

Hak Cipta Dilindungi  
Undang-undang



## **OLIMPIADE SAINS NASIONAL 2015**

### **DESKRIPSI SOAL**



### **INFORMATIKA/KOMPUTER**

#### **SESI – 0**

Waktu: 2 Jam

**Daftar Soal:** A. Tekotek  
B. Sengketa Tanah  
C. Menimbang

# Tekotek

Time limit: 1000 ms

Memory limit: 32768 KB

## Deskripsi

*Tek kotek, kotek kotek  
Anak ayam turun berkotek  
Tek kotek, kotek kotek*

*Anak ayam turun berkotek  
Anak ayam turunlah 4  
Mati satu tinggallah 3  
Anak ayam turunlah 3  
Mati satu tinggallah 2  
Anak ayam turunlah 2  
Mati satu tinggallah 1  
Anak ayam turunlah 1  
Mati satu tinggallah induknya*

Pak Dengklek sedang berjalan-jalan di Malioboro sambil menyanyikan lagu tersebut. Akan tetapi lagu tersebut sudah selesai dinyanyikan sebelum Pak Dengklek menyelesaikan perjalanannya. Sehingga, Pak Dengklek berencana menambah jumlah ayam pada lirik lagu tersebut. Karena Pak Dengklek sedang sibuk memilih souvenir, Pak Dengklek meminta bantuan anda untuk membuatkan lirik untuk Pak Dengklek.

## 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 kedua berisi satu bilangan  $N$  yaitu jumlah ayam yang diinginkan Pak Dengklek.

### Format Keluaran

2N baris lirik yang diinginkan Pak Dengklek.

### Contoh Masukan

```
0..34
3
```

### Contoh Keluaran

```
Anak ayam turunlah 3
Mati satu tinggallah 2
Anak ayam turunlah 2
Mati satu tinggallah 1
Anak ayam turunlah 1
Mati satu tinggallah induknya
```

### Subsoal

#### Subsoal 1 (10 poin)

Hanya berisi kasus uji ini:

```
.1.34
5
```

#### Subsoal 2 (20 poin)

Hanya berisi kasus uji ini:

```
..234
10
```

#### Subsoal 3 (30 poin)

- $1 \leq N \leq 100$

#### Subsoal 4 (40 poin)

- $1 \leq N \leq 1.000$

# Sengketa Tanah

Time limit: 1000 ms

Memory limit: 65536 KB

## Deskripsi

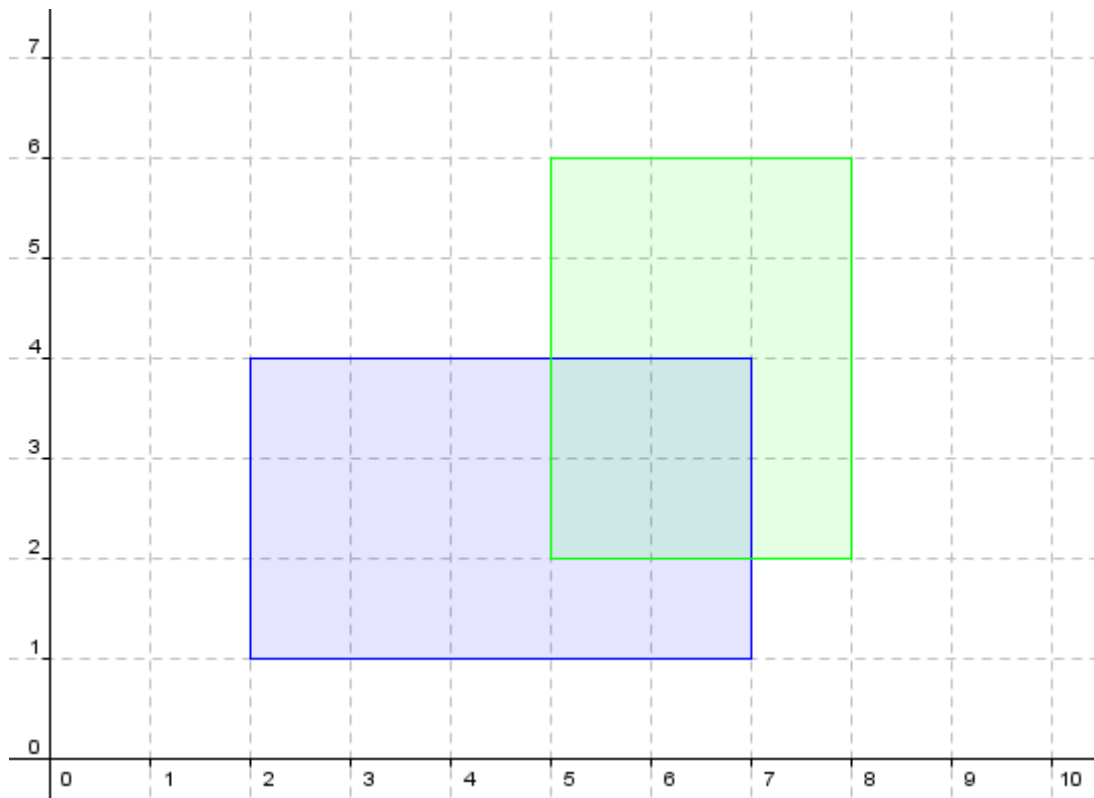
Candi Sewu adalah candi yang terletak tepat di perbatasan daerah Jawa Tengah dan Daerah Istimewa Yogyakarta. Mungkin kalian pernah mendengar legenda bahwa candi ini dibangun dalam semalam. Bandung Bandawasa ingin meminang putri Roro Jonggrang, namun sang putri memberikan satu persyaratan: membangun 1.000 candi dalam semalam. Singkatnya, dia digagalkan oleh Roro Jonggrang ketika sudah mencapai 999 candi. Akibatnya dia marah, dan mengutuk Roro Jonggrang untuk menjadi candi terakhir.

Begitulah kisah yang selama ini diceritakan turun-temurun. Namun, ada satu bagian yang terlewat dari kisah itu. Ketika ingin memulai membangun candi, Bandung Bandawasa membeli sepetak tanah berukuran persegi panjang, tempat akan dibangunnya candi tersebut. Namun, Mpu Dengklek (leluhur pak Dengklek) marah karena menurutnya tanah yang ingin dibeli Bandung Bandawasa beririsan dengan tanah miliknya. Tentu saja Bandung Bandawasa harus menyelesaikan sengketa tanah ini secepatnya sebelum memulai membangun candi.

