

Angka Tebak

Batas Waktu	1 detik
Batas Memori	32 MB

Deskripsi

Semoga Anda masih ingat soal **Tebak Angka** di mana Anda diharuskan menebak angka yang dipikirkan Pak Dengklek (dari rentang 1 hingga N) dengan menebak maksimal Q kali dan Pak Dengklek akan menjawab TERLALU KECIL, TERLALU BESAR, atau SELAMAT apabila berhasil menebak angka yang dipikirkan.

Sekarang, mari kita bertukar posisi!

Anda diminta untuk memikirkan sebuah angka dari rentang 1 hingga N. Anda dinyatakan menang apabila Pak Dengklek tidak dapat menebak angka yang Anda pikirkan hingga akhir tebakan (tebakan ke-Q).

Sama seperti strategi Anda pada soal **Tebak Angka**, Pak Dengklek memakai prinsip *Divide & Conquer* untuk menebak angka yang Anda pikirkan. Apabila angka yang Anda pikirkan berada pada rentang A hingga B, maka Pak Dengklek akan menebak $C = (A + B) / 2$ sebagai angka yang Anda pikirkan. Apabila C bukan berupa bilangan bulat, maka secara acak Pak Dengklek akan memilih pembulatan ke atas atau pembulatan ke bawah. Tentu saja apabila Anda menjawab TERLALU BESAR maka angka yang Anda pikirkan berada pada rentang A hingga C - 1 dan apabila Anda menjawab TERLALU KECIL maka angka yang Anda pikirkan berada pada rentang C + 1 hingga B.

Perlu diperhatikan bahwa Pak Dengklek dapat mengecek apakah jawaban-jawaban Anda konsisten sehingga kecurangan-kecurangan yang bisa saja Anda lakukan secara otomatis akan langsung terdeteksi. Salah satu contoh kecurangan yaitu Anda menjawab TERLALU BESAR ketika Pak Dengklek menebak angka 1.

Format Interaksi

Pada awalnya, program Anda akan menerima label kasus uji. Label kasus uji berisi sebuah *string* yang dijelaskan sebagai berikut:

- Panjang *string* tersebut adalah banyaknya subsoal ditambah satu.
- Digit pertama dari label adalah karakter ke-0, digit kedua dari label adalah karakter ke-1, dan seterusnya.
- Karakter ke-0 akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, dan berisi '.' 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 '.'

Sebagai contoh apabila label sebuah kasus uji sebuah soal adalah 0 . . 3 4 5, 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.

Selanjutnya, program Anda akan menerima input dua buah bilangan N dan Q.

Kemudian, Anda akan menerima sebuah bilangan bulat C yaitu tebakan Pak Dengklek untuk angka yang Anda pikirkan. Anda diminta untuk menjawab TERLALU KECIL, TERLALU BESAR, atau SELAMAT berdasarkan angka yang Anda pikirkan.

Apabila Anda menjawab SELAMAT, artinya Pak Dengklek berhasil menebak angka yang Anda pikirkan dan Anda dinyatakan kalah. Apabila Anda menjawab selain SELAMAT, maka Pak Dengklek akan menebak kembali angka yang Anda pikirkan.

Anda dinyatakan menang apabila hingga tebakan ke-Q Pak Dengklek masih belum mendapatkan jawaban SELAMAT.

Contoh Interaksi

Keluaran Program Peserta	Umpan Balik Grader
	0 . . . 4 .
	9 3
	5
TERLALU BESAR	
	2
TERLALU KECIL	
	4
TERLALU BESAR	
	(interaksi selesai)

Penjelasan Contoh Interaksi

Pada kasus tersebut, nilai N dan Q secara berturut-turut adalah 9 dan 3. Dari angka 1 hingga 9, Anda memilih angka 3.

Untuk tebakan pertama, nilai A dan B adalah 1 dan 9 sehingga $C = (1 + 9) / 2 = 5$.

Karena 5 terlalu besar dibandingkan dengan 3, maka Anda menjawab **TERLALU BESAR**. Sehingga untuk tebakan kedua, nilai A dan B adalah 1 dan 4. Karena nilai $C = (1 + 4) / 2 = 2,5$ bukan merupakan bilangan bulat, maka Pak Dengklek secara acak memilih untuk membulatkan ke bawah.

