

*Berkas  
Kompetisi  
Soal Hari 1*

Olimpiade Sains Nasional XI  
Bidang Komputer/Informatika  
2-7 September 2012, Jakarta



[www.tokilearning.org](http://www.tokilearning.org)



[www.siswapsma.org](http://www.siswapsma.org)

**SEGITIGA**

Batas Waktu	1 detik
Batas Memori	64 MB

Anda mungkin pernah mendengar soal ini. Diberikan sebuah segitiga angka-angka seperti di bawah ini:

```
      7
     3 8
    8 1 0
   2 7 4 4
  4 5 2 6 5
```

Kemudian, Anda mulai dari angka paling atas, lalu secara bertahap turun ke bawah. Setiap kali turun, Anda hanya boleh turun ke salah satu dari dua angka yang langsung bersebelahan sisi dengan angka sebelumnya. Anda ingin mengunjungi angka-angka sedemikian sehingga totalnya adalah semaksimal mungkin.

Banyak orang berpikir bahwa solusi optimal didapat dengan memilih bilangan yang lebih besar di antara dua bilangan yang mungkin pada setiap langkah (solusi *greedy*). Apabila kedua bilangan tersebut sama, ia memilih yang lebih kiri. Akan tetapi, solusi ini ternyata tidak selalu merupakan yang paling optimal.

Tahukah Anda, membuat *testcase* soal tidak semudah yang Anda bayangkan. Oleh karena itu, problem setter ingin berbagi kesedihan dengan Anda.

Pada soal ini, Anda diminta untuk membuat *testcase* soal segitiga tersebut sedemikian sehingga selisih dari solusi optimal dan solusi *greedy* adalah maksimum. Tentu saja, agar Anda lebih repot, bilangan-bilangan yang boleh Anda gunakan sudah diberikan (Anda harus menggunakan semua bilangan pada masukan masing-masing tepat sekali).

**Format Masukan**

Masukan diawali dengan sebuah baris berupa sebuah string dengan format "`Kasus #X`" (tanpa tanda kutip), di mana **X** adalah nomor subtask. Baris kedua masukan terdiri dari tepat sebuah bilangan bulat **N**, yang menyatakan tinggi segitiga. Baris ketiga berisi  $N * (N + 1) / 2$  bilangan bulat yang masing-masing terpisah tepat sebuah spasi, yakni bilangan-bilangan yang harus Anda gunakan pada segitiga. Tentu bisa ada dua atau lebih bilangan yang sama.

**Format Keluaran**

Baris ke-*i* pada keluaran berisi tepat *i* buah bilangan, tanpa diawali maupun diakhir spasi dan masing-masing angka bersebelahan terpisah tepat sebuah spasi, yang menyatakan baris ke-*i* dari segitiga pada solusi. Apabila ada lebih dari satu kemungkinan, keluarkan yang mana saja.

**Contoh Masukan 1**

```
Kasus #101
3
1 2 3 4 5 6
```

**Contoh Masukan 2**

```
Kasus #102
3
1 1 1 1 1 1
```

**Contoh Keluaran 1**

```
3
5 4
2 1 6
```

**Contoh Keluaran 2**

```
1
1 1
1 1 1
```

**Penjelasan Contoh**

Pada contoh masukan 1, perhatikan bahwa segitiga dibawah ini juga optimal.

```
5
4 3
2 1 6
```

Pada contoh kasus ini, solusi *greedy* menghasilkan jawaban  $5 + 4 + 2 = 11$ , sedangkan solusi optimal menghasilkan jawaban  $5 + 3 + 6 = 14$ . Selisih mereka adalah  $14 - 11 = 3$ , dan tidak ada solusi lain yang lebih baik.

Pada contoh kasus kedua, perhatikan bahwa seperti yang tertulis di deskripsi, apabila kedua bilangan yang tepat di bawah dan bersisian dengan bilangan yang dikunjungi sebelumnya adalah sama, maka solusi *greedy* akan memilih mengunjungi bilangan yang lebih kiri. Pada kasus ini, bilangan-bilangan yang dikunjungi pada solusi *greedy* adalah bilangan-bilangan paling kiri pada setiap baris. Pada solusi ini, solusi optimal dan solusi *greedy* sama-sama optimal sehingga memiliki selisih total bilangan 0.

**Penjelasan Subsoal**

Subsoal 1 (15 poin) : download kasus uji

Subsoal 2 (15 poin) : download kasus uji

Subsoal 3 (15 poin) : bilangan-bilangan masukan pasti adalah  $N * (N + 1) / 2$  bilangan asli pertama, yakni 1, 2, ...,  $N * (N + 1) / 2$ . Masukan tidak diberikan dalam urutan apapun (bisa acak).  $1 \leq N \leq 300$

Subsoal 4 (10 poin) :  $1 \leq N \leq 4$ , bilangan-bilangan ada di antara 0 dan 1 milyar, inklusif.

