

# Sang Pelompat

Batas Waktu	1 detik
Batas Memori	64 MB

## Deskripsi

Kwik adalah bebek Pak Dengklek yang senang menonton TV. Salah satu film favoritnya adalah serial “*The Indiana Duck*”. Serial ini mengisahkan seekor bebek yang bekerja sebagai arkeologi dan menemukan harta karun historis di seluruh dunia. Indiana Duck terkenal dengan atribut legendarisnya berupa topi koboi dan cambuk untuk membela diri dari serangan musuh.

Pada suatu hari, Kwik menemukan peta harta karun dalam kotak kecil di salah satu sudut gudang Pak Dengklek. Dalam peta itu tertulis jika seseorang mendaki turun melalui sumur tua di belakang gudang, maka ia akan menemui gua raksasa berukuran  $R \times C$  dengan lautan magma di bawahnya. Dari dasar lautan magma tersebut menyembul beberapa bongkahan batu keras yang dapat digunakan sebagai pijakan. Pada salah satu bongkahan batu terdapat harta karun yang sudah dijaga selama beberapa generasi keluarga Dengklek. Kwik sangat senang karena dia bisa berlagak meniru Indiana Duck tokoh idolanya.

Satu-satunya cara untuk berpindah dari suatu bongkahan batu ke bongkahan batu lainnya adalah dengan melompat (Kwik tidak boleh menyentuh lautan magma jika ingin kembali hidup-hidup). Selama berada dalam suatu bongkahan batu, Kwik dapat menjelajahi bongkahan batu tersebut tanpa perlu melompat (misalnya, untuk berpindah ke sisi lain kemudian baru melompat). 2 petak batu akan membentuk sebuah bongkahan batu besar jika kedua petak batu tersebut berbagi sisi.

Karena Kwik masih kecil, dia tidak dapat melakukan gerakan yang sulit. Kwik hanya bisa melompat untuk menyeberangi lautan magma secara garis lurus (tidak dapat berbelok di udara). Kwik juga hanya dapat melompat ke arah utara, timur, selatan, atau barat. Karena perjalanan panjang, Kwik akan menghemat tenaganya dan akan **selalu** mendarat di bongkahan batu terdekat yang ditemuinya di arah lompatan.

Kwik berencana mendapatkan harta karun yang disebutkan dalam peta Kwik pun menceritakan rencana hebatnya kepada Anda dan meminta agar Anda tidak melapor ke Pak Dengklek. Anda sangat khawatir dengan keselamatan Kwik, namun Anda juga tidak bisa menemani Kwik dalam petualangannya karena Anda tidak ingin ketinggalan acara “*Duck’s Got Talent*”. Karena itu Anda hendak membantu Kwik dengan menentukan berapa lompatan minimal yang harus dilakukan Kwik agar dapat sampai ke harta karun (tentunya, semakin sedikit melompat akan menghemat energi Kwik dan menjamin keselamatannya).

## 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, atau 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 dalam berkas masukan berisi 2 buah bilangan *R* dan *C* yang dipisahkan oleh tepat sebuah spasi. Baris berikutnya berisi *C* buah karakter yang menyatakan konfigurasi petak batu dan lokasi harta karun sesuai deskripsi dalam peta. Sebuah petak dalam gua raksasa tersebut hanya dapat berupa salah satu dari kemungkinan karakter di bawah ini.

1. '#' : petak tersebut merupakan petak batu yang bisa diinjak.
2. '.' : petak tersebut merupakan lautan magma yang tidak boleh diinjak.
3. 'S' : petak ini merupakan bagian akhir dari tangga pada sumur, atau dengan kata lain tempat Kwik memulai petualangannya. Petak ini pasti merupakan bagian dari petak batu yang bisa diinjak.
4. 'T' : petak ini tempat harta karun yang dimaksud. Petak ini pasti merupakan bagian dari petak batu yang diinjak.