Karena 2 terlalu kecil dibandingkan dengan 3, maka Anda menjawab **TERLALU KECIL**. Sehingga untuk tebakan ketiga, nilai A dan B adalah 3 dan 4. Karena nilai $C = (3 + 4) / 2 = 3,5$ bukan merupakan bilangan bulat, maka Pak Dengklek secara acak memilih untuk membulatkan ke atas.

Karena 4 terlalu besar dibandingkan dengan 3, maka Anda menjawab **TERLALU BESAR**. Karena setelah $Q = 3$ tebakan Pak Dengklek tidak mendapatkan jawaban **SELAMAT**, maka Anda dinyatakan menang pada permainan ini.

Subsoal

Subsoal 1 (9 poin)

- $N = 31$
- $Q = 4$

Subsoal 2 (9 poin)

- $N = 127$
- $Q = 6$

Khusus untuk subsoal 1 dan subsoal 2:

- Hanya terdapat sebuah kasus uji (satu subsoal dinyatakan oleh satu kasus uji), yang dapat dimainkan [di sini](#).
- 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 (20 poin)

- $N = 4$
- $Q = 2$

Subsoal 4 (27 poin)

- $N = 9$
- $Q = 3$

Subsoal 5 (35 poin)

- $N = 16$
- $Q = 4$

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 memikirkan angka 42 tanpa memperdulikan nilai N dan Q yang diberikan.

```
var subsoal: string;
    N, Q, C, i: longint;

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

  for i := 1 to Q do begin
    readln(C);
    if (C < 42) then begin
      writeln('TERLALU KECIL');
      flush(output);
    end else if (C > 42) then begin
      writeln('TERLALU BESAR');
      flush(output);
    end else begin
      writeln('SELAMAT');
      flush(output);
    end;
  end;
end.
```

Dan berikut adalah contoh *source code* dalam bahasa C++.

```
#include <cstdio>
#include <cstring>

char subsoal[100];
int N, Q, C, i;

int main() {
  scanf("%s", subsoal);
  scanf("%d %d", &N, &Q);

  for (i = 1; i <= Q; i++) {
    scanf("%d", &C);
    if (C < 42) {
```

```
        printf("TERLALU KECIL\n");
        fflush(stdout);
    } else if (C > 42) {
        printf("TERLALU BESAR\n");
        fflush(stdout);
    } else {
        printf("SELAMAT\n");
        fflush(stdout);
    }

    return 0;
}
```

Peringatan

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

- melakukan tindakan tidak sesuai format sehingga tidak dikenali oleh *grader*,
- melakukan kecurangan dengan tidak konsistennya jawaban Anda, atau
- gagal memenangkan permainan,

maka program Anda akan dihentikan secara otomatis dan Anda mendapatkan *verdict Wrong Answer* pada kasus uji yang bersangkutan.

Tebak Pokémon

Batas Waktu	4 detik
Batas Memori	256 MB

Deskripsi



Pokédex adalah sebuah komputer saku yang mutlak dimiliki oleh pelatih Pokémon untuk mengidentifikasi spesies Pokémon yang ditemui. Dengan mengarahkan Pokédex menghadap Pokémon, pengguna akan mendapatkan data lengkap mengenai Pokémon tersebut (seperti morfologi, informasi evolusi, dan kekuatan khusus) hampir secara instan. Namun apakah Anda tahu bagaimana proses yang terjadi di dalam Pokédex sehingga bisa menampilkan data yang sesuai dengan sangat cepat?

Pada Pokédex tertanam sebuah kamera mini khusus yang berfungsi merekam ciri-ciri dari Pokémon yang hendak diidentifikasi. Ciri-ciri ini akan direkam ketika pengguna mengarahkan Pokédex menghadap Pokémon. Dalam sekali proses identifikasi, kamera mini mampu merekam sebanyak N ciri-ciri. Ciri-ciri tersebut kemudian diterjemahkan ke dalam bahasa manusia (dalam kasus ini bahasa Inggris), lalu dilakukan pencocokan dengan *database* di Pokémon Center agar didapat nama beserta data lengkap Pokémon yang diidentifikasi. Hal yang menarik adalah proses penerjemahan sangat canggih sedemikian sehingga bahasa manusia yang didapat selalu merupakan *substring* dari data lengkap Pokémon tersebut di *database*.

