Concurso de Programación
AdaByron 2016

`http://www.ada-byron.es`

Cuadernillo de problemas

Patrocinado por

CORITEL
An Accenture Company

Facultad de Informática

UNIVERSIDAD COMPLUTENSE MADRID

Realizado en la **Facultad de Informática (UCM)**
26-27 de febrero de 2016

Universidad Complutense Madrid

In almost every computation a great variety of arrangements for the succession of the processes is possible, and various considerations must influence the selections amongst them for the purposes of a calculating engine. One essential object is to choose that arrangement which shall tend to reduce to a minimum the time necessary for completing the calculation.

Ada Byron

26-27 de febrero de 2016

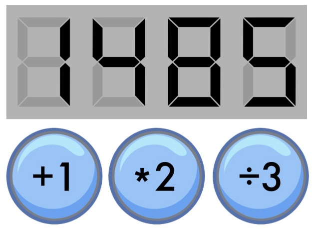
Listado de problemas

Autores de los problemas (Universidad Complutense de Madrid):

- Marco Antonio Gómez Martín
- Pedro Pablo Gómez Martín
- Enrique Martín Martín
- Isabel Pita Andreu
- Antonio Sánchez Ruiz-Granados
- Clara Segura Díaz
- Alberto Verdejo López

A

La máquina calculadora



A Javier le gusta la electrónica y cacharrear para construir máquinas que tengan cierto propósito. Ahora que su hijo Luis está aprendiendo a calcular le ha construido una máquina con un marcador, en el que aparecen cuatro dígitos y tres botones marcados con etiquetas `+1`, `*2` y `÷3`, que al ser pulsados actualizan el marcador realizando la operación correspondiente (sumar uno, multiplicar por dos o dividir entre tres). Como el marcador solamente tiene cuatro dígitos, las operaciones se realizan módulo 10.000 y la división es entera.

Luis ha entendido perfectamente el funcionamiento de la máquina y la utiliza para comprobar que los cálculos que hace mentalmente antes de pulsar un botón son correctos. Ahora Javier le ha retado con un juego: él configura el marcador para que aparezca un número concreto y le pide a Luis que consiga llegar a otro número pulsando los botones el menor número de veces.

¿Puedes ayudarles calculando cuál es el menor número de pulsaciones que hay que realizar para conseguir que aparezca el número final a partir del original?

Entrada

El programa dará respuesta a una serie de casos de prueba. Cada caso consiste en una única línea con dos números (entre 0 y 9.999), el que aparece originalmente en el marcador y el que Luis debe conseguir pulsando los botones de la máquina calculadora.

Salida

Para cada caso de prueba, se escribirá en una línea el menor número de pulsaciones necesarias para conseguir el número final a partir del original.

Entrada de ejemplo

```
0 1024
5000 0
9999 6666
```

Salida de ejemplo

```
11
1
2
```

B

Palmeras en la nieve

El alcalde de Marbella ha recibido una noticia climatológica algo preocupante. El día 28 de febrero se espera una tremenda nevada al nivel del mar. Los jardineros del ayuntamiento le han advertido de que hay ciertas palmeras del paseo marítimo que podrían vencerse bajo el peso de la nieve, y está preocupado por el deterioro que puede sufrir el aspecto del paseo si quedan demasiados huecos por esas caídas.

El paseo es uno de los lugares más populares de la ciudad, donde gran cantidad de gente acude a los chiringuitos en verano. Los cientos de palmeras que allí crecen proporcionan sombra y frescor. El servicio de jardinería ha hecho un gran esfuerzo para que no falte ni una sola palmera en todo el paseo. Si una palmera cae no dará tiempo a que crezca para el verano y no dará sombra, por lo que en aquellas zonas con pocas palmeras no podrán colocarse chiringuitos.

El servicio meteorológico ha hecho una estimación del peso de la nieve que caerá sobre las palmeras. Por su parte, los jardineros han determinado el peso máximo que cada una de ellas será capaz de soportar sin derrumbarse. Con esta información, el alcalde quiere saber qué parte del paseo será la más afectada; en particular, cuál será la franja más larga en la que, tras la nevada, quedarán un máximo de 5 palmeras en pie.

Entrada

La primera línea contiene un número que indica el número de casos de prueba que aparecen a continuación. Cada caso de prueba se compone de los siguientes datos: en la primera línea el peso en kilogramos de la nieve que el servicio meteorológico ha calculado que puede caer sobre cada palmera (un número natural); en la siguiente línea el número de palmeras del paseo (mayor o igual que 1 y menor o igual a 100.000) y la secuencia de pesos en kilogramos (números naturales) que pueden soportar las palmeras del paseo marítimo, tal como se ven de izquierda a derecha desde el interior.

Salida

Para cada caso de prueba se pide escribir en una línea la longitud de la franja más afectada, es decir la más larga que conserve como mucho 5 palmeras en pie.