Subsoal 5 (15 poin) :  $1 \leq N \leq 10$ , bilangan-bilangan ada di antara 0 dan 1 milyar, inklusif.

Subsoal 6 (15 poin) :  $1 \leq N \leq 40$ , bilangan-bilangan ada di antara 0 dan 1 milyar, inklusif.

Subsoal 7 (15 poin) :  $1 \leq N \leq 300$ , bilangan-bilangan ada di antara 0 dan 1 milyar, inklusif.

**INVERSI MATRIKS**

Batas Waktu	1 detik
Batas Memori	32 MB

Diberikan sebuah matriks **S** yang berukuran **N x N** berisi 0 atau 1. Anda dapat melakukan sebuah inversi, sebagai berikut:

1. Pilihlah satu buah baris atau kolom.
2. Lakukan operasi XOR terhadap baris atau kolom tersebut dan matriks **X**. Matriks ini akan menjadi matriks **S** yang baru. Matriks **X** dijamin berukuran  $1 \times N$  (jika yang dipilih adalah baris) atau berukuran  $N \times 1$  (jika yang dipilih adalah kolom, dan matriks yang digunakan adalah  $X^T$  atau **X** transpose).

Contoh, misalkan matriks  $S = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$  dan matriks  $X = [0 \ 1 \ 1]$  (maka  $X^T = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ ).

Jika anda memilih baris kedua, maka matriks **S** akan menjadi  $\begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$

Jika anda memilih kolom pertama (sehingga operasi inversi akan dilakukan antara **S** dan  $X^T$ ),

maka matriks **S** akan menjadi  $\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$

Tugas anda adalah melakukan inversi sesedikit mungkin agar semua elemen pada matriks **S** berisi 0. Outputkan berapa banyak inversi minimal yang dibutuhkan.

**Format Masukan**

Baris pertama berisi string "Kasus #X" dengan **X** menyatakan nomor kasus uji. Baris kedua berisi bilangan bulat **N**.

Baris ketiga berisi **N** bilangan (0 atau 1) yang menyatakan matriks **X**. **N** baris berikutnya masing-masing berisi **N** bilangan (0 atau 1) yang menyatakan matriks **S** mula-mula.

**Format Keluaran**

Satu baris berisi bilangan bulat yang menyatakan inversi minimal yang dibutuhkan agar matriks  $S$  berisi 0 saja. Jika anda tidak mungkin membuat matriks  $S$  berisi 0 saja, outputkan “-1” (tanpa tanda kutip).

### Contoh Masukan

```
Kasus #0
0 1 1
0 0 0
1 0 1
1 1 0
```

### Contoh Keluaran

```
3
```

### Penjelasan

Matriks  $X$  pada contoh masukan sama dengan yang ada pada deskripsi soal.

Langkah invers yang anda bisa lakukan:

$$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Operasi logika XOR adalah operasi antara dua nilai kebenaran dan menghasilkan nilai true jika kedua nilai kebenaran tersebut tidak sama. Perhatikan tabel di bawah ini!

A	B	A XOR B
0 (FALSE)	0 (FALSE)	0 (FALSE)
0 (FALSE)	1 (TRUE)	1 (TRUE)
1 (TRUE)	0 (FALSE)	1 (TRUE)
1 (TRUE)	1 (TRUE)	0 (FALSE)

Jika  $A$  adalah matriks berukuran  $M \times N$  dan  $B$  adalah  $A^T$  (atau  $A$  transpose), maka  $B$  adalah matriks berukuran  $N \times M$  dengan  $B[i,j] = A[j,i]$  untuk  $i \leq N$  dan  $j \leq M$  ( $B[i,j]$  adalah elemen pada matriks  $B$  pada baris ke- $i$  dan kolom ke- $j$ ).

### Penjelasan Subsoal

Subsoal 1 (10 poin) : download kasus uji

Subsoal 2 (10 poin) : download kasus uji

- Subsoal 3 (7 poin) :  $1 \leq N \leq 5$
- Subsoal 4 (10 poin) :  $1 \leq N \leq 15$
- Subsoal 5 (7 poin) :  $1 \leq N \leq 100$ , semua elemen pada matriks X berisi 1.
- Subsoal 6 (20 poin) :  $1 \leq N \leq 100$
- Subsoal 7 (36 poin) :  $1 \leq N \leq 1000$





**Format Masukan**

Baris pertama berisi string "Kasus #X" (tanpa tanda kutip) dengan **X** menyatakan nomor kasus uji. Baris kedua terdiri dari tepat empat buah bilangan bulat  $r_1$ ,  $c_1$ ,  $r_2$ , dan  $c_2$  sesuai deskripsi soal, di mana masing-masing bilangan bersebelahan terpisah tepat sebuah spasi.