## Format Keluaran

Cetak sebuah bilangan dalam 1 baris yang menyatakan berapa banyak lompatan **paling sedikit** yang harus dilakukan Kwik agar dapat sampai ke petak harta karun. Cetak -1 jika Kwik tidak akan pernah menemukan harta karun.

## Contoh Masukan

```
0 . . . . 56
7 7
# . . . . #T
# . . # . .
. . # # . . #
. . . # . . #
. # . # . . #
. # . . S . .
. # . . # . .
```

## Contoh Keluaran



- $1 \leq R, C \leq 100$
- Semua bongkahan batu terdiri dari tepat satu petak batu.

**Subsoal 4 (21 poin)**

- $1 \leq R, C \leq 100$
- Semua bongkahan batu berbentuk persegi panjang.

**Subsoal 5 (26 poin)**

- $1 \leq R, C \leq 100$

**Subsoal 6 (16 poin)**

- $1 \leq R, C \leq 1.000$

# Suten

Batas Waktu	1 detik
Batas Memori	32 MB

## Deskripsi

Tahukah anda apa itu suten? Suten, sut, suit (suwit), atau pingsut adalah cara mengundi untuk dua orang dengan cara mengadu jari untuk menentukan siapa yang menang.

Di Indonesia, jari yang dipergunakan dalam suten adalah ibu jari yang diumpamakan sebagai gajah, jari telunjuk yang diumpamakan sebagai manusia, dan jari kelingking yang diumpamakan sebagai semut. Dua orang yang bersuten secara serentak mengacungkan jari yang dipilihnya. Hasilnya seri terjadi bila kedua belah pihak yang bersuten mengacungkan jari yang berkekuatan sama, misalnya: kelingking melawan kelingking. Biasanya, apabila terjadi seri maka suten diulang hingga ada pihak yang menang, namun peraturan ini tidak berlaku untuk saat ini.

Jari yang menjadi pemenang suten:

- Ibu jari versus telunjuk: pemenang adalah ibu jari.
- Telunjuk versus kelingking: pemenang adalah telunjuk.
- Kelingking versus ibu jari: pemenang adalah kelingking.

Anda, sebagai salah satu finalis OSN Komputer 2014, diminta untuk menyelesaikan tantangan berikut.

Ada  $N$  anak yang bermain suten bersama-sama. Anak-anak ini memiliki jari favorit mereka masing-masing sehingga setiap melakukan suten, mereka selalu mengeluarkan jari favoritnya untuk diadu. Setiap anak melakukan suten dengan anak lainnya tepat sekali, sehingga ada  $(N \times (N - 1)) / 2$  buah pertandingan suten yang akan dilakukan.

Diberikan nilai  $N$ , Anda (dan program Anda) akan ditanyakan  $(N \times (N - 1)) / 2$  buah pertandingan suten dan menentukan siapa pemenangnya (atau menentukan adanya seri). Apabila Anda salah menjawab, maka secara otomatis Anda gagal menyelesaikan tantangan dan akan mendapatkan nilai 0.

Karena Anda tidak mengetahui jari favorit masing-masing anak, Anda diperbolehkan untuk memilih PASS atau memilih untuk tidak menjawab sebuah pertandingan. Apabila Anda memilih untuk PASS, maka anda akan diberikan hasil pertandingan tersebut. Anda diminta untuk memilih PASS **sesedikit mungkin**, dengan kata lain Anda diminta untuk menjawab setiap hasil pertandingan sebanyak mungkin.

## 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 . . 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 input sebuah bilangan  $N$  yaitu banyaknya anak yang bermain suten. Kemudian, sebanyak  $(N \times (N - 1)) / 2$  kali program Anda akan menerima input 2 buah bilangan  $X$  dan  $Y$  yang menandakan pertandingan suten antara anak ke- $X$  dan anak ke- $Y$ . Anda diminta untuk mencari tahu hasil pertandingan suten tersebut satu per satu.

