

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.