**Format Keluaran**

Keluaran terdiri dari tepat sebuah baris berisi sebuah bilangan bulat yang merupakan banyak lompatan minimum yang dibutuhkan NyanCoder untuk mengunjungi semua batu pada sungai itu setidaknya sekali.

**Contoh Masukan**

```
Kasus #0
167 50 174 59
```

**Contoh Keluaran**

```
5
```

**Penjelasan Contoh**

Untuk contoh masukan, perhatikan ilustrasi di bawah ini yang menunjukkan bagian matriks antara baris  $r_1$  hingga baris  $r_2$  dan kolom  $c_1$  hingga kolom  $c_2$ .

	50	51	52	53	54	55	56	57	58	59
167	0	0	0	1	0	1	0	0	1	1
168	0	0	0	1	0	1	0	1	0	0
169	0	0	0	1	0	1	0	1	0	0
170	0	0	0	1	0	1	0	1	0	1
171	0	0	0	1	0	1	0	1	0	1
172	0	0	0	1	0	1	0	1	1	0
173	0	0	0	1	0	1	0	1	1	0
174	0	0	0	1	0	1	0	1	1	1

Salah cara melompat untuk mendapatkan banyak lompatan minimum adalah sebagai berikut.

- Pertama-tama, lompat ke batu di pojok kanan atas matriks di atas (terkanan pada baris 167). Kemudian, telusuri semua batu yang bisa ditelusuri tanpa melompat (semua batu berwarna merah).
- Lalu lompat untuk kedua kalinya ke batu pada baris 170 dan kolom 59 (semua batu berwarna oranye).
- Kemudian lompat ketiga kalinya ke batu pada baris 174 dan kolom 59 lalu telusuri semua batu yang bisa ia telusuri tanpa melompat (semua batu berwarna ungu).
- Lalu lompat keempat kalinya ke salah satu batu pada kolom ke-55 dan telusuri semua batu yang bisa ia telusuri tanpa melompat (semua batu berwarna biru).
- Terakhir, lompat untuk kelima kalinya ke salah satu batu pada kolom 53 dan telusuri semua batu yang tersisa tanpa melompat (semua batu berwarna hijau).

### Penjelasan Subsoal

Subsoal 1 (15 poin): download kasus uji

Subsoal 2 (15 poin): download kasus uji

Subsoal 3 (10 poin):  $1 \leq r_1 \leq r_2 \leq 100$ ,  $c_1 = 1$ , dan  $c_2 = 60$ .

Subsoal 4 (10 poin):  $r_1 = 1$ ,  $1 \leq r_2 \leq 2^{60}-1$ ,  $c_1 = 1$ , dan  $c_2 = 60$ .

Subsoal 5 (10 poin):  $r_1 = 1$ ,  $1 \leq r_2 \leq 2^{60}-1$ ,  $1 \leq c_1 \leq 60$ , dan  $c_2 = 60$ .

Subsoal 6 (10 poin):  $r_1 = 1$ ,  $1 \leq r_2 \leq 2^{60}-1$ ,  $c_1 = 1$ , dan  $1 \leq c_2 \leq 60$ .

Subsoal 7 (10 poin):  $r_1 = 1$ ,  $1 \leq r_2 \leq 2^{60}-1$ , dan  $1 \leq c_1 \leq c_2 \leq 60$ .

Subsoal 8 (10 poin):  $1 \leq r_1 \leq r_2 \leq 2^{60}-1$ ,  $c_1 = 1$ , dan  $c_2 = 60$ .

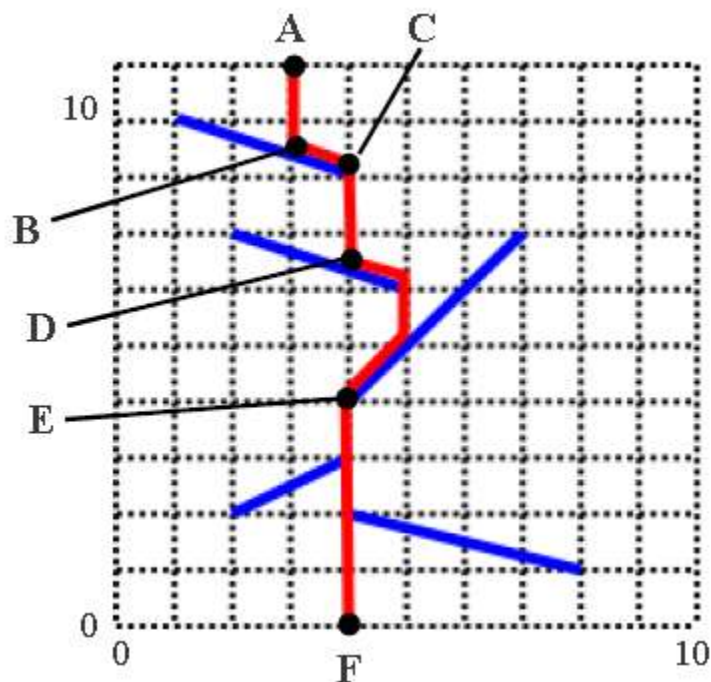
Subsoal 9 (10 poin):  $1 \leq r_1 \leq r_2 \leq 2^{60}-1$ , dan  $1 \leq c_1 \leq c_2 \leq 60$ .