Kedua tanah berbentuk persegi panjang, di mana sisi-sisinya sejajar dengan sumbu  $x$  dan  $y$  pada koordinat kartesian. Kedua tanah didefinisikan dengan koordinat kiri bawah dan kanan atas persegi panjang. Tanah Bandung Bandawasa memiliki koordinat kiri bawah  $(Xb1, Yb1)$  dan koordinat kanan atas  $(Xb2, Yb2)$ . Sementara itu, tanah Mpu Dengklek memiliki koordinat kiri bawah  $(Xd1, Yd1)$  dan koordinat kanan atas  $(Xd2, Yd2)$ .

Anda akan membantu mereka menyelesaikan sengketa, dengan menentukan apakah benar bahwa kedua tanah beririsan. Maksud dari beririsan adalah terdapat area dengan luas lebih besar dari nol yang masuk ke dalam wilayah kedua tanah.

Sebagai contoh, misalkan tanah Bandung Bandawasa mempunyai koordinat  $(2, 1)$  hingga  $(7, 4)$ , dan tanah Mpu Dengklek mempunyai koordinat  $(5, 2)$  hingga  $(8, 6)$ . Pada gambar berikut, terlihat bahwa kedua tanah beririsan di koordinat  $(5, 2)$  hingga  $(7, 4)$ . Tanah Bandung Bandawasa adalah yang berwarna biru, dan tanah Mpu Dengklek adalah yang berwarna hijau.



### 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 berikutnya berisi 4 buah bilangan bulat  $Xb1$ ,  $Yb1$ ,  $Xb2$ , dan  $Yb2$ , yang mendeskripsikan tanah Bandung Bandawasa.

Baris berikutnya berisi 4 buah bilangan bulat  $Xd1$ ,  $Yd1$ ,  $Xd2$ , dan  $Yd2$ , yang mendeskripsikan tanah Mpu Dengklek.

### Format Keluaran

Keluaran terdiri dari sebuah baris yang berisi sebuah string. Jika kedua tanah beririsan, maka keluarkan **SENGKETA**. Jika tidak, keluarkan **DAMAI**.

### Contoh Masukan 1

```
0...4
2 1 7 4
5 2 8 6
```

### Contoh Keluaran 1

SENGKETA

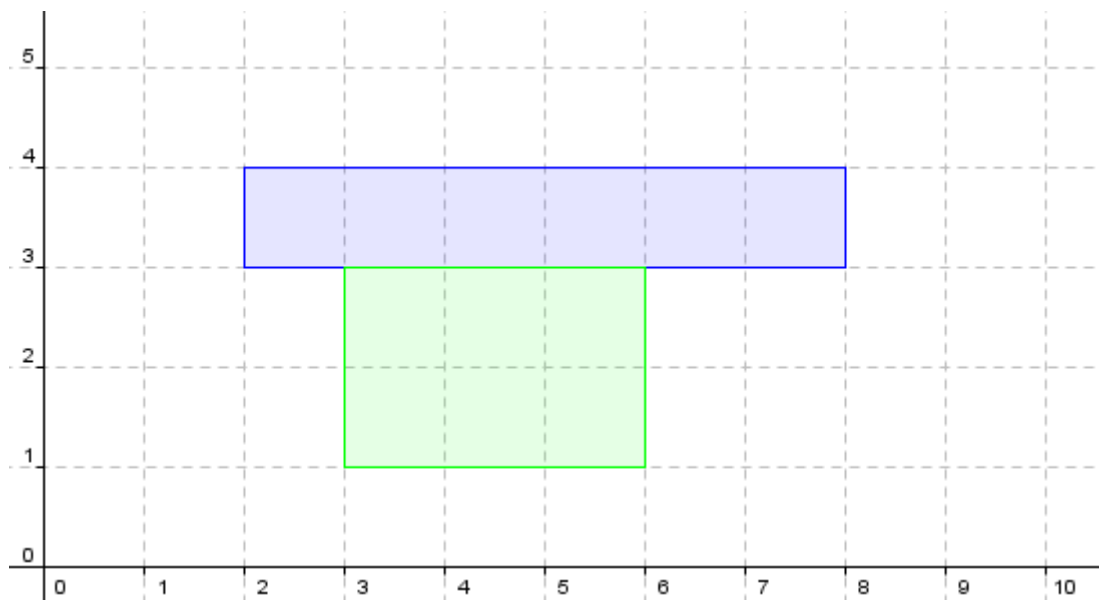
### Contoh Masukan 2

```
0...4
2 3 8 4
3 1 6 3
```

### Contoh Keluaran 2

DAMAI

### Penjelasan Contoh 2



Walaupun kedua tanah saling menempel, tapi tidak ada area yang masuk ke dalam wilayah kedua tanah, sehingga tidak terjadi sengketa.

### Subsoal

Untuk semua subsoal, berlaku:

- $X_{b_1} < X_{b_2}$

- $Y_{b1} < Y_{b2}$
- $X_{d1} < X_{d2}$
- $Y_{d1} < Y_{d2}$
- Semua bilangan pada koordinat adalah bilangan bulat positif dan kurang dari 1.000.

### Subsoal 1 (10 poin)

Hanya berisi kasus uji ini:

```
.1 . .4
0 0 100 120
50 80 75 100
```

### Subsoal 2 (10 poin)

Hanya berisi kasus uji ini:

```
. .234
1 2 3 4
1 6 8 9
```

### Subsoal 3 (30 poin)

- $X_{b1} = X_{d1}$

### Subsoal 4 (50 poin)

- Tidak ada batasan tambahan.

# Menimbang

Time limit: 1000 ms

Memory limit: 32768 KB

## Deskripsi

Pak Dengklek baru saja kembali dari wisata ke Yogyakarta. Selama berada di Yogyakarta, Pak Dengklek sering sekali belanja. Ia berhasil membeli banyak oleh-oleh untuk bebek-bebeknya. Setelah puas, Pak Dengklek pulang ke rumah dengan menaiki bus AC. Ia memutuskan untuk tidur selama perjalanan pulang karena terlalu lelah.

Ketika Pak Dengklek bangun dan bersiap untuk turun, Pak Dengklek menyadari sesuatu yang aneh. Dia tidak bisa membedakan tasnya dengan tas milik penumpang lain! Ternyata, model tas yang digunakan Pak Dengklek sangat populer. Dari luar, semua tas terlihat sama saja. Satu-satunya yang Pak Dengklek tahu, tas miliknya adalah satu-satunya tas yang lebih berat dari tas-tas lain. Semua tas lain beratnya sama.

Pak Dengklek akhirnya menaruh seluruh tas secara berjejer dan menomori tas-tas tersebut dengan angka 1..N dari kiri ke kanan. Pak Dengklek memiliki kemampuan yang menarik: ia bisa mengangkat sejumlah tas berbeda di masing-masing tangannya dan mengetahui kumpulan tas yang mana yang lebih berat. Dengan melakukan hal ini, dia yakin bisa menemukan tasnya. Namun, sebentar lagi bus akan berjalan lagi dan Pak Dengklek harus cepat turun sehingga ia hanya bisa melakukan K kali penimbangan. Bantulah Pak Dengklek memilih tas-tas mana yang harus ditimbang agar bisa menemukan tasnya secepat mungkin!

