ .

Mulanya, setiap petak berwarna putih. Anda diminta untuk mewarnai nol atau lebih petak dengan warna merah (M) atau biru (B), yang memenuhi seluruh petunjuk pada pinggir luar keempat sisi papan.

Petak-petak pada pinggir luar sisi atas papan berisi petunjuk berupa salah satu dari:

- Sebuah bilangan bulat positif diikuti dengan karakter 'M' atau 'B': bilangan menyatakan jarak petak berwarna terdekat pada kolom tersebut dari sisi atas, dan karakter menyatakan warnanya.
- Karakter '0': menyatakan bahwa tidak ada petak berwarna apapun pada kolom tersebut.
- Karakter '-': tidak menyatakan petunjuk apapun sehingga dapat diabaikan.

Petak-petak pada pinggir luar sisi kiri, kanan, dan bawah didefinisikan dengan cara yang mirip (untuk sisi kiri dan kanan, "kolom" diganti dengan "baris" pada aturan di atas).

Dua buah petak berwarna sama (selain putih) yang berbagi sisi yang sama dikatakan berada pada satu wilayah yang sama.

Sebagai contoh, berikut adalah dua buah solusi berbeda dari permainan Pajel yang sama. Pada kedua solusi berikut, terdapat 3 buah wilayah.

|           |           |   |           |   |   |           |
|-----------|-----------|---|-----------|---|---|-----------|
|           | <b>1M</b> | - | -         | - | - |           |
| -         |           |   |           |   |   | -         |
| <b>1B</b> |           |   |           |   |   | <b>3B</b> |
| <b>0</b>  |           |   |           |   |   | -         |
| -         |           |   |           |   |   | -         |
| <b>5M</b> |           |   |           |   |   | <b>1M</b> |
|           | <b>4B</b> | - | <b>2M</b> | - | - |           |
|           | <b>1M</b> | - | -         | - | - |           |
| -         |           |   |           |   |   | -         |
| <b>1B</b> |           |   |           |   |   | <b>3B</b> |
| <b>0</b>  |           |   |           |   |   | -         |
| -         |           |   |           |   |   | -         |
| <b>5M</b> |           |   |           |   |   | <b>1M</b> |
|           | <b>4B</b> | - | <b>2M</b> | - | - |           |

|  |           |   |           |   |   |  |
|--|-----------|---|-----------|---|---|--|
|  | <b>4B</b> | - | <b>2M</b> | - | - |  |
|--|-----------|---|-----------|---|---|--|

Berikut ini adalah solusi lainnya, yang mengandung 6 wilayah.

|           |           |   |           |   |   |           |
|-----------|-----------|---|-----------|---|---|-----------|
|           | <b>1M</b> | - | -         | - | - |           |
| -         |           |   |           |   |   | -         |
| <b>1B</b> |           |   |           |   |   | <b>3B</b> |
| <b>0</b>  |           |   |           |   |   | -         |
| -         |           |   |           |   |   | -         |
| <b>5M</b> |           |   |           |   |   | <b>1M</b> |
|           | <b>4B</b> | - | <b>2M</b> | - | - |           |

Pada permainan ini, tujuan Anda adalah untuk mewarnai petak-petak sesuai dengan petunjuk-petunjuk yang diberikan, sedemikian sehingga banyaknya wilayah berbeda sesedikit mungkin. Misalkan banyaknya wilayah berbeda pada solusi juri adalah P, dan pada solusi Anda adalah Q. Nilai Anda akan bergantung pada seberapa dekat Q ke P, menggunakan rumus yang akan dijelaskan pada bagian Penilaian.

### Informasi Tipe Soal

Soal ini adalah soal "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 1 + 10 masukan untuk diselesaikan: osn-2018-pajel\_0.in, osn-2018-pajel\_1.in, osn-2018-pajel\_2.in, ..., osn-2018-pajel\_10.in. Masukan contoh (nomor 0) tidak termasuk dalam penilaian peserta.