**MALING**

Batas Waktu	1 detik
Batas Memori	32 MB

“Maliiiiing!!!!!!” terdengar suara dari kamar paling atas di Hotel Dengklek. Pencurinya diketahui kabur melalui jendela di kamar tersebut. Hotel Dengklek memang unik, di mana terdapat semacam perosotan di luar jendela kamar sebagai jalan keluar darurat jika terjadi kebakaran. Sayangnya kali ini, perosotan tersebut malah dipakai oleh pencuri untuk kabur.

Perosotan darurat tersebut terdiri dari segmen-segmen lurus. Tiap segmen akan selalu miring (tidak akan ada segmen yang tepat mendatar atau tepat vertikal). Jika Anda berada di tengah suatu segmen, Anda akan otomatis merosot ke arah yang lebih rendah. Segmen-segmen tersebut juga tidak selalu berhubungan, dan jika Anda berada di ujung segmen, Anda akan jatuh ke bawah hingga mengenai segmen yang lebih bawah, atau hingga Anda mencapai tanah, manapun yang lebih dulu terjadi.



Dalam ilustrasi di atas, segmen perosotan diwarnai biru, sementara warna merah menunjukkan rute meluncurnya maling dari jendela kamar (titik A) hingga mencapai tanah. Saat maling kabur dari jendela di titik A, ia terjatuh hingga mendarat di segmen perosotan di titik B. Maling tersebut meluncur ke arah yang lebih rendah hingga ujung segmen di titik C. Dari situ, ia terjatuh hingga mendarat di segmen berikutnya di titik D. Hal ini terus terjadi hingga maling menyentuh tanah (ditandai dengan koordinat  $y = 0$ ). Perlu diperhatikan

bahwa saat maling terjatuh dari titik E, ia terjatuh langsung ke titik F walaupun ia menyentuh dua segmen di bawahnya. Ini disebabkan karena ia hanya menyentuh ujung dari segmen-segmen tersebut dan bukan mendarat di tengah-tengah segmen.

Nah, persoalannya tidak selesai di situ. Mendengar teriakan tadi,  $N$  orang polisi bergegas mengejar maling tersebut. Namun, karena reaksi mereka agak lambat, mereka baru mulai mengejar maling  $T$  detik setelah maling melompat dari jendela kamar. Lalu,  $N$  orang polisi tersebut juga mulai mengejar dari jendela yang berbeda-beda (tetapi tidak menutup kemungkinan mereka mulai dari jendela yang sama). Baik maling maupun polisi akan merosot atau terjatuh dengan kecepatan  $V$  unit/detik. Untuk mempermudah, dianggap kecepatan terjatuh adalah tetap dan tidak dipengaruhi oleh efek gravitasi atau gesekan.

Karena polisi dan maling dapat mulai dari jendela yang berbeda-beda, mereka dapat mencapai tanah pada lokasi yang berbeda-beda pula. Begitu polisi mencapai tanah, ia dapat berlari dengan kecepatan  $R$  petak/detik. Berapa kecepatan lari maling minimal sehingga ia tidak tertangkap oleh polisi? Untuk soal ini, definisi tertangkap adalah jika maling bertemu dengan polisi di lokasi yang sama setelah maling mencapai tanah (polisi tidak dapat menangkap maling saat mereka masih merosot/terjatuh). Ingat, maling dapat kabur baik ke arah kanan (sumbu  $x$  positif) maupun kiri (sumbu  $x$  negatif).

### Format Masukan

Baris pertama berisi string "`Kasus #X`" (tanpa tanda kutip) dengan  $X$  menyatakan nomor kasus uji.

Baris kedua berisi bilangan-bilangan  $S$ ,  $N$ ,  $T$ ,  $R$ , dan  $V$  ( $0 \leq S, T \leq 1000$ ,  $0 \leq N \leq 20$ ,  $0 < R, V \leq 1000$ ), masing-masing dipisahkan spasi. Nilai  $S$  adalah jumlah segmen perosotan yang ada di Hotel Dengklek. Nilai  $S$  dan  $N$  akan selalu berupa bilangan bulat.

Baris ketiga berisi dua buah bilangan bulat  $X_0$  dan  $Y_0$  yang merupakan koordinat jendela tempat maling kabur.