Keluaran yang dapat Anda keluarkan adalah:

- Apabila pertandingan suten dimenangkan oleh anak ke- $Z$ , program Anda diminta untuk mengeluarkan output  $Z$  MENANG.
- Apabila pertandingan suten berlangsung seri, program Anda diminta untuk mengeluarkan output SERI.
- Apabila Anda memilih untuk PASS atau tidak menjawab, program Anda diminta untuk mengeluarkan output PASS. Apabila memilih PASS, maka program Anda akan menerima hasil pertandingan suten tersebut dalam bentuk  $Z$  MENANG ataupun SERI.

### Contoh Interaksi

Keluaran Program Peserta	Umpan Balik Grader
	0 . . . 567
	4
	1 2
SERI	

Keluaran Program Peserta	Umpan Balik <i>Grader</i>
	1 3
3 MENANG	
	1 4
1 MENANG	
	2 3
PASS	
	3 MENANG
	2 4
PASS	
	2 MENANG
	3 4
4 MENANG	
	(interaksi selesai)

### Penjelasan Contoh Interaksi

Pada kasus tersebut terdapat 4 anak yang bermain suten, sehingga terdapat 6 buah pertandingan. Pada kasus tersebut pula, jari favorit masing-masing anak adalah sebagai berikut:

- Anak ke-1 : Ibu Jari.
- Anak ke-2 : Ibu Jari.
- Anak ke-3 : Kelingking.
- Anak ke-4 : Telunjuk.

Karena tidak ada jawaban interaksi yang salah maka interaksi yang terjadi merupakan interaksi yang valid dan program tersebut berhasil menyelesaikan tantangan dengan melakukan 2 kali PASS.

### Subsoal

Pada semua subsoal, berlaku:

- $3 \leq N \leq 100$
- Dimisalkan P adalah banyaknya PASS paling banyak yang boleh dilakukan peserta.

#### Subsoal 1 (6 poin)

- $N = 5$
- $P = 6$
- Untuk setiap pertandingan, berlaku  $X < Y$ .
- Input pertandingan akan diurutkan dengan aturan berikut:
  - Pertandingan diurut mula-mula dari nilai X terkecil.
  - Apabila 2 buah pertandingan memiliki nilai X yang sama, maka diurut mula-mula dari nilai Y terkecil.

#### Subsoal 2 (12 poin)

- $N = 8$
- $P = 7$

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 (8 poin)

- $N = 3$
- $P = 2$

#### Subsoal 4 (13 poin)

- $P = (N \times (N - 1)) / 2 - N + 2$

#### Subsoal 5 (14 poin)

- $P = N - 1$
- Untuk setiap pertandingan, berlaku  $X < Y$
- Input pertandingan akan diurutkan dengan aturan berikut:
  - Pertandingan diurut mula-mula dari nilai X terkecil.
  - Apabila 2 buah pertandingan memiliki nilai X yang sama, maka diurut mula-mula dari nilai Y terkecil.

#### Subsoal 6 (19 poin)



- $P = N - 1$
- Untuk  $N - 1$  pertandingan pertama, berlaku  $X < Y$ .
- Untuk  $N - 1$  pertandingan pertama, pada pertandingan ke- $i$  nilai  $Y = i + 1$ .

**Subsoal 7 (28 poin)**

- $P = N - 1$

**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 menjawab PASS tanpa mempedulikan nilai  $N$  yang diberikan.

```
var subsoal, jawaban: string;
    N, X, Y, i: longint;

begin
    readln(subsoal);
    readln(N);

    for i := 1 to (N*(N-1) div 2) do begin
        readln(X,Y);

        writeln('PASS');
        flush(output);

        readln(jawaban);
    end;
end.
```

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

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

char subsoal[100], jawaban[100];
int N, X, Y, i;

int main() {
    gets(subsoal);
    scanf("%d\n", N);

    for (i = 1; i <= N*(N-1)/2; i++) {
        scanf("%d %d\n", X, Y);

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

        gets(jawaban);
    }
}
```

```
    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*,
- salah menjawab di suatu hasil pertandingan, atau
- selesai menjawab semua hasil pertandingan namun memilih PASS lebih dari P kali,