Dalam pengembangan Pokédex generasi berikutnya, Anda direkrut oleh Professor Dengklek, seorang peneliti Pokémon ternama yang juga merupakan rekan sejawat Professor Oak, untuk mengimplementasikan algoritma pencocokan antara ciri-ciri yang direkam kamera mini khusus dengan *database* di Pokémon Center.

Pada pengembangan iterasi pertama, Anda hanya perlu mencocokkan data dari Pokémon generasi pertama. Professor Dengklek tidak memedulikan algoritma yang Anda gunakan, asalkan hasilnya dapat sangat akurat. Dapatkah Anda menyelesaikan tugas Anda dengan baik?

Format Masukan

Pada awalnya, program Anda akan menerima label kasus uji. Label kasus uji berisi sebuah *string* yang dijelaskan sebagai berikut:

- Panjang *string* tersebut adalah banyaknya subsoal ditambah satu.
- Digit pertama dari label adalah karakter ke-0, digit kedua dari label adalah karakter ke-1, dan seterusnya.

- Karakter ke-0 akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, dan berisi '.' 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 '.'

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 sebuah bilangan bulat N yang merupakan banyaknya ciri-ciri Pokémon yang direkam oleh kamera mini khusus pada Pokédex generasi terbaru.

Berikutnya terdapat N baris. Pada tiap baris berisi *string* S yang menyatakan hasil terjemahan ciri-ciri Pokémon ke dalam bahasa manusia. Seperti yang disinggung di soal, **hasil terjemahan selalu merupakan substring dari data lengkap Pokémon tersebut di database.**

Sebagai referensi, *database* Pokémon yang lengkap dapat diunduh [di sini](#). Semua data yang terdapat dalam berkas tersebut diambil dari situs [Bulbapedia](#).

Format Keluaran

Untuk ciri-ciri Pokémon yang diberikan, cetak keluaran dalam 1 baris berisi nama Pokémon yang dimaksud dari hasil analisis ciri-ciri Pokémon tersebut.

Contoh Masukan

```
0.....6
6
The flame that burns at the tip of its tail is an indication of its
emotions
a bipedal, reptilian Pokemon with an orange body
A fire burns at the tip of this Pokemon's slender tail
one of three starter Pokemon of Kanto available at the beginning of Pokemon
Red, Green, Blue, FireRed, and LeafGreen
The flame at the tip of its tail makes a sound as it burns
Obviously prefers hot places
```

Contoh Keluaran

Charmander

Penjelasan Contoh Kasus Uji

Pada beberapa *browser*/peramban atau versi cetak, tampilan contoh masukan di atas mungkin akan terpecah menjadi beberapa baris untuk baris keempat karena terlalu panjang. Namun pada berkas masukan sebenarnya semua ciri-ciri terpisah berada di baris terpisah.

Ciri-ciri Pokémon yang diberikan diambil dari data lengkap Charmander. Bandingkan contoh masukan di atas dengan data lengkap Charmander (*substring* sebagai contoh masukan di-*highlight* kuning. Sama seperti penjelasan di contoh masukan, tampilan ciri-ciri di bawah mungkin terpecah menjadi beberapa baris).

Charmander 16

Charmander is a Fire-type Pokemon.

It evolves into Charmeleon starting at level 16, which evolves into Charizard starting at level 36.

Along with Bulbasaur and Squirtle, Charmander is one of three starter Pokemon of Kanto available at the beginning of Pokemon Red, Green, Blue, FireRed, and LeafGreen.

Charmander is a bipedal, reptilian Pokemon with an orange body, though its underside and soles are cream-colored. It has two small fangs visible in its upper and lower jaws and blue eyes. Its arms and legs are short with four fingers and three clawed toes. A fire burns at the tip of this Pokemon's slender tail, and has blazed there since Charmander's birth. The flame can be used as an indication of Charmander's health and mood, burning brightly when the Pokemon is strong, weakly when it is exhausted, wavering when it is happy, and blazing when it is enraged. It is said that Charmander dies if its flame goes out.

Charmander can be found in hot, mountainous areas. However, it is found far more often in the ownership of Trainers. As shown in Pokemon Snap, Charmander exhibits pack behavior, calling others of its species if it finds food.

Obviously prefers hot places. When it rains, steam is said to spout from the tip of its tail.