$N$  baris berikutnya masing-masing berisi dua buah bilangan bulat  $X_i$  dan  $Y_i$  yang merupakan koordinat jendela tempat polisi ke- $i$  mulai mengejar.  $S$  baris berikutnya masing-masing berisi empat buah bilangan bulat  $A$ ,  $B$ ,  $C$ ,  $D$  yang menyatakan segmen perosotan dengan ujung di koordinat  $(A, B)$  dan  $(C, D)$ . Setiap segmen akan mempunyai panjang positif. Tidak ada dua segmen yang ujung-ujungnya bersentuhan, dan juga tidak ada dua segmen yang bersinggungan atau menyilang. Tidak ada ujung segmen yang menyentuh tanah. Semua nilai koordinat di masukan akan berupa bilangan bulat non-negatif yang tidak lebih dari 1000.

**Format Keluaran**

Baris pertama berisi bilangan **P** yang merupakan koordinat sumbu x dari tempat maling menjejak tanah pertama kali (koordinat sumbu y nya tentu saja 0).

Isi baris kedua tergantung apakah maling bisa kabur dari kejaran polisi atau tidak. Jika maling mungkin dapat kabur dari kejaran polisi, tuliskan "KABUR" (tanpa tanda kutip) diikuti dengan kecepatan minimum lari maling dipisahkan dengan sebuah spasi. Jika maling tidak mungkin kabur dari kejaran polisi, tuliskan "TERTANGKAP" (tanpa tanda kutip).

**Contoh Masukan**

Kasus #0

5 2 7.0 2.0 1.0

3 10

3 5

7 5

4 2 8 1

2 2 4 3

7 7 4 4

2 7 5 6

4 8 1 9

**Contoh Keluaran**

4

KABUR 2.62363215

**Penjelasan**

Contoh di atas adalah sesuai dengan ilustrasi yang digambarkan (lokasi polisi tidak ada di ilustrasi).

Perbedaan perhitungan absolut di bawah  $10^{-6}$  akan diterima.

**Penjelasan Subsoal**

Subsoal 1 (15 poin) : download kasus uji.

Subsoal 2 (15 poin) : download kasus uji.

Subsoal 3 (20 poin) : Semua masukan (S, N, T, R, V, Xi, Yi) dan keluaran dijamin bilangan bulat.

Subsoal 4 (20 poin) : Semua masukan (S, N, T, R, V, Xi, Yi) dan keluaran dijamin bilangan bulat.

Subsoal 5 (30 poin) : Masukan T, R, V dan angka dapat berupa bilangan pecahan

*Berkas  
Kompetisi  
Soal Hari 2*

Olimpiade Sains Nasional XI  
Bidang Komputer/Informatika  
2-7 September 2012, Jakarta



[www.tokilearning.org](http://www.tokilearning.org)



[www.siswapsma.org](http://www.siswapsma.org)

**BARISAN BILANGAN**

Batas Waktu	1 detik
Batas Memori	32 MB

Diberikan sebuah barisan bilangan yang terdiri dari  $N$  buah bilangan bulat:  $a_1, a_2, \dots, a_N$ .

Sebanyak tepat  $K$  kali, Pak Dengklek akan mengambil dua bilangan bersebelahan, menghapus mereka, kemudian mengganti keduanya dengan bilangan baru yang merupakan hasil penjumlahan kedua bilangan tersebut. Sebagai contoh, pada barisan bilangan 1, 2, 3, 4, 5, Pak Dengklek dapat mengambil bilangan bulat ke-2 dan 3 dan menggantinya menjadi: 1, 5, 4, 5.

Bantulah Pak Dengklek agar barisan bilangan yang dihasilkan memiliki selisih nilai maksimum dan nilai minimum sekecil mungkin!

**Format Masukan**

Baris pertama berisi string "Kasus #X" (tanpa tanda kutip) dengan  $X$  menyatakan nomor kasus uji.

Baris kedua terdiri dari tepat dua buah bilangan  $N$  dan  $K$  yang dipisahkan sebuah spasi.

Baris ketiga dan sekaligus baris terakhir masukan terdiri dari  $N$  buah bilangan bulat yang masing-masing terpisah tepat sebuah spasi yang menyatakan barisan Pak Dengklek, sesuai urutannya.

**Format Keluaran**

Baris pertama dan satu-satunya keluaran berisi selisih terkecil yang mungkin dari maksimum dan minimum.

**Contoh Masukan**

Kasus #101

6 3

5 1 1 1 3 2



**Contoh Keluaran**

2

**Penjelasan Contoh**

Pada contoh di atas, solusi optimal didapat sebagai berikut:

- Gabungkan bilangan kedua dan ketiga menjadi: 5 2 1 3 2.
- Gabungkan bilangan kedua dan ketiga menjadi: 5 3 3 2.
- Gabungkan kedua bilangan terakhir menjadi: 5 3 5.