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

# Hiasan Dinding

Batas Waktu	1 detik
Batas Memori	64 MB

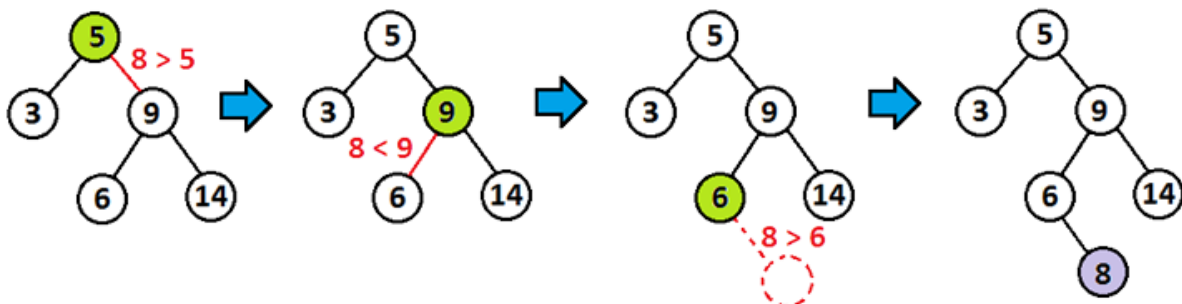
## Deskripsi

Untuk menyambut OSN 2014, Pak Dengklek berencana membuat sebuah hiasan dinding di laboratorium tempat Bidang Informatika dilaksanakan. Hiasan ini cukup sederhana, yakni hanya terdiri atas paku-paku dan untaian tali. Pak Dengklek memiliki  $N$  buah paku untuk membuat hiasan ini. Paku-paku ini dinomori secara unik dari 1 sampai dengan  $N$ .

Cara membuat hiasan dinding ini adalah sebagai berikut. Mula-mula, Pak Dengklek membariskan paku-paku tersebut dalam sebuah barisan. Lalu, Pak Dengklek memasang paku pertama pada dinding. Untuk setiap paku kedua sampai dengan ke- $N$  secara berurutan, Pak Dengklek melakukan langkah-langkah berikut:

1. Tinjau paku paling atas pada dinding.
2. Misalkan  $X$  = paku yang sedang ditinjau, dan  $Y$  = paku yang ingin dipasang.
  - a) Jika nomor dari  $Y <$  nomor dari  $X$ :
    - i. Jika  $X$  tidak memiliki paku yang terhubung dengan tali di sebelah kiri bawahnya, pasang  $Y$  persis disebelah kiri bawah  $X$ , dan hubungkan keduanya menggunakan tali. Pemasangan paku dianggap selesai.
    - ii. Jika  $X$  memiliki paku yang terhubung dengan tali di sebelah kiri bawahnya, tinjau paku tersebut dan kembali ke Langkah 2.
  - b) Jika nomor dari  $Y >$  nomor dari  $X$ :
    - i. (i) Jika  $X$  tidak memiliki paku yang terhubung dengan tali di sebelah kanan bawahnya, pasang  $Y$  persis di sebelah kanan bawah  $X$ , dan hubungkan keduanya menggunakan tali. Pemasangan paku dianggap selesai.
    - ii. Jika  $X$  memiliki paku yang terhubung dengan tali di sebelah kanan bawahnya, tinjau paku tersebut dan kembali ke Langkah 2.

Sebagai contoh, misalkan Pak Dengklek ingin memasang paku keenam bernomor 8. Berikut adalah ilustrasi langkah-langkah pemasangan paku ini:



Pak Dengklek sudah memiliki sebuah desain hiasan dinding yang dibuat oleh panitia dekorasi. Ia ingin segera memasang paku-paku agar terbentuk hiasan dinding tersebut. Tentu saja, mula-mula ia harus membariskan paku-paku tersebut. Mungkin saja ada lebih dari satu kemungkinan barisan yang menghasilkan hiasan dinding tersebut. Pak Dengklek, sebagai seorang yang senang dengan teka-teki logika, justru penasaran dengan pertanyaan berikut: jika barisan-barisan yang mungkin menghasilkan hiasan dinding tersebut diurutkan dengan urutan leksikografis, barisan apa yang berada pada posisi ke-K?