## Format Interaksi

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 kedua berisi bilangan bulat N dan K, yang merupakan banyak tas yang berada di bus dan banyak penimbangan maksimum yang bisa Pak Dengklek lakukan. Setelah itu, program Anda dapat melakukan serangkaian tindakan, yang masing-masing merupakan salah satu dari:



- Memilih tas. Program Anda harus mengeluarkan sebuah baris berisi **AMBIL Y** yakni, kata AMBIL diikuti dengan nomor tas pilihan Anda.
- Bertanya. Program Anda harus mengeluarkan sebuah baris berisi **TIMBANG A T1 T2 .. TAB T1 T2 .. TB** yakni, kata TIMBANG diikuti dengan banyak tas yang ingin anda pegang di tangan kiri dan nomor-nomor tas yang ingin anda pegang di tangan kanan.

Setiap kali program Anda selesai mengeluarkan pertanyaan, program Anda membaca sebuah string, **KIRI** jika lebih berat tas-tas di kiri, **KANAN** jika lebih berat tas-tas di kanan, **SAMA** jika berat kedua sisi sama.

Setelah anda memilih tas, program akan berhenti dan anda akan mendapat poin jika tebakan anda benar.

Pastikan program Anda berhenti melakukan interaksi setelah menebak tas mana yang paling berat.

### Contoh Interaksi

Berikut adalah contoh interaksi program, dengan tas terberatnya adalah tas nomor 8.

Keluaran Program Anda	Keluaran Program Grader
	01...
	8 7
TIMBANG 1 1 1 2	
	SAMA
TIMBANG 2 1 2 1 4	
	KIRI
TIMBANG 2 1 8 1 2	
	KIRI
AMBIL 8	
	(interaksi selesai)

### Subsoal

Pada semua subsoal, harus berlaku:

- Jika program Anda memilih tas, maka  $1 \leq Y \leq N$
- Jika program Anda bertanya, maka  $1 \leq A, B$ , nomor-nomor tas  $\leq N$

#### Subsoal 1 (10 poin)

- $N = 8$
- $K = 7$

#### Subsoal 2 (20 poin)

- $N = 16$
- $K = 8$

#### Subsoal 3 (30 poin)

- $N = 32$
- $K = 6$

#### Subsoal 4 (40 poin)

- $N = 81$
- $K = 4$

### Peringatan

Jika program Anda melakukan salah satu dari hal-hal di bawah ini:

- melakukan tindakan di luar format dan batasan yang ditentukan,
- bertanya lebih dari  $K$  kali, atau
- salah mengambil tas (bukan yang terberat)

maka nilai Anda untuk subsoal yang bersangkutan adalah nol.

Selain itu, pastikan pula bahwa program Anda harus berhenti jika selesai melakukan interaksi. Bila tidak, maka Anda mungkin mendapatkan *Time Limit Exceeded* atau *Runtime Error* untuk kasus uji yang bersangkutan.

Hak Cipta Dilindungi  
Undang-undang



## **OLIMPIADE SAINS NASIONAL 2015**

### **DESKRIPSI SOAL**



### **INFORMATIKA/KOMPUTER**

#### **SESI – 1**

Waktu: 5 Jam

**Daftar Soal:** A. Pertahanan Yogya  
B. Menyiram Sawah  
C. Bisa Jadi Teebak Angka

# Pertahanan Yogya

Time limit: 1000 ms

Memory limit: 65536 KB

## Deskripsi

Yogya dalam bahaya! Sesosok makhluk jahat ingin menguasai Yogya dengan kekuatan kegelapan. Pak Dengklek tidak ingin tinggal diam walaupun ia sedang berlibur. Ia berusaha untuk mengalahkan dan mengusir makhluk jahat tersebut dari Yogya. Tentu tugas ini tidaklah mudah, karena selain makhluk jahat tersebut merupakan petarung yang kuat, ia juga memiliki banyak anak buah.

Sebagai seorang petarung senior, Pak Dengklek tahu bahwa ia tidak boleh gegabah saat melawan makhluk jahat tersebut. Berkat kemampuan analisisnya, ia punya informasi keterampilan bertarung makhluk jahat dan seluruh anak buahnya. Untuk kemudahan, makhluk-makhluk dinomori dari 1 sampai dengan  $N$ . Keterampilan bertarung makhluk ke- $i$  dinyatakan dalam dua pasang bilangan  $S_i$  dan  $T_i$ , dengan  $S_i$  adalah stamina dan  $T_i$  adalah tenaga makhluk tersebut. Untuk memudahkan, makhluk ke-1 adalah sang makhluk jahat, dan makhluk ke- $i$  untuk  $i > 1$  adalah prajurit-prajuritnya. Pak Dengklek juga memiliki keterampilan bertarung sendiri yang dinyatakan oleh  $S_d$  dan  $T_d$ .

Proses pertarungan digambarkan oleh alur berikut: Pak Dengklek menyerang lawan, mengurangi stamina musuh sebanyak  $T_d$ . Kemudian musuh balik menyerang pak Dengklek, mengurangi stamina pak Dengklek sebanyak  $T_i$ . Proses tersebut berulang terus-menerus hingga salah satu di antara mereka kehabisan stamina ( $\text{stamina} \leq 0$ ) dan dinyatakan kalah.

Pak Dengklek sadar, mungkin sang makhluk jahat terlalu kuat untuk dikalahkan, karena itu ia bisa saja melawan prajurit-prajuritnya terlebih dahulu. Setiap kali Pak Dengklek mengalahkan musuh, kemampuannya bertambah sebanyak  $L_s$  dan  $L_t$ . Artinya, stamina pak Dengklek bertambah sebanyak  $L_s$  dan tenaganya bertambah  $L_t$  dari nilai yang sebelumnya. Sebagai tambahan, setiap kali Pak Dengklek selesai dari suatu pertempuran, ia menyempatkan diri beristirahat sehingga staminanya kembali pulih seperti semula.

Meskipun Pak Dengklek semakin kuat jika semakin banyak prajurit yang dikalahkan, ia juga tak ingin berlama-lama. Bantulah Pak Dengklek mengatur urutan prajurit-prajurit yang perlu dikalahkan sampai ia mampu mengalahkan sang makhluk jahat, sedemikian sehingga banyaknya prajurit yang perlu dikalahkan adalah minimum.

## 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 berikutnya terdiri dari 5 bilangan:  $S_d$ ,  $T_d$ ,  $L_s$ ,  $L_t$ , dan  $N$ .

$N$  baris berikutnya terdiri dari dua buah bilangan  $S_i$  dan  $T_i$  yang menyatakan keahlian bertarung musuh ke-i. Musuh ke-1 merupakan makhluk jahat yang perlu dikalahkan oleh Pak Dengklek.