Nilai maksimumnya adalah 5, nilai minimumnya adalah 3, dan selisihnya adalah 2. Sehingga keluaran untuk masukan ini adalah 2.

**Penjelasan Subsoal**

Subsoal 1 (X poin): download kasus uji

Subsoal 2 (X poin): download kasus uji

Subsoal 3 (X poin):  $1 \leq K < N \leq 10, K \leq 5, 1 \leq a_i \leq 10^6$ .

Subsoal 4 (X poin):  $1 \leq K < N \leq 20, 1 \leq a_i \leq 10^6$ .

Subsoal 5 (X poin):  $1 \leq K < N \leq 40, 1 \leq a_i \leq 2$ .

Subsoal 6 (X poin):  $1 \leq K < N \leq 40, 1 \leq a_i \leq 10^6$ .

**BATAS NILAI**

Batas Waktu	1 detik
Batas Memori	64 MB

Pada sebuah olimpiade, panitia ingin memilih  $N$  dari  $M$  peserta tingkat provinsi untuk berpartisipasi di tingkat nasional. Terdapat  $Q$  provinsi di negara tersebut, dinomori dari 1 hingga  $Q$ . Misalkan  $K$  adalah maksimal banyak peserta tingkat nasional yang berasal dari suatu provinsi.

Koordinator olimpiade ini, Pak Dengklek, ingin memilih  $K$  buah bilangan bulat  $a_1, a_2, \dots, a_K$  sebagai *passing grade* sedemikian sehingga:

1. Seluruh peserta dengan peringkat provinsi  $i$ , akan lolos jika dan hanya jika nilainya tidak kurang dari  $a_i$ .
2. Apabila pada provinsi  $X$  peserta dengan peringkat  $i$  lolos, maka semua peserta dengan peringkat  $j < i$  harus lolos. Dengan kata lain, apabila nilai peserta dengan peringkat  $i$  tidak kurang dari  $a_i$ , nilai peserta dengan peringkat  $j < i$  tidak boleh kurang dari  $a_j$ .
3. Untuk memastikan kualitas, *passing grade* juga harus dipilih sedemikian sehingga  $T$  peserta terbaik secara nasional lolos.
4. Banyak peserta yang lolos ke tingkat nasional tepat  $N$ . Tidak kurang, tidak lebih. Dan tentu saja,  $N$  peserta ini belum tentu merupakan  $N$  peserta terbaik secara nasional (perhatikan kriteria lain untuk pemilihan *passing grade*).
5. Selisih antara banyak peserta yang lolos di provinsi yang memiliki peserta lolos terbanyak dengan banyak peserta yang lolos di provinsi yang memiliki peserta lolos tersedikit harus sekecil mungkin.

Tentu saja, peserta yang mendapatkan peringkat lebih tinggi (peringkat dengan bilangan lebih kecil) adalah peserta dengan nilai lebih besar. Bantulah panitia olimpiade tersebut untuk menentukan *passing grade* yang memenuhi semua batasan di atas.

**Format Masukan**

Baris pertama berisi string "Kasus #X" (tanpa tanda kutip) dengan  $X$  menyatakan nomor kasus uji.

Baris kedua berisi empat buah bilangan bulat dipisahkan spasi **N**, **M**, **Q**, dan **T** ( $T \leq N \leq M$ ) sesuai deskripsi soal.

Baris ke-**i** dari **M** baris berikutnya berisi dua buah bilangan bulat dipisahkan spasi, **S<sub>i</sub>** ( $0 \leq S_i \leq 100000$ ) dan **P<sub>i</sub>** ( $1 \leq P_i \leq Q$ ) yang masing-masing menyatakan nilai peserta ke-**i** dan provinsi asal peserta tersebut.

Dipastikan tidak ada dua peserta yang memiliki nilai yang persis sama. Selain itu, masing-masing provinsi pasti memiliki setidaknya seorang peserta.

### Format Keluaran

Sebanyak **K** buah bilangan bulat pada sebuah baris yang masing-masing dipisahkan tepat sebuah spasi, yang merupakan **a<sub>1</sub>**, **a<sub>2</sub>**, ..., **a<sub>K</sub>** secara berturut-turut. Masing-masing **a<sub>i</sub>** harus berada di antara 0 dan 100001, inklusif. Apabila ada lebih dari satu kemungkinan keluaran, keluarkan yang mana saja.

### Contoh Masukan

Kasus #0

6 8 3 5

30 2

70 1

50 1

60 3

80 3

10 1

40 3

20 2

### Contoh Keluaran

30 25 30

**Penjelasan**

Perhatikan bahwa data pada contoh kasus di atas sama dengan data di bawah ini.

- Provinsi 1: **70, 50, 10**
- Provinsi 2: 30, 20
- Provinsi 3: **80, 60, 40**

Angka yang dicetak tebal adalah nilai peserta yang masuk **T** terbaik secara nasional. Untuk masing-masing provinsi, nilai diurutkan berdasarkan peringkat, dari peringkat 1, 2, 3, hingga seterusnya.

Pada contoh di atas, **K** = 3, yakni banyaknya peserta pada provinsi 1 ataupun 3. Pada solusi contoh masukan, karena  $a_1 = 30$ , maka peringkat pertama di semua provinsi lolos. Karena  $a_2 = 25$ , hanya peringkat 2 pada provinsi 2 dan provinsi 3 yang lolos. Karena  $a_3 = 30$ , hanya peringkat 3 pada provinsi 3 yang lolos.

Perhatikan bahwa solusi ini memenuhi semua ketentuan soal.

Selisih antara banyak peserta yang lolos di provinsi yang memiliki peserta lolos terbanyak (provinsi 3, yang memiliki 3 peserta lolos) dengan banyak peserta yang lolos di provinsi yang memiliki peserta lolos tersedikit (provinsi 2, yang hanya memiliki 1 peserta lolos) adalah  $3 - 1 = 2$ , yakni nilai terkecil yang mungkin.

**Penjelasan Subsoal**

Subsoal 1 (10 poin): download kasus uji.

Subsoal 2 (10 poin): download kasus uji.

Subsoal 3 (15 poin):  $1 \leq N \leq M \leq 100, 1 \leq Q \leq 100, T = 0$ .

Subsoal 4 (15 poin):  $1 \leq T \leq N \leq M \leq 50, 1 \leq Q \leq 5, 1 \leq K \leq \min(10, M)$ .

Subsoal 5 (15 poin):  $1 \leq T \leq N \leq M \leq 100, 1 \leq Q \leq 100, 1 \leq K \leq M$ .

Subsoal 6 (35 poin):  $1 \leq T \leq N \leq M \leq 10000, 1 \leq Q \leq 100, 1 \leq K \leq M$ .

**MENGGAMBAR**

Batas Waktu	Interaktif (1 detik)
Batas Memori	16 MB

Pak Dengklek sedang belajar menggambar bersama **M** ekor bebeknya. Gambar Pak Dengklek sangatlah unik, dia menggambar menggunakan titik pada bidang  $N \times N$  (dimana  $N$  adalah kelipatan pangkat 2 [ $2^y = N$ ,  $y$  bil bulat positif],  $N \leq 256$ ). Warna pada gambar Pak Dengklek hanya ada 2 (angka 0 dan angka 1).

Pak Dengklek menggambar dengan salah satu dari tiga cara sebagai berikut:

1. Mewarnai seluruh bidang gambar dengan 0.
2. Mewarnai seluruh bidang gambar dengan 1.
3. Membagi menjadi empat bagian gambar sama luas, dimana satu bagian diwarnai dengan 0, satu bagian lagi diwarnai dengan 1, dan kedua bagian sisanya dianggap sebagai bidang baru untuk digambar.

**Catatan:** langkah no.3 hanya berlaku untuk bagian dengan jumlah titik lebih dari 1

Karena para bebek tidak mengetahui gambar Pak Dengklek, mereka berusaha menebak gambar aslinya. Untuk setiap tebakan Pak Dengklek akan memberitahu banyak titik dengan warna yang berbeda. Karena menggambar sangatlah melelahkan, maka seekor bebek hanya bisa menggambar 1 kali saja. Bantulah para bebek mengatur strategi untuk menebak gambar yang dibuat Pak Dengklek.

**Format Interaksi**

Pada awalnya grader memberikan nilai **N**. Untuk setiap jawaban yang diberikan, grader akan mengeluarkan selisihnya.

Jika program anda menebak lebih dari jumlah bebek, program akan langsung diberhentikan.

**Contoh Interaksi**

Keluaran Anda	Keluaran Juri
<pre> 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111 </pre>	<pre> Kasus #0 8 </pre>
<pre> 11110011 11110011 11110000 11110000 00000010 00000001 00001011 00000111 </pre>	<pre> 36 </pre>
	<pre> selesai </pre>
<pre> Program anda benar dan langsung dihentikan </pre>	

**Penjelasan Subsoal**

Subsoal 1.3: [masing-masing 7 poin]

- $1 \leq N \leq 8$
- $M = 1000000$
- <subsoal 1>, <subsoal 2>, <subsoal 3>

Subsoal 4: [15 poin]

- $1 \leq N \leq 8$
- $M = 7000$

Subsoal 5: [15 poin]

- $1 \leq N \leq 256$
- $M = 70000$

Subsoal 6: [24 poin]

- $1 \leq N \leq 256$
- $M = 4100$

Subsoal 7: [25 poin]

- $1 \leq N \leq 256$
- $M = 1100$

**Peringatan**

**Jika program anda menebak lebih dari M kali, program anda akan langsung dihentikan**

**MEMBACA**

Batas Waktu	1 detik
Batas Memori	32 MB

Pak Dengklek sedang mengajarkan bebeknya membaca huruf-huruf. Begitu cintanya Pak Dengklek akan Tim Olimpiade Komputer Indonesia, huruf-huruf pertama yang ia ajarkan pada bebeknya adalah T, O, K, dan I. Walau tampak mudah, bebek Pak Dengklek mengalami kesulitan untuk memahami beberapa bentuk yang serupa tapi tak sama yang menyatakan suatu huruf yang sama. Sebagai contoh, beberapa bentuk yang serupa tapi tak sama berikut ini sama-sama menyatakan "TOKI".



Pada persoalan ini, satu huruf dinyatakan oleh satu matriks berukuran  $N \times N$ . Masing-masing sel pada matriks bernilai 1 atau 0. Setiap sel bernilai 1 adalah bagian dari huruf yang bersangkutan, sedangkan setiap sel bernilai 0 adalah background atau latar dimana huruf dituliskan. Sebagai contoh, matriks berukuran  $10 \times 10$  berikut ini menyatakan huruf T.

```

1111111111
1111111111
0000110000
0000110000
0000110000
0000110000
0000110000
0000110000
0000110000
0000110000

```

Jika setiap sel pada matriks diubah menjadi piksel pada gambar, maka gambar berikut ekuivalen dengan matriks di atas.



Bantulah bebek Pak Dengklek mengenali huruf T, O, K, dan I dengan membaca matriks seperti yang dijelaskan di atas dan menebak huruf yang dinyatakan.

Untuk setiap kasus, dijamin bahwa:

- Huruf T pada dasarnya terdiri dari satu garis vertikal dan satu garis horizontal pada bagian paling atas,
- Huruf O pada dasarnya terdiri dari titik-titik yang membentuk satu garis melingkar,



- Huruf K pada dasarnya terdiri dari satu garis vertikal, satu garis diagonal dari tengah garis vertikal ke kanan atas, satu garis diagonal dari tengah garis vertikal ke kanan bawah,
- Huruf I pada dasarnya terdiri dari satu garis vertikal saja,
- Matriks pada masukan selalu menyatakan suatu huruf di antara T, O, K, atau I (tidak mungkin menyatakan huruf lain atau tidak menyatakan suatu huruf apapun sama sekali),
- Dengan definisi **a** adalah indeks baris paling atas yang memiliki digit 1, **b** adalah indeks baris paling bawah yang memiliki digit 1, **c** adalah indeks kolom paling kiri yang memiliki digit 1, **d** adalah indeks baris paling kanan yang memiliki digit 1,  $(\text{abs}(\mathbf{a}-\mathbf{b}) * \text{abs}(\mathbf{c}-\mathbf{d})) / (\mathbf{N} * \mathbf{N}) \geq 25\%$ ,

Jika setiap sel pada matriks diubah menjadi piksel pada gambar, maka gambar yang ekuivalen dengan matriks tersebut akan dapat dikenali dengan mudah oleh manusia.

### Format Masukan

Baris pertama berisi string "Kasus #X" (tanpa tanda kutip) dengan **X** menyatakan nomor kasus uji.

Baris kedua berisi sebuah bilangan bulat **N** ( $20 \leq \mathbf{N} \leq 200$ ) yang menyatakan ukuran dari matriks.

**N** baris berikutnya masing-masing berisi karakter '1' atau '0' (tanpa tanda kutip).

### Format Keluaran

Sebuah karakter, salah satu di antara 'T', 'O', 'K', 'I' (tanpa tanda kutip).

**Contoh Masukan**

Kasus #0

20

00111111111111111100

01111111111111111110

11100000000000000111

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11000000000000000011

11100000000000000111

01111111111111111110

00111111111111111100

**Contoh Keluaran**

O

**Penjelasan Subsoal**

Subsoal 1 (18 poin) : download kasus uji.

Subsoal 2 (19 poin) : Seluruh sel bernilai 1 membentuk suatu bentuk tanpa potongan. Ada sel bernilai 1 pada baris pertama, baris terakhir, kolom pertama, dan kolom terakhir.

Subsoal 3 (20 poin) : Seluruh sel bernilai 1 membentuk suatu bentuk tanpa potongan.

Subsoal 4 (21 poin) : Ada sel bernilai 1 pada baris pertama, baris terakhir, kolom pertama, dan kolom terakhir.

Subsoal 5 (22 poin) : Tidak ada tambahan batasan selain yang dijelaskan di atas.