Untuk setiap berkas masukan yang diselesaikan (Anda tidak harus menyelesaikan semua masukan), buatlah berkas keluaran dengan nama osn-2018-pajel\_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:

```
N P
U[1] U[2] ... U[N]
L[1] R[1]
L[2] R[2]
.
.
L[N] R[N]
D[1] D[2] .. D[N]
```

dengan:

- $U[i]$  menyatakan petunjuk di atas kolom ke- $i$ .
- $L[i]$  menyatakan petunjuk di kiri baris ke- $i$ .
- $R[i]$  menyatakan petunjuk di kanan baris ke- $i$ .
- $D[i]$  menyatakan petunjuk di bawah kolom ke- $i$ .

## Format Keluaran

$N$  baris berisi  $N$  karakter, yang menunjukkan warna setiap petak: 'M' untuk merah, 'B' untuk biru, '-' untuk putih.

## Contoh Masukan

```
5 3
1M - - - -
- -
1B 3B
0 -
- -
5M 1M
4B - 2M - -
```

## Contoh Keluaran

```
MM---
BBB--
-----
-MMMM
----M
```

## Penjelasan Contoh

Contoh keluaran tersebut menyatakan solusi dari gambar pertama di atas.

## Penilaian

Untuk setiap kasus uji:

| Kondisi                | Poin  |
|------------------------|---|
| $Q < P$                | 11  |
| $Q = P$                | 10  |
| $1 \leq Q - P \leq 40$ | $\left\lfloor 10 - \sqrt{2 \times (Q - P)} \right\rfloor$ |
| $Q - P > 40$           | 0   |

### Kasus-Kasus Uji

Untuk setiap kasus uji:

- $5 \leq N \leq 500$
- $1 \leq P \leq N^2$
- Dijamin terdapat setidaknya satu solusi yang memenuhi seluruh petunjuk.

#### osn-2018-pajel\_1.in

- $N = 10$
- $P = 4$

#### osn-2018-pajel\_2.in

- $N = 10$
- $P = 9$

#### osn-2018-pajel\_3.in

- $N = 30$
- $P = 9$
- Pandang seluruh petunjuk pada sisi-sisi papan sebagai mata-mata rantai yang tersambung mengelilingi papan. Di antara dua petunjuk bukan '-', selalu terdapat setidaknya 4 petunjuk '-'.

#### osn-2018-pajel\_4.in

- $N = 60$
- $P = 139$
- Tidak ada petunjuk '-'.

#### osn-2018-pajel\_5.in

- $N = 100$
- $P = 137$
- Tidak ada petunjuk berwarna 'B'.

#### osn-2018-pajel\_6.in

- $N = 150$

- $P = 152$
- Bilangan pada setiap petunjuk (jika ada), tidak lebih dari 9.

#### **osn-2018-pajel\_7.in**

- $N = 250$
- $P = 181$
- Seluruh petunjuk pada sisi kiri dan sisi kanan adalah '-'.

#### **osn-2018-pajel\_8.in**

- $N = 357$
- $P = 156$
- Petunjuk bukan '-' hanya terdapat pada baris dan kolom yang indeksnya sama dengan 3 dalam modulo 4.
- Bilangan pada setiap petunjuk (jika ada), sama dengan 3 dalam modulo 4.

#### **osn-2018-pajel\_9.in**

- $N = 499$
- $P = 427$
- Pandang seluruh petunjuk pada sisi-sisi papan sebagai mata-mata rantai yang tersambung mengelilingi papan. Di antara dua petunjuk bukan '-', selalu terdapat setidaknya 2 petunjuk '-'.

#### **osn-2018-pajel\_10.in**

- $N = 500$
- $P = 978$



# Festival FPB

## Deskripsi

Di sebuah desa kecil di Padang, terdapat  $N$  rumah yang berjajar bersebelahan membentuk sebuah garis lurus. Rumah ke- $i$  memiliki  $A[i]$  anggota keluarga.

Untuk menambah pendapatan daerah, sang kepala desa mencanangkan program "Festival FPB". Setiap harinya, kepala desa akan memilih satu atau lebih rumah, lalu membaginya ke dalam tepat  $K$  buah kelompok sedemikian sehingga:

- Setiap rumah termasuk ke dalam paling banyak satu kelompok.
- Semua rumah dalam setiap kelompok yang sama terletak saling bersebelahan, tidak diselingi oleh rumah lain yang bukan merupakan kelompok tersebut.