Catatan: barisan paku A lebih kecil daripada barisan paku B secara leksikografis, apabila pada posisi pertama di mana A dan B berbeda, nomor paku pada posisi tersebut di A lebih kecil daripada nomor paku pada posisi tersebut di B.

Bantulah Pak Dengklek menjawab pertanyaan tersebut, agar ia dapat segera memasang paku-paku 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 dua buah bilangan bulat  $N$  dan  $K$ .  $N-1$  baris berikutnya mendeskripsikan sebuah desain hiasan dinding. Masing-masing baris berisi tiga buah bilangan bulat  $u$ ,  $v$ , dan  $t$ :

- Apabila  $t = 0$ , maka paku bernomor  $v$  terhubung dengan tali di sebelah kiri bawah dengan paku bernomor  $u$ .
- Apabila  $t = 1$ , maka paku bernomor  $v$  terhubung dengan tali di sebelah kanan bawah dengan paku bernomor  $u$ .

## Format Keluaran

Sebuah baris berisi barisan  $N$  buah bilangan, dipisahkan oleh spasi, yang menyatakan barisan nomor paku, sedemikian sehingga barisan ini merupakan barisan terkecil ke- $K$  secara leksikografis.

### Contoh Masukan 1

```
0..34.6
4 1
2 1 0
2 4 1
4 3 0
```

### Contoh Keluaran 1

```
2 1 4 3
```

### Contoh Masukan 2

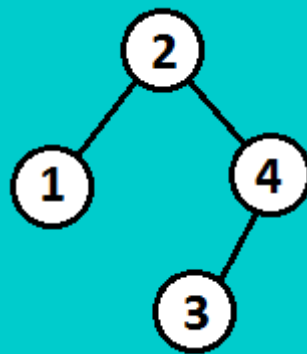
```
0..3..6
4 3
2 1 0
2 4 1
4 3 0
```

### Contoh Keluaran 2

```
2 4 3 1
```

### Penjelasan

Kedua contoh masukan mendeskripsikan desain hiasan dinding yang sama, yakni:



Terdapat 3 buah barisan nomor paku yang menghasilkan pohon tersebut, yakni:

1. 2 1 4 3
2. 2 4 1 3
3. 2 4 3 1

### Subsoal

Pada semua subsoal, berlaku:

- Setiap paku memiliki nomor unik antara 1 sampai dengan N.
- Masukan selalu merupakan desain hiasan dinding yang benar.

- K tidak lebih daripada banyaknya cara penyusunan hiasan dinding yang bersangkutan.
- Meskipun cara penyusunan hiasan dinding bisa banyak sekali, nilai K tidak akan lebih daripada 2.000.000.000.

**Subsoal 1 (7 poin)**

- $N = 8$
- $K = 1$
- Kasus uji dapat diunduh [di sini](#).

**Subsoal 2 (14 poin)**

- $N = 8$
- $K = 3$
- Kasus uji dapat diunduh [di sini](#).

**Subsoal 3 (17 poin)**

- $1 \leq N \leq 8$

**Subsoal 4 (11 poin)**

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

**Subsoal 5 (20 poin)**

- $1 \leq N \leq 100$
- Semua paku dengan nomor lebih kecil daripada nomor paku teratas tidak memiliki paku yang terhubung dengan tali di sebelah kanan bawah.
- Semua paku dengan nomor lebih besar daripada nomor paku teratas tidak memiliki paku yang terhubung dengan tali di sebelah kiri bawah.

