Berkas
Kompetisi
Soal Hari 1

Olimpiade Sains Nasional XI

Bidang Komputer/Informatika

2-7 September 2012, Jakarta

www.tokilearning.org   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 semaksimum 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). $1 \leq \mathbf{N} \leq 300$

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

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

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

Subsoal 7 (15 poin) : $1 \leq \mathbf{N} \leq 300$, 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 x **N** (jika yang dipilih adalah baris) atau berukuran **N** × 1 (jika yang dipilih adalah kolom, dan matriks yang digunakan adalah $\mathbf{X}^{\mathbf{T}}$ atau **X** transpose).

Contoh, misalkan matriks $\mathbf{S} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ dan matriks $\mathbf{X} = [0 \quad 1 \quad 1]$ (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 $\mathbf{X}^{\mathbf{T}}$),,
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 × N dan B adalah A^T (atau A transpose), maka B adalah matriks berukuran N × M dengan B[i,j] = A[j,i] untuk i ≤ N dan j ≤ 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) : $1 \leq \mathbf{N} \leq 5$

Subsoal 4 (10 poin) : $1 \leq \mathbf{N} \leq 15$

Subsoal 5 (7 poin) : $1 \leq \mathbf{N} \leq 100$, semua elemen pada matriks X berisi 1.

Subsoal 6 (20 poin) : $1 \leq \mathbf{N} \leq 100$

Subsoal 7 (36 poin) : $1 \leq \mathbf{N} \leq 1000$

SUNGAI BINER

Batas Waktu	1 detik
Batas Memori	64 MB

Di peternakan Pak Dengklek terdapat sungai yang mengalir. Pada sungai itu terdapat pula batu-batu. Setelah diamati, sungai tersebut beserta batu-batunya dari ujung awal ke ujung akhirnya membentuk barisan biner dari 1 hingga 2^{60}-1 dengan lebar 60 bit, di mana bit 1 merupakan batu dan bit 0 merupakan air. Untuk memudahkan, kita anggap sungai ini sebagai matriks dengan 2^{60}-1 baris dan 60 kolom sehingga baris ke-i pada matriks ini merupakan representasi biner dari bilangan i dan kolom ke-j pada matriks ini adalah kolom ke-j dari paling kiri. Untuk lebih jelasnya, perhatikan ilustrasi berikut ini.

```
          1         2         3         4         5         6
  123456789012345678901234567890123456789012345678901234567890
1 000000000000000000000000000000000000000000000000000000000001
2 000000000000000000000000000000000000000000000000000000000010
3 000000000000000000000000000000000000000000000000000000000011
4 000000000000000000000000000000000000000000000000000000000100
5 000000000000000000000000000000000000000000000000000000000101
  ...
```

NyanCoder, kucing Pak Dengklek yang nakal, ingin mengunjungi semua batu yang ada di antara baris ke-r_1 hingga baris ke-r_2 dan antara kolom ke-c_1 hingga kolom ke-c_2 dari matriks sungai tersebut. Setiap batu harus dikunjungi setidaknya sekali, dan tentu boleh beberapa kali. Tentu saja, ia tidak bisa berpijak di atas air karena ia takut dengan air. Selain itu, ia tidak boleh berpijak di batu-batu lain selain yang ada di antara baris ke-r_1 hingga baris ke-r_2 dan antara kolom ke-c_1 hingga kolom ke-c_2.

NyanCoder tidak perlu melompat ketika berpindah dari suatu batu ke batu lain yang bersisian dengannya baik di utara, selatan, timur, maupun barat. Pada ilustrasi matriks di atas, atas menunjukkan arah utara. Akan tetapi, ia akan melompat tepat satu kali ketika berpindah ke batu lain manapun yang tidak bersisian dengan batu yang sedang ia pijak. Perlu diketahui bahwa NyanCoder sangat atletis dan bisa melompat sejauh apapun.

Di awal, NyanCoder berada pada tepi sungai di sebelah kanan matriks tersebut. Pada mulanya, ia harus melompat sekali menuju salah satu batu terlebih dahulu.

Hitunglah banyak lompatan minimum yang dibutuhkan NyanCoder untuk dapat mengunjungi semua batu.

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 $\mathbf{r_1}$, $\mathbf{c_1}$, $\mathbf{r_2}$, dan $\mathbf{c_2}$ 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 loncatan 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 $\mathbf{r_1}$ hingga baris $\mathbf{r_2}$ dan kolom $\mathbf{c_1}$ hingga kolom $\mathbf{c_2}$.

```
    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): $1 \le r_1 \le r_2 \le 100$, $c_1 = 1$, dan $c_2 = 60$.

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

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

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

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

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

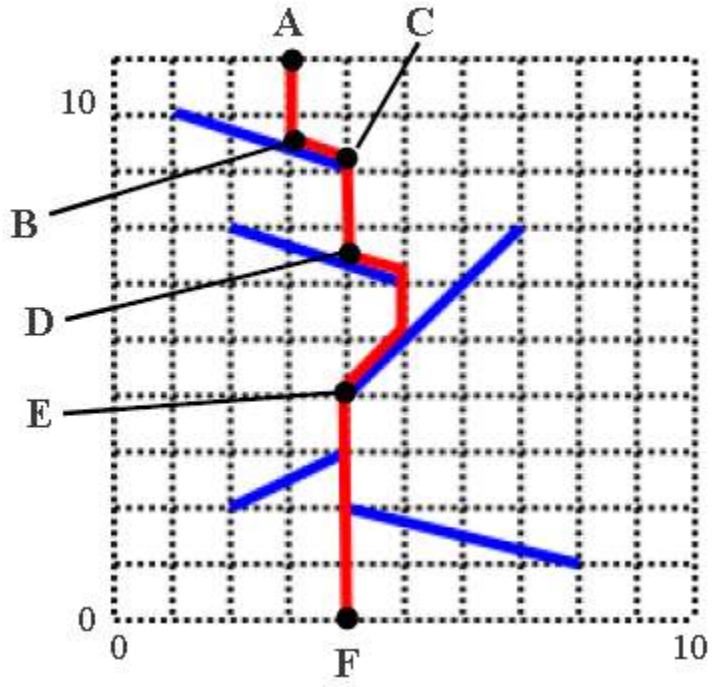
Subsoal 9 (10 poin): $1 \le r_1 \le r_2 \le 2^{60}-1$, dan $1 \le c_1 \le c_2 \le 60$.

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** ($0 \le$ **S, T** ≤ 1000, $0 \le$ **N** ≤ 20, $0 <$ **R**, **V** ≤ 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 $\mathbf{X_0}$ dan $\mathbf{Y_0}$ yang merupakan koordinat jendela tempat maling kabur.

N baris berikutnya masing-masing berisi dua buah bilangan bulat $\mathbf{X_i}$ dan $\mathbf{Y_i}$ 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