The flame at the tip of its tail makes a sound as it burns. You can only hear it in quiet places.

Even the newborns have flaming tails. Unfamiliar with fire, babies are said to accidentally burn themselves.

The flame on its tail shows the strength of its life force. If it is weak, the flame also burns weakly.

The flame on its tail indicates Charmander's life force. If it is healthy, the flame burns brightly.

If it's healthy, the flame on the tip of its tail will burn vigorously, even if it gets a bit wet.

The flame that burns at the tip of its tail is an indication of its emotions. The flame wavers when Charmander is enjoying itself. If the Pokemon becomes enraged, the flame burns fiercely.

The flame that burns at the tip of its tail is an indication of its emotions. The flame wavers when Charmander is happy, and blazes when it is enraged.

From the time it is born, a flame burns at the tip of its tail. Its life would end if the flame were to go out.

It has a preference for hot things. When it rains, steam is said to spout from the tip of its tail.

The fire on the tip of its tail is a measure of its life. If healthy, its tail burns intensely.

Berikut adalah gambar Charmander sebagai ilustrasi visual.

Panduan Membaca *Database* Lengkap Pokémon

Seperti yang disebutkan di bagian format masukan, *database* Pokémon yang lengkap dapat diunduh [di sini](#). Berkas ini dapat Anda gunakan sebagai panduan untuk menentukan nama Pokémon berdasarkan ciri-ciri yang diberikan. Semua data yang terdapat dalam berkas tersebut diambil dari situs [Bulbapedia](#).

Berikut adalah format dari *database* Pokémon. Baris pertama berisi sebuah bilangan bulat P yang menyatakan banyaknya Pokémon yang diberikan data lengkapnya. Berikutnya terdapat P set data yang dijelaskan sebagai berikut. Tiap set data diawali oleh nama Pokémon tersebut dan banyaknya data lengkap dari Pokémon tersebut yang diberikan dalam satu baris.

Berikutnya diikuti oleh data lengkap dari Pokémon tersebut, dipisah per baris. Sebagai ilustrasi berikut ada struktur dari berkas *database* Pokémon.

```
P
<nama pokémon ke-1><spasi><D1>
<data pokémon 1 ke-1>
<data pokémon 1 ke-2>
...
<data pokémon 1 ke-D1>
<nama pokémon ke-2><spasi><D2>
<data pokémon 2 ke-1>
<data pokémon 2 ke-2>
...
<data pokémon 2 ke-D2>
...
<nama pokémon ke-P><spasi><DP>
<data pokémon P ke-1>
<data pokémon P ke-2>
...
<data pokémon P ke-DP>
```

Pembagian Subsoal

Pada semua subsoal berlaku:

- $4 \leq N \leq 10$
- $1 \leq |S| \leq 150$
- Dijamin nama Pokémon tersebut tidak akan muncul pada ciri-ciri Pokémon yang diberikan.

Subsoal 1 (3 poin)

- Hanya terdapat sebuah kasus uji untuk subsoal ini, yang dapat diunduh [di sini](#).

Subsoal 2 (4 poin)

- Hanya terdapat sebuah kasus uji untuk subsoal ini, yang dapat diunduh [di sini](#).

Subsoal 3 (5 poin)

- Hanya terdapat sebuah kasus uji untuk subsoal ini, yang dapat diunduh [di sini](#).

Subsoal 4 (4 poin)

- Hanya terdapat sebuah kasus uji untuk subsoal ini, yang dapat diunduh [di sini](#).

Subsoal 5 (3 poin)

- Hanya terdapat sebuah kasus uji untuk subsoal ini, yang dapat diunduh [di sini](#).

Subsoal 6 (hingga 81 poin)

- Untuk subsoal ini, terdapat 45 kasus uji. Berbeda dengan model subsoal umum, pada subsoal ini semua kasus uji akan dinilai secara independen (Anda akan tetap mendapat nilai untuk kasus uji yang benar walaupun ada beberapa kasus uji yang salah).
- Bobot tiap kasus uji adalah 1,8 poin.

Catatan

Untuk menghindari terjadinya *error* pada eksekusi program, semua karakter "é" pada berkas masukan maupun berkas *database* Pokémon telah diganti menjadi karakter "e" biasa.