### Format Keluaran

Keluarkan sebuah bilangan  $Q$  yang menyatakan banyaknya prajurit minimum yang perlu dikalahkan pak Dengklek sebelum siap melawan makhluk jahat.  $Q$  baris berikutnya adalah indeks  $i$  ( $1 < i \leq N$ ) yang menyatakan urutan prajurit musuh yang dikalahkan. Apabila terdapat lebih dari satu kemungkinan urutan, keluarkan yang mana saja.

Apabila Pak Dengklek tidak bisa membunuh makhluk jahat, keluarkan -1.

### Contoh Masukan

```
0..34567
10 5 2 0 4
15 6
11 5
6 4
14 5
```

### Contoh Keluaran

```
2
3
2
```

### Subsoal

Pada setiap subsoal, berlaku

- $1 \leq S_d, T_d, S_i, T_i \leq 100.000$

- $0 \leq L_s, L_t \leq 100.000$

#### Subsoal 1 (5 poin)

Subsoal ini hanya berisi kasus uji berikut ini:

```
. 1 . 3 4 . . 7
5 2 1 1 4
6 4
2 10
20 1
5 5
```

#### Subsoal 2 (7 poin)

Subsoal ini hanya berisi kasus uji berikut ini:

```
. . 2 3 4 . . 7
25 25 25 25 5
93 52
90 60
32 92
14 78
19 77
```

#### Subsoal 3 (10 poin)

- $1 \leq N \leq 10$

#### Subsoal 4 (12 poin)

- $1 \leq N \leq 100$

#### Subsoal 5 (15 poin)

- $1 \leq N \leq 1.000$
- $L_t = 0$

#### Subsoal 6 (20 poin)

- $1 \leq N \leq 100.000$
- $L_t = 0$

#### Subsoal 7 (31 poin)

- $1 \leq N \leq 100.000$

## Menyiram Sawah

Time limit: 1000 ms

Memory limit: 65536 KB

### Deskripsi

Pak Dengklek memiliki sawah yang sangat luas. Uniknya, sawah Pak Dengklek berada di pegunungan, sehingga ketinggian sawah Pak Dengklek bervariasi antar petaknya. Sawah Pak Dengklek berukuran  $N \times M$  petak. Petak sawah pojok atas kiri dinomori petak  $[1,1]$  dan pojok kanan bawah dinomori  $[N,M]$ . Setiap petak sawah Pak Dengklek memiliki ketinggian  $T[i,j]$ .

Agar padi-padi di sawah Pak Dengklek tumbuh dengan subur, ia harus menyiram sawahnya setiap hari. Karena struktur ketinggian sawah Pak Dengklek yang sedemikian rupa, ketika Pak Dengklek menyiram sawah pada suatu petak, air akan selalu mengalir ke petak dengan ketinggian yang lebih kecil. Lebih jelasnya, ketika Pak Dengklek menyiram petak  $[i,j]$ , air akan mengalir ke seluruh petak yang bertetangga (4 arah) dengan  $[i,j]$  jika ketinggian petak tetangganya lebih rendah dibandingkan ketinggian petak  $[i,j]$ . Tetangga dari suatu petak adalah 4 petak lainnya yang berbagi sisi. Air ini akan terus mengalir, hingga ia berada di petak dimana semua tetangganya memiliki ketinggian yang lebih tinggi atau sama.

Sebagai ilustrasi, perhatikan sawah Pak Dengklek berikut ini:

1	2	4	12	44
50	200	8	9	40
55	3	100	10	7
35	1	1	5	80

Angka-angka tersebut menunjukkan ketinggian dari petak yang bersangkutan. Jika Pak Dengklek menyiram Petak  $[3,4]$ , maka hasilnya adalah:

1	2	4	12	44
50	200	8	9	40
55	3	100	10	7
35	1	1	5	80

Petak berwarna biru adalah petak yang dibasahi baik disiram secara langsung maupun karena aliran air melewati petak tersebut.

Karena Pak Dengklek menyukai tantangan, Ia memberi anda teka-teki. Anda diberikan sub-persegi dari sawah Pak Dengklek yang ditandai dengan 2 petak ujung  $[a,b]$  dan  $[c,d]$ . Petak  $[a,b]$  menandakan petak ujung kiri atas, sementara  $[c,d]$  menandakan petak ujung kanan bawah. Anda harus menjawab pertanyaan, apakah mungkin seluruh petak yang berada di sub-persegi  $[a,b]$  sampai  $[c,d]$  dapat dibasahi hanya dengan sekali siram di petak kiri atas (sawah di petak  $[a,b]$ ) atau kanan bawah (sawah di petak  $[c,d]$ ) sub-persegi tersebut? Air TIDAK bisa mengalir keluar dari sub-persegi tersebut. Jika ada petak yang bisa terbasahi namun berada di luar sub-persegi, petak tersebut tidak akan terbasahi. Sebagai contoh, jika sawahnya seperti contoh sebelumnya dan sub-perseginya adalah  $[2,2]..[3,4]$  maka kemungkinan sawah setelah terbasahi adalah:

1	2	4	12	44
50	200	8	9	40
55	3	100	10	7
35	1	1	5	80

jika yang dibasahi awalnya adalah petak  $[2,2]$ , dan

1	2	4	12	44
50	200	8	9	40
55	3	100	10	7
35	1	1	5	80

jika yang dibasahi awalnya adalah petak  $[3,4]$

### 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 kedua terdiri dari 3 bilangan N,M dan Q.
- N baris berikutnya terdiri dari M bilangan yang mendeskripsikan nilai ketinggian petak-petak sawah Pak Dengklek.
- Q baris berikutnya adalah teka-teki Pak Dengklek yang terdiri dari 4 bilangan a, b, c, dan d.

### Format Keluaran

Untuk setiap teka-teki, keluarkan 'Kiri Atas' jika seluruh petak pada sub-persegi bisa dibasahi dengan menyiram petak [a,b]. Keluarkan 'Kanan Bawah' jika seluruh petak pada sub-persegi bisa dibasahi dengan menyiram petak [c,d], atau 'Tidak Mungkin' jika kedua hal tersebut tidak memungkinkan.