Kemudian, setiap kelompok memberikan sumbangan sebesar faktor persekutuan terbesar (FPB) dari banyaknya anggota keluarga di rumah-rumah pada kelompok tersebut.

Kepala desa akan selalu memilih pengelompokan yang berbeda setiap harinya. Festival ini berakhir ketika semua kemungkinan pemilihan  $K$  buah kelompok sudah terjadi.

Kepala desa penasaran, pada akhirnya, berapakah total pendapatan dari festival ini? Karena totalnya bisa sangat besar, ia hanya tertarik pada hasilnya modulo  $1.000.000.007$ .

## Format Masukan

Baris pertama akan berisi "label kasus uji". Label kasus uji adalah sebuah string yang dijelaskan sebagai berikut:

- Panjang string tersebut adalah banyaknya subsoal ditambah satu.
- Karakter ke- $0$  (indeks dimulai dari  $0$ ) akan berisi  $0$  jika kasus uji tersebut merupakan contoh kasus uji, atau berisi  $'.'$  (titik) jika bukan.
- Untuk setiap nilai  $i$  di antara  $1$  hingga banyaknya subsoal, berlaku:
  - jika kasus uji tersebut memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi  $i$ , atau
  - jika kasus uji tersebut tidak memenuhi batasan subsoal ke- $i$ , maka karakter ke- $i$  berisi karakter  $'.'$  (titik).

Sebagai contoh, apabila label sebuah kasus uji sebuah soal adalah  $0. .345$ , maka:

- Soal tersebut memiliki  $5$  buah subsoal,
- Kasus uji tersebut merupakan contoh kasus uji, dan
- Kasus uji tersebut memenuhi batasan subsoal ke- $3$ , ke- $4$ , dan ke- $5$ .

Baris-baris berikutnya diberikan dalam format berikut:

|                                    |
|------------------------------------|
| $N$ $K$<br>$A[1]$ $A[2]$ .. $A[N]$ |
|------------------------------------|

## Format Keluaran

Sebuah baris berisi total pendapatan, modulo  $1.000.000.007$ .

## Contoh Masukan

```
0..3.56..9
3 2
6 4 5
```

## Contoh Keluaran

```
44
```

## Penjelasan Contoh

Berikut adalah semua cara pemilihan K kelompok berikut pendapatannya:

- [6] [4] 5:  $\text{FPB}(6) + \text{FPB}(4) = 6 + 4 = 10$ .
- [6] 4 [5]:  $\text{FPB}(6) + \text{FPB}(5) = 6 + 5 = 11$ .
- [6] [4 5]:  $\text{FPB}(6) + \text{FPB}(4, 5) = 6 + 1 = 7$ .
- [6 4] [5]:  $\text{FPB}(6, 4) + \text{FPB}(5) = 2 + 5 = 7$ .
- 6 [4] [5]:  $\text{FPB}(4) + \text{FPB}(5) = 4 + 5 = 9$ .

Maka, total pendapatan adalah  $10 + 11 + 7 + 7 + 9 = 44$ .

## Subsoal

Untuk semua subsoal, berlaku:

- $1 \leq N \leq 50.000$
- $1 \leq K \leq \min(N, 20)$
- $1 \leq A[i] \leq 100.000$

### Subsoal 1 (7 poin):

Hanya berisi kasus uji berikut:

```
.1.34567.9
9 1
30 15 3 48 24 8 12 100 52
```

### Subsoal 2 (8 poin):

Hanya berisi kasus uji berikut:

```
..23..6.89
9 3
1 1 1 1 1 1 1 1 1
```

### Subsoal 3 (7 poin):

- $N \leq 10$

### Subsoal 4 (8 poin):

- $N \leq 500$

- $K = 1$

**Subsoal 5 (10 poin):**

- $N \leq 500$
- $K \leq \min(N, 2)$

**Subsoal 6 (15 poin):**

- $N \leq 500$

**Subsoal 7 (19 poin):**

- $K = 1$

**Subsoal 8 (11 poin):**

- Semua nilai  $A[i]$  sama.

**Subsoal 9 (15 poin):**

- Tidak ada batasan tambahan.