Entrada de ejemplo

```
3
30
10
10 30 50 20 40 60 30 40 50 36
40
10
10 30 50 20 40 20 10 10 20 36
20
3
40 10 15
```

Salida de ejemplo

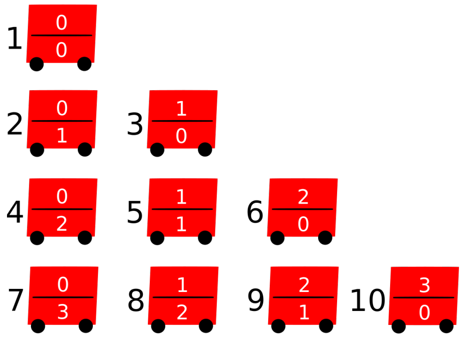
```
7
10
3
```

C

Double decker

La empresa Turistas-A-Cholón (TAC) está especializada en autobuses de dos pisos para recorridos turísticos. Raimundo Revisor tiene una aburrida tarea: debe revisar todos los autobuses que salen de la parada inicial a lo largo del día y comunicar a la central cuántos turistas hay en el piso superior y cuántos en el piso inferior.

Sin embargo, Raimundo Revisor está cansado de estos repetitivos mensajes y ha ideado un método más compacto para describir los turistas que hay en cada piso: en lugar de transmitir las dos cantidades transmite un único número (llamado *rango*) que resume perfectamente la ocupación del autobús. Para ello sigue el siguiente esquema (donde el rango de cada autobús aparece a su lado), distribuyendo en filas los autobuses con el mismo número de turistas totales.



Raimundo no tiene aún mucha pericia, y necesita un programa para calcular el rango de un autobús en base al número de turistas que transporta en cada piso.

Entrada

La primera línea contiene el número A de autobuses para calcular su rango. Luego siguen A líneas con el formato N M, donde N es el número de turistas que viajan en el piso superior y M el número de turistas en el piso inferior. La máxima ocupación de cada piso es 1.000.

Salida

Para cada caso de prueba, el programa escribirá el rango del autobús.

Entrada de ejemplo

```
3
0 0
1 2
2 0
```

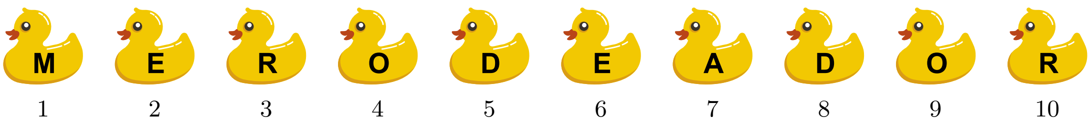
Salida de ejemplo

```
1
8
6
```

D

Tiro al patíndromo

Ya es tradición que cuando en el pueblo de Juan Filloy[1] están en fiestas, él se pase por el puesto del tiro con escopeta. Cuando saben que va a visitarles, en vez de premios en los patos a los que hay que disparar colocan letras de tal forma que Juan pueda divertirse disparando a algunos de los patos de manera que se forme un palíndromo con las letras en los patos supervivientes (es decir, que al leer tanto de izquierda a derecha como de derecha a izquierda, se lea siempre la misma palabra). Por ejemplo, si se enfrentara a la serie de patos de la figura, Juan dispararía a los patos en las posiciones 1, 2 y 6, formando el palíndromo `RODADOR`.



Además, a Juan le gusta alardear ante las mozas del pueblo que le miran expectantes, por lo que no se conforma solamente con encontrar un palíndromo sino que lo intenta conseguir tirando el menor número de patos. En caso de empate, Juan prefiere tirar los patos de la izquierda, porque a la derecha le molesta más el Sol que le deslumbra. ¿Sabrías tú hacer lo mismo?

Entrada

La entrada estará compuesta por múltiples casos de prueba, cada uno en una línea. Cada caso consiste en una sucesión de un mínimo de 1 y un máximo de 1.000 letras mayúsculas del alfabeto inglés, sin símbolos especiales ni espacios.

Salida

Por cada caso de prueba se escribirá una línea que contenga el palíndromo más largo que puede formarse eliminando (si es necesario) algunas de las letras de la entrada. En caso de empate, se debe imitar a Juan, eliminando letras más a la izquierda.

Entrada de ejemplo

```
MERODEADOR
RECONOCER
ARADAROSOSOMI
OSORASODAR
```

Salida de ejemplo

```
RODADOR
RECONOCER
OSOSO
RADAR
```

[1] Escritor argentino nacido en 1894, autoproclamado "*recordman mundial de palindromía*", gracias al cual hoy conocemos más de 8.000 palíndromos en español.

E

El conteo de la rosa

Estando ya al final de mi vida, esperando que mi enfermo y pesado cuerpo libere mi alma, vienen a mi memoria lejanos recuerdos que aún conservo pese a que, al mismo tiempo, no me sea posible recordar lo que desayuné tras mis rezos de Maitines.

Siendo apenas un novicio, desconocedor de la férrea disciplina del monasterio, me enfrenté con el abad, cuyo nombre prefiero ahora cubrir con un piadoso manto de silencio, por lo que mi joven entendimiento consideró una afrenta del bibliotecario mayor. Éste nos encargó a Adso, otro novicio que a la par llegaría a ser sabio franciscano, y a mí que numeráramos las 200 páginas de una nueva edición de la *Poética* de *Aristóteles* que varios monjes habían estado copiando durante meses atrás. El bibliotecario le asignó a Adso la numeración de las 100 primeras páginas del manuscrito, de la 1 a la 100, mientras que a mí me fueron asignadas las 100 siguientes, de la 101 a la 200. Yo, que había oído preocupantes rumores sobre una