**Subsoal 6 (31 poin)**

- $1 \leq N \leq 100$

# Komunikasi Bebek

Batas Waktu	0,4 detik
Batas Memori	256 MB

## Deskripsi

Pak Dengklek memiliki sebuah kebun yang panjangnya tak hingga. Kebun Pak Dengklek dapat direpresentasikan dalam garis sumbu  $x$  pada koordinat Kartesian. Pak Dengklek memiliki  $N$  ekor bebek pada kebun tersebut. Bebek ke- $i$  terletak pada koordinat  $x_i$  pada sumbu  $x$ , dengan  $x_i$  adalah bilangan bulat. Dijamin tidak ada dua bebek pada posisi yang sama.

Sepasang bebek dapat berkomunikasi jika dan hanya jika tidak ada bebek lain yang berada di antara dua bebek tersebut. Karena  $N$  bebek semuanya berada pada satu garis, tentu saja ada  $N - 1$  pasang bebek yang dapat berkomunikasi. Jarak dari sebuah komunikasi adalah jarak dua bebek yang sedang melakukan komunikasi tersebut.

Sebagai contoh, jika pada awalnya ada 4 bebek yang berada pada posisi  $\{1, 2, 5, 6\}$ , maka jarak komunikasi bebek 1 dan 2 adalah 1, jarak komunikasi bebek 2 dan 3 adalah 3, dan jarak komunikasi bebek 3 dan 4 adalah 1.

Pada suatu saat mungkin saja terdapat jarak komunikasi yang berbeda-beda. Pak Dengklek merasa komunikasi bebek berjalan tidak lancar jika terdapat lebih dari satu nilai jarak komunikasi. Untuk itu, Pak Dengklek mungkin harus mengganti konfigurasi kebunnya. Bebek-bebek Pak Dengklek sudah terlalu nyaman dengan posisinya, sehingga mereka tidak mau dipindah posisinya. Sehingga, yang bisa dilakukan oleh Pak Dengklek adalah menambahkan bebek baru ke kebun atau mengambil bebek lama keluar dari kebun.

Pak Dengklek akan mengambil  $X$  bebek dan meletakkan  $Y$  bebek, sehingga pada akhirnya terdapat  $(N - X + Y)$  bebek, dan posisi bebek-bebek tersebut harus berbeda-beda dan bebek ke- $i$  terletak pada  $x_i'$  dimana  $x_i'$  adalah bilangan bulat, dan semua jarak komunikasi untuk  $(N - X + Y - 1)$  pasangan bebek bernilai sama. Tentukan nilai  $X + Y$  minimal yang dapat dilakukan oleh Pak Dengklek.

## 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, atau 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 menunjukkan banyaknya bebek mula-mula.

Baris ketiga berisi N bilangan bulat yang dipisahkan spasi. Bilangan ke-i menunjukkan nilai dari  $x_i$ .

### Format Keluaran

Sebuah bilangan bulat yang menunjukkan nilai  $X + Y$  minimal agar konfigurasi kebun Pak Dengklek memenuhi ketentuan di atas.

### Contoh Masukan

```
0..3456
5
2 4 6 10 18
```

### Contoh Keluaran

```
2
```

### Penjelasan

Pak Dengklek akan meletakkan bebek pada posisi  $x = 14$  dan mengambil bebek pada posisi  $x = 4$ , sehingga pada akhirnya akan ada bebek-bebek pada posisi  $\{2, 6, 10, 14, 18\}$ , dimana setiap komunikasi bebek berjarak 4. Solusi ini membutuhkan 1 pengambilan bebek dan 1 penambahan bebek. Tidak ada solusi yang lebih optimal dari solusi ini.

### Subsoal

Pada semua subsoal, berlaku:

- $-2.500 \leq x_i \leq 2.500$

### Subsoal 1 (6 poin)

- $N = 6$
- Kasus uji dapat diunduh [di sini](#).

### Subsoal 2 (14 poin)



- $N = 9$
- Kasus uji dapat diunduh [di sini](#).

**Subsoal 3 (19 poin)**

- $1 \leq N \leq 15$

**Subsoal 4 (21 poin)**

- $1 \leq N \leq 50$

**Subsoal 5 (13 poin)**

- $1 \leq N \leq 300$

**Subsoal 6 (27 poin)**

- $1 \leq N \leq 2.000$