Semua Pokémon dengan karakter khusus pada namanya telah dihilangkan dari berkas *database* Pokémon sehingga tidak akan diujikan lewat kasus uji. Secara lengkap, berikut adalah daftar nama Pokémon yang dihilangkan walaupun berasal dari generasi pertama.

- Farfetch'd
- Nidoran♂
- Nidoran♀
- Mr. Mime

sehingga dijamin setiap nama Pokémon hanya terdiri dari alfabet huruf kapital dan huruf kecil.

Pokémon dan nama/gambar setiap karakter adalah merek dagang dan hak cipta dari pemiliknya masing-masing. Kami tidak berafiliasi dengan **Nintendo**, **Pokémon Company**, **Creatures Inc**, **Game Freak**, atau **Bulbapedia**. Tidak ada maksud untuk melanggar hak cipta atau merek dagang pelanggaran.

Stupa Borobudur

Batas Waktu	1 detik
Batas Memori	32 MB

Deskripsi

Pak Dengklek mengunjungi kawasan wisata Candi Borobudur. Kawasan ini menarik karena banyak terdapat stupa yang indah. Sang penjaga berkata bahwa ada keajaiban di balik bilangan bulat yang terjadi pada kawasan tersebut. Untuk setiap bilangan bulat positif X , terdapat tepat 0 atau X buah stupa setinggi X meter.

Pak Dengklek, yang tidak sempat berkeliling ke seluruh kawasan, hanya sempat melihat N stupa. Tinggi dari stupa ke- i yang dilihatnya adalah A_i meter. Pak Dengklek pun penasaran mengenai banyaknya stupa di kawasan tersebut.



Tentukan banyaknya stupa minimum di kawasan wisata tersebut!

Format Masukan

Pada awalnya, program Anda akan menerima label kasus uji. Label kasus uji berisi sebuah *string* yang dijelaskan sebagai berikut:

- Panjang *string* tersebut adalah banyaknya subsoal ditambah satu.
- Digit pertama dari label adalah karakter ke-0, digit kedua dari label adalah karakter ke-1, dan seterusnya.
- Karakter ke-0 akan berisi 0 jika kasus uji tersebut merupakan contoh kasus uji, dan berisi '.' 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 '.'

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 sebuah bilangan bulat N . Baris berikutnya berisi N buah bilangan bulat A_1, A_2, \dots, A_N .

Format Keluaran

Sebuah baris berisi banyaknya stupa minimum. Apabila masukan tidak sesuai dengan perkataan sang penjaga (mungkin saja Pak Dengklek salah lihat!), keluarkan sebuah baris berisi -1.

Contoh Masukan 1

```
0..34567
5
2 1 4 2 3
```

Contoh Keluaran 1

```
10
```

Contoh Masukan 2

```
0..34567
4
2 9 2 2
```

Contoh Keluaran 2

```
-1
```

Penjelasan

Untuk contoh masukan pertama, di kawasan tersebut terdapat:

- 1 buah stupa setinggi 1 meter
- 2 buah stupa setinggi 2 meter
- 3 buah stupa setinggi 3 meter
- 4 buah stupa setinggi 4 meter

sehingga terdapat total 10 buah stupa.

Untuk contoh masukan kedua, Pak Dengklek melihat 3 buah stupa setinggi 2 meter. Hal ini tidak sesuai dengan perkataan sang penjaga, oleh karena itu Pak Dengklek pasti salah lihat.

Subsoal

Subsoal 1 (7 poin)

- $N = 10$
- $A_i = i$
- Kasus uji dapat diunduh [di sini](#).

Subsoal 2 (11 poin)

- $N = 8$

- $1 \leq A_i \leq 10$
- Kasus uji dapat diunduh [di sini](#).

Subsoal 3 (7 poin)

- $N = 1$
- $1 \leq A_1 \leq 1.000$

Subsoal 4 (11 poin)

- $1 \leq N \leq 1.000$
- $1 \leq A_1 = A_2 = \dots = A_N \leq 1.000$

Subsoal 5 (17 poin)

- $1 \leq N \leq 1.000$
- $1 \leq A_i \leq 1.000$

Subsoal 6 (20 poin)

- $1 \leq N \leq 100.000$
- $1 \leq A_i \leq 100.000$

Subsoal 7 (27 poin)

- $1 \leq N \leq 100.000$
- $1 \leq A_i \leq 1.000.000.000$