### Contoh Masukan

```
0...4.6
3 3 4
1 2 3
4 5 4
1 2 1
1 1 2 2
2 2 3 3
1 1 2 3
1 2 3 3
```

### Contoh Keluaran

```
Kanan Bawah
Kiri Atas
Tidak Mungkin
Tidak Mungkin
```

### Subsoal

Untuk semua subsoal, berlaku:

- $0 \leq \text{tinggi masing-masing petak sawah} \leq 10^9$
- $1 \leq a, c \leq N$
- $1 \leq b, d \leq M$

- $1 \leq N, M \leq 1000$
- $b < d$

### Subsoal 1 (7 poin)

Subsoal ini hanya berisi kasus uji berikut ini:

```
.1..4.6
5 5 6
28 58 49 39 63
77 78 25 55 100
76 79 73 72 22
49 78 70 71 58
86 47 26 60 61
1 1 2 2
4 4 5 5
3 4 5 5
2 1 3 2
4 4 5 5
3 3 4 4
```

### Subsoal 2 (9 poin)

Subsoal ini hanya berisi kasus uji berikut ini:

```
..23456
1 50 8
58 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 49 50 51
52 53 54 55 56 70 69 68 67 66 65 64 45 8 91 63 32 15 82 30 86 93 2 64 87
1 13 1 27
1 19 1 37
1 12 1 25
1 30 1 36
1 19 1 21
1 2 1 24
1 30 1 37
1 23 1 30
```

### Subsoal 3 (10 poin)

- $N = 1$
- $1 \leq Q \leq 10$

**Subsoal 4 (10 poin)**

- $1 \leq Q \leq 10$

**Subsoal 5 (19 poin)**

- $N = 1$
- $1 \leq Q \leq 10^6$

**Subsoal 6 (45 poin)**

- $1 \leq Q \leq 10^6$

**Peringatan**

Untuk menangani masukan/keluaran yang banyak, hindari menggunakan cin/cout

# Bisa Jadi Tebak Angka

Time limit: 100 ms

Memory limit: 32768 KB

## Deskripsi

Bebek-bebek Pak Dengklek sedang berwisata di Yogyakarta. Sekarang, mereka sedang berada di Keraton Ngayogyakarta Hadiningrat. Selain berfungsi sebagai tempat tinggal sultan, di sana terdapat museum yang menyimpan berbagai koleksi milik kesultanan, mulai dari pemberian raja-raja Eropa, replika pusaka keraton, hingga gamelan yang dapat didengarkan secara langsung pada saat-saat tertentu. Tentu saja bebek-bebek Pak Dengklek sangat tertarik dengan semua itu.

Semua bebek kecuali Kwak dan Kwek. Ketika bebek-bebek lainnya asyik mengeksplorasi kompleks Keraton yang sangat luas itu, Kwak dan Kwek malah berdiam di bawah pohon karena bosan. Akhirnya mereka memutuskan untuk memainkan sebuah permainan tebak angka sambil menunggu yang lain.

Permainan tersebut berjalan sebagai berikut. Pada awalnya, Kwak memikirkan sebuah bilangan bulat antara 1 sampai dengan  $N$ , inklusif (misalkan bilangan tersebut adalah  $X$ ). Kwek harus menebak nilai dari  $X$ . Untuk itu, Kwek dapat mengajukan serangkaian pertanyaan.

Pada setiap pertanyaan, Kwek memberikan Kwak sebuah himpunan bilangan bulat  $S$ . Kemudian, Kwak harus memberikan salah satu dari ketiga kemungkinan jawaban berikut:

- **YA**, jika  $X$  terdapat pada  $S$ .
- **BISAJADI**, jika  $X$  tidak terdapat pada  $S$ , namun  $X+1$  atau  $X-1$  terdapat pada  $S$ .
- **TIDAK**, jika kedua kondisi di atas tidak terpenuhi.

Pada akhirnya, Kwek harus menebak bilangan yang dipikirkan Kwak. Jika tebakan tersebut sama dengan  $X$ , maka Kwek dinyatakan menang.

Tentu saja mereka tidak dapat bermain terlalu lama karena sebentar lagi rombongan bebek Pak Dengklek akan melanjutkan wisata ke tempat menarik lainnya di Yogyakarta. Oleh karena itu, mereka membatasi permainan ini: Kwek hanya dapat bertanya maksimum  $K$  kali sebelum menebak bilangan yang dipikirkan Kwak. Bantulah Kwek memenangkan permainan ini!

## Format Interaksi

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.

Kemudian program Anda akan menerima dua buah bilangan bulat N dan K.

Setelah itu, program Anda dapat melakukan serangkaian tindakan, yang masing-masing merupakan salah satu dari:

- Menebak. Program Anda harus mengeluarkan sebuah baris berisi **0 Y** yakni, bilangan 0 diikuti dengan tebakan Anda.
- Bertanya. Program Anda harus mengeluarkan sebuah baris berisi **M S1 S2 ... SM** yakni, sebuah bilangan bulat M diikuti dengan M buah bilangan bulat yang menyatakan himpunan yang Anda tanyakan.

Setiap kali program Anda selesai mengeluarkan tebakan, program Anda membaca sebuah string yang dijamin merupakan salah satu dari YA, BISAJADI, dan TIDAK, sesuai dengan definisi pada deskripsi soal.

Pastikan program Anda berhenti melakukan interaksi setelah menebak bilangan yang dipikirkan Kwak.

### Contoh Interaksi

Berikut adalah contoh interaksi program, dengan bilangan yang dipikirkan Kwak adalah 8.

Keluaran Program Anda	Keluaran Program Grader
	0..... 10 3
4 0 3 4 5	
	TIDAK
3 6 7 8	
	YA

Keluaran Program Anda	Keluaran Program Grader
1 7	
	BISAJADI
0 8	
	(interaksi selesai)

### Subsoal

Pada semua subsoal, harus berlaku:

- Jika program Anda bertanya, maka  $1 \leq Y \leq N$
- Jika program Anda menebak, maka  $1 \leq M \leq 10.000$  dan  $0 \leq Si \leq 9.999$

### Subsoal 1 (6 poin)

- $N = 8$
- $K = 3$

### Subsoal 2 (8 poin)

- $N = 64$
- $K = 6$

Khusus untuk subsoal 1 dan subsoal 2:

- Hanya terdapat sebuah kasus uji (satu subsoal dinyatakan oleh satu kasus uji), yang berupa game dan dapat dilihat di halaman pengumuman kontes.
- Anda boleh memainkan permainan ini berulang kali tanpa mendapatkan penalti.
- Jika Anda sudah memenangkan permainan untuk subsoal tertentu, Anda dapat memilih pilihan pada permainan untuk mengeluarkan *source code* yang dapat langsung Anda kirimkan ke *grader* dan menjawab dengan benar pada subsoal yang telah Anda menangkan.
- Anda tidak diwajibkan memainkan permainan ini untuk mengerjakan kedua subsoal ini. Anda diperbolehkan untuk menulis kode Anda sendiri untuk mengerjakan kedua subsoal ini.

### Subsoal 3 (9 poin)

- $N = 9$
- $K = 3$

### Subsoal 4 (12 poin)

- $N = 9$

- $K = 2$

#### Subsoal 5 (23 poin)

- $N = 25$
- $K = 3$

#### Subsoal 6 (12 poin)

- $N = 6.500$
- $K = 15$

#### Subsoal 7 (30 poin)

- $N = 6.500$
- $K = 8$

#### Catatan

Yang perlu diperhatikan adalah bahwa untuk tipe soal interaktif seperti ini, Anda harus selalu memberikan perintah `fflush(stdout);` (bagi pengguna C/C++) atau `flush(output);` (bagi pengguna PASCAL) setiap kali Anda mencetak keluaran (dengan kata lain, setiap kali ada perintah mencetak keluaran misalnya `write`, `writeln`, `printf`, `cout`, atau `puts`, tepat di bawahnya harus ada perintah `fflush/flush`).

Sebagai contoh, berikut adalah contoh *source code* dalam bahasa Pascal yang akan selalu bertanya `1 1` dan kemudian menebak angka 1 tanpa mempedulikan nilai  $N$  dan  $K$  yang diberikan maupun bacaan hasil interaksi.

```
var subsoal, jawaban: string;
    N, K: longint;

begin
    readln(subsoal);
    readln(N, K);

    writeln('1 1');
    flush(output);

    readln(jawaban);

    writeln('0 1');
    flush(output);
end.
```

Dan berikut adalah contoh *source code* yang melakukan hal yang sama dalam bahasa C++.

```
#include <stdio>
#include <cstring>

char subsoal[100], jawaban[100];
int N, K;

int main() {
    gets(subsoal);
    scanf("%d %d", &N, &K);

    printf("1 1\n");
    fflush(stdout);

    gets(jawaban);

    printf("0 1\n");
    fflush(stdout);

    return 0;
}
```



## Peringatan

Jika program Anda melakukan salah satu dari hal-hal di bawah ini:

- melakukan tindakan di luar format dan batasan yang ditentukan,
  - bertanya lebih dari K kali, atau
  - salah menjawab bilangan yang dipikirkan oleh Kwak
- maka nilai Anda untuk subsoal yang bersangkutan adalah nol.

Selain itu, pastikan pula bahwa program Anda harus berhenti jika selesai melakukan interaksi. Bila tidak, maka Anda mungkin mendapatkan *Time Limit Exceeded*, *Runtime Error*, atau *Internal Error* untuk kasus uji yang bersangkutan.

Hak Cipta Dilindungi  
Undang-undang



## **OLIMPIADE SAINS NASIONAL 2015**

### **DESKRIPSI SOAL**



### **INFORMATIKA/KOMPUTER**

#### **SESI – 2**

Waktu: 5 Jam

**Daftar Soal:** A. Belanja di Malioboro  
B. Motif Batik  
C. Ayam Aneh

# Belanja di Malioboro

Time limit: 1000 ms

Memory limit: 65536 KB

## Deskripsi

Bebek-bebek Pak Dengklek sangat senang karena Pak Blangkon mengajak mereka untuk pergi berbelanja di Malioboro, suatu tempat berbelanja yang terkenal di Yogyakarta.

Untuk soal ini, Malioboro bisa dianggap sebagai pertokoan melingkar yang terdiri dari  $M$  toko yang diberi nomor dari 1 sampai dengan  $M$ . Jika ditelusuri secara searah jarum jam mulai dari toko nomor 1, toko di sebelahnya adalah toko nomor 2, disusul dengan toko nomor 3, dan seterusnya hingga toko nomor  $M$ , lalu kembali ke toko nomor 1.

Terdapat  $N$  ekor bebek yang pergi ke Malioboro. Bebek ke- $i$  pada awalnya berada di toko nomor  $P_i$ . Kondisi awal ini bisa dianggap terjadi pada menit ke-0. Bebek-bebek sangat bersemangat dalam berbelanja, dan diketahui mereka bergerak dengan kelajuan konstan, yaitu 1 toko per menit. Diketahui bahwa setiap bebek bergerak searah jarum jam atau berlawanan arah jarum jam, dan tidak pernah berganti arah pergerakan.

Pak Blangkon sewaktu-waktu ingin mengetahui jarak terdekat antar bebek. Ia akan bertanya sebanyak  $K$  kali dengan pertanyaan berbunyi “Pada menit ke- $t$ , berapa jarak terdekat antara bebek-bebek yang ada?” Jarak antara dua bebek didefinisikan sebagai banyaknya toko minimal yang perlu dilewati oleh salah satu bebek untuk pergi ke toko tempat bebek lainnya berada.

Bantulah Pak Blangkon menjawab pertanyaannya dengan efisien!

## 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 berikutnya berisi tiga bilangan bulat, yaitu  $M$ ,  $N$ , dan  $K$ .

N baris berikutnya berisi informasi untuk setiap bebek. Baris ke-i ini berisi dua bilangan, yaitu  $P_i$ ,  $U_i$ . Jika  $U_i$  bernilai -1, artinya bebek ke-i bergerak berlawanan arah jarum jam. Jika  $U_i$  bernilai 1, artinya bebek ke-i bergerak searah jarum jam.

K baris berikutnya berisi sebuah bilangan bulat, yaitu  $t$  yang menyatakan menit ke berapa pada pertanyaan Pak Blangkon. Tidak dijamin bilangan pada baris-baris ini terurut.

### Format Keluaran

Untuk setiap pertanyaan, cetak jarak minimal antara bebek-bebek yang ada pada menit yang ditanyakan.

### Contoh Masukan

```
0..34567
```

```
7 3 4
```

```
2 -1
```

```
3 1
```

```
5 -1
```

```
1
```

```
2
```

```
4
```

```
6
```

### Contoh Keluaran

```
0
```

```
2
```

```
1
```

```
1
```

### Penjelasan Contoh

Berikut ini adalah posisi bebek-bebek dari menit 0 sampai menit 6:

t	Bebek 1	Bebek 2	Bebek 3
0	2	3	5
1	1	4	4
2	7	5	3
3	6	6	2
4	5	7	1
5	4	1	7
6	3	2	6

Untuk jawaban pertanyaannya:

- Pada menit ke-1, jarak terdekat adalah 0 (bebek 2 dengan bebek 3).
- Pada menit ke-2, jarak terdekat adalah 2 (bebek 1 dengan bebek 2, atau bebek 2 dengan bebek 3).
- Pada menit ke-4, jarak terdekat adalah 1 (bebek 2 dengan bebek 3).
- Pada menit ke-6, jarak terdekat adalah 1 (bebek 1 dengan bebek 2).

### Subsoal

Untuk setiap subsoal berlaku

- $U_i = -1$ , atau  $U_i = 1$

#### Subsoal 1 (5 poin)

Subsoal ini hanya berisi kasus uji berikut ini:

```
.1.34567
30 4 10
5 -1
29 -1
12 1
16 1
6
13
17
7
```

20

1

15

16

2

19

### Subsoal 2 (8 poin)

Subsoal ini hanya berisi kasus uji berikut ini:

..234567

70 7 8

37 1

33 1

59 1

67 -1

10 -1

39 -1

48 -1

5

3

4

12

14

6

11

9

### Subsoal 3 (18 poin)

- $2 \leq N \leq 50$
- $1 \leq t \leq 1.000$
- $1 \leq M \leq 1.000$
- $1 \leq K \leq 1.000$

### Subsoal 4 (27 poin)

- $2 \leq N \leq 1.000$
- $1 \leq t \leq 1.000$
- $1 \leq M \leq 1.000$
- $1 \leq K \leq 1.000$

**Subsoal 5 (14 poin)**

- $2 \leq N \leq 1.000$
- $1 \leq t \leq 250.000$
- $1 \leq M \leq 1.000$
- $1 \leq K \leq 250.000$

**Subsoal 6 (19 poin)**

- $2 \leq N \leq 1.000$
- $1 \leq t \leq 250.000$
- $1 \leq M \leq 250.000$
- $1 \leq K \leq 250.000$

**Subsoal 7 (9 poin)**

- $2 \leq N \leq 1.000$
- $1 \leq t \leq 1.000.000.000 (10^9)$
- $1 \leq M \leq 250.000$
- $1 \leq K \leq 250.000$

# Motif Batik

Time limit: 1000 ms

Memory limit: 65536 KB

## Deskripsi

Yogyakarta memiliki beberapa jenis motif batik. Masing-masing jenis motif memiliki makna dan filosofinya tersendiri.

Pak Dengklek yang sedang berada di Yogyakarta mampir ke sejumlah butik dan membeli  $N$  baju batik. Dari batik-batik yang dibeli Pak Dengklek, terdapat  $M$  jenis motif batik yang dinomori dari 1 sampai dengan  $M$ . Batik ke- $i$  yang dibeli Pak Dengklek memiliki motif jenis  $C_i$ , dan memiliki tingkat kecerahan warna berupa suatu bilangan positif  $W_i$ .

Batik-batik yang dibeli ini akan dipakai oleh keluarga besar Dengklek, dan diabadikan dalam sebuah foto keluarga. Pak Dengklek menyadari bahwa keindahan batik terletak pada keanekaragamannya. Dari  $N$  batik yang telah dibeli, Pak Dengklek ingin mengukur “total keindahan” dari seluruh batiknya. Menurut Pak Blangkon, sang ahli batik, total keindahan dari suatu kumpulan batik adalah jumlah dari selisih tingkat kecerahan warna untuk setiap pasang batik yang berbeda motif.

Sebagai contoh, jika  $N = 5$ ,  $C = [1, 2, 1, 2, 2]$ , dan  $W = [5, 3, 2, 4, 6]$ , maka total keindahannya adalah :

$$\begin{aligned} & |W_1 - W_2| + |W_1 - W_4| + |W_1 - W_5| + |W_2 - W_3| + |W_3 - W_4| + |W_3 - W_5| \\ &= |5 - 3| + |5 - 4| + |5 - 6| + |3 - 2| + |2 - 4| + |2 - 6| \\ &= 2 + 1 + 1 + 1 + 2 + 4 \\ &= 11. \end{aligned}$$

Anda diberikan jenis motif dan tingkat kecerahan warna setiap batik yang dibeli Pak Dengklek. Bantulah Pak Dengklek menentukan total keindahan dari batik-batik 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 kedua berisi dua bilangan bulat, yaitu N dan M.

N baris berikutnya berisi dua bilangan bulat. Bilangan-bilangan di baris ke-i ini adalah  $C_i$  dan  $W_i$ .

### Format Keluaran

Sebuah bilangan yang menyatakan total keindahan dari kumpulan batik yang diberikan.

### Contoh Masukan

0 . . 3456 . 8

5 2

1 5

2 3

1 2

2 4

2 6

### Contoh Keluaran

11

### Subsoal

Pada setiap subsoal, berlaku

- $1 \leq C_i \leq M$
- $1 \leq M \leq N$

### Subsoal 1 (6 poin)

Hanya berisi kasus uji ini:

. 1 . 3456 . 8

10 10

10 6

9 10

4 10

2 5

2 7

1 10

3 8

9 4

6 7

6 2

### Subsoal 2 (8 poin)

Hanya berisi kasus uji ini:

..23456.8

11 3

3 7

3 9

3 10

1 9

1 6

1 10

1 9

2 8

1 1

3 7

2 9

### Subsoal 3 (34 poin)

- $1 \leq N \leq 1.000$
- $1 \leq M \leq 50$
- $1 \leq W_i \leq 200$

### Subsoal 4 (6 poin)

- $1 \leq N \leq 10^5$
- $1 \leq M \leq 50$
- $1 \leq W_i \leq 200$

### Subsoal 5 (7 poin)

- $1 \leq N \leq 10^5$
- $1 \leq M \leq 200$
- $1 \leq W_i \leq 200$

**Subsoal 6 (9 poin)**

- $1 \leq N \leq 10^5$
- $1 \leq M \leq 2.000$
- $1 \leq W_i \leq 2.000$

**Subsoal 7 (13 poin)**

- $1 \leq N \leq 10^5$
- $M = N$
- $C_i = i$
- $1 \leq W_i \leq 10^9$

**Subsoal 8 (17 poin)**

- $1 \leq N \leq 10^5$
- $1 \leq W_i \leq 10^9$

**Peringatan**

Bagi pengguna C/C++, gunakan `"%lld"` atau `cin/cout` untuk membaca/menulis bilangan bulat 64-bit.

# Ayam Aneh

Time limit: 100 ms

Memory limit: 32768 KB

## Deskripsi

Untuk menghemat biaya perjalanan, Pak Dengklek dan bebek-bebeknya menginap di rumah Pak Blangkon. Di sana, Pak Dengklek melihat kebun Pak Blangkon yang penuh dengan ayam. Warna ayam Pak Blangkon bermacam-macam. Setiap hari, Pak Blangkon memberikan ayamnya makanan. Makanan ini dapat mengakibatkan DNA seekor ayam bermutasi dan alhasil menjadikan warna ayam tersebut berubah.

Pak Dengklek yang tertarik melihat warna ayam-ayam itu ingin mencoba mengubah warna bebek-bebeknya sendiri. Pak Dengklek pun bertanya kepada Pak Blangkon tentang DNA ayam-ayam yang dimiliki Pak Blangkon agar Pak Dengklek bisa mengimplementasikannya ke bebeknya sendiri. Tetapi Pak Blangkon yang iseng menyuruh Pak Dengklek menebak sendiri apa DNA ayamnya tersebut.

DNA ayam-ayam tersebut dapat direpresentasikan sebagai sebuah string yang terdiri atas N huruf kapital dari A sampai dengan Z. **Semua huruf yang sama dalam sebuah string DNA pasti akan berada bersebelahan.** Sebagai contoh, AAASSSDDDDFF adalah sebuah string DNA, sedangkan AAASSSDDDAFFFF bukan karena di antara A terdapat kelompok huruf S dan D. Cara Pak Dengklek menebak adalah sebagai berikut: Pak Blangkon memberikan suatu bilangan N yang merupakan panjang string DNA dari ayam yang ingin Pak Dengklek ketahui DNA-nya. Selain itu, Pak Blangkon juga memberikan bilangan K yang merupakan batas pertanyaan yang boleh Pak Dengklek tanyakan sebelum Pak Dengklek harus menebak string DNA yang dimiliki ayam tersebut.

Setiap kali Pak Dengklek bertanya, ia akan memberikan suatu string. Setelah itu Pak Blangkon akan membalas YA atau TIDAK berdasarkan apakah string yang ditanyakan Pak Dengklek merupakan substring dari DNA ayam yang sedang ingin Pak Dengklek ketahui (YA jika substring dan TIDAK jika bukan). Sebagai informasi, X merupakan substring dari Y apabila X terdiri atas setidaknya satu karakter, dan X dapat dihasilkan dengan membuang nol atau lebih huruf-huruf awalan Y dan nol atau lebih huruf-huruf akhiran Y. Sebagai contoh, BEBE, EBE, BEK, dan BEBEK adalah substring dari BEBEK, sedangkan BEEK bukan substring dari BEBEK.

Pak Dengklek pun kebingungan dengan cara untuk menebak DNA ayam-ayam Pak Blangkon. Bantulah Pak Dengklek menentukan DNA ayam Pak Blangkon!

## Format Interaksi

Pada mulanya, program Anda akan menerima sebuah baris 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.

Kemudian, program Anda akan menerima sebuah baris berisi dua buah bilangan bulat  $N$  dan  $K$ . Setelah itu, program Anda dapat melakukan serangkaian tindakan, yang masing-masing merupakan salah satu dari:

- Bertanya. Program Anda harus mengeluarkan sebuah baris berisi **TANYA** `<pertanyaan>` yakni, string **TANYA** diikuti dengan substring yang Anda tanyakan. Setiap kali program Anda selesai bertanya, program Anda membaca sebuah string yang dijamin merupakan salah satu dari **YA** atau **TIDAK**, sesuai dengan definisi pada deskripsi soal.
- Menebak. Program Anda harus mengeluarkan sebuah baris berisi **JAWAB** `<jawaban>` yakni, string **JAWAB** diikuti dengan string DNA tebakan Anda. Program Anda **harus berhenti** setelah melakukan ini.

### Contoh Interaksi

Misalkan string DNA yang sebenarnya adalah **BAASSDDFF**. Berikut adalah contoh interaksi yang mungkin terjadi.

Keluaran Program Anda	Keluaran Program Grader
	0..... 10 3
TANYA ABCD	
	TIDAK
TANYA AASSDDFF	
	YA
TANYA DFFF	

Keluaran Program Anda	Keluaran Program Grader
	YA
JAWAB BAASSDDFFF	
	(interaksi selesai)

### Subsoal

Pada semua subsoal, berlaku:

- String DNA hanya terdiri atas huruf kapital A s.d. Z

#### Subsoal 1 (6 poin)

- $N = 10$
- $K = 10$
- String DNA hanya terdiri atas huruf A, B, atau C

#### Subsoal 2 (10 poin)

- $N = 10$
- $K = 350$

Khusus untuk subsoal 1 dan subsoal 2:

- Hanya terdapat sebuah kasus uji (satu subsoal dinyatakan oleh satu kasus uji), yang berupa game dan dapat dilihat di halaman pengumuman kontes.
- Anda boleh memainkan permainan ini berulang kali tanpa mendapatkan penalti.
- Jika Anda sudah memenangkan permainan untuk subsoal tertentu, Anda dapat memilih pilihan pada permainan untuk mengeluarkan *source code* yang dapat langsung Anda kirimkan ke *grader* dan menjawab dengan benar pada subsoal yang telah Anda menangkan.
- Anda tidak diwajibkan memainkan permainan ini untuk mengerjakan kedua subsoal ini. Anda diperbolehkan untuk menulis kode Anda sendiri untuk mengerjakan kedua subsoal ini.

### Subsoal 3 (31 poin)

- $N = 26$
- $K = 350$
- String DNA mengandung semua karakter dari A s.d. Z masing-masing tepat sekali

### Subsoal 4 (24 poin)

- $1 \leq N \leq 100$
- $K = 500$

### Subsoal 5 (29 poin)

- $1 \leq N \leq 1.000$
- $K = 750$

### Catatan

Yang perlu diperhatikan adalah bahwa untuk tipe soal interaktif seperti ini, Anda harus selalu memberikan perintah `fflush(stdout);` (bagi pengguna C/C++) atau `flush(output);` (bagi pengguna PASCAL) setiap kali Anda mencetak keluaran (dengan kata lain, setiap kali ada perintah mencetak keluaran misalnya `write`, `writeln`, `printf`, `cout`, atau `puts`, tepat di bawahnya harus ada perintah `fflush/flush`).

Sebagai contoh, berikut adalah contoh *source code* dalam bahasa Pascal yang akan selalu bertanya "TANYA ABC" dan kemudian menjawab "JAWAB DEF" tanpa mempedulikan nilai N dan K yang diberikan maupun bacaan hasil interaksi.

```
var subsoal, jawaban: string;
    N, K: longint;

begin
    readln(subsoal);
    readln(N, K);

    writeln('TANYA ABC');
    flush(output);

    readln(jawaban);

    writeln('JAWAB DEF');
    flush(output);
end.
```

Dan berikut adalah contoh *source code* yang melakukan hal yang sama dalam bahasa C++.

```
#include <cstdio>
```

```

#include <cstring>
char subsoal[100], jawaban[100];
int N, K;

int main() {
    gets(subsoal);
    scanf("%d %d", &N, &K);

    printf("TANYA ABC\n");
    fflush(stdout);

    gets(jawaban);

    printf("JAWAB DEF\n");
    fflush(stdout);

    return 0;
}

```

### Peringatan

Jika program Anda melakukan salah satu dari hal-hal di bawah ini:

- melakukan tindakan di luar format dan batasan yang ditentukan,
- bertanya lebih dari K kali, atau
- salah menjawab DNA

maka nilai Anda untuk subsoal yang bersangkutan adalah nol.

Selain itu, pastikan pula bahwa program Anda harus berhenti jika selesai melakukan interaksi. Bila tidak, maka Anda mungkin mendapatkan *Time Limit Exceeded*, *Runtime Error*, atau *Internal Error* untuk kasus uji yang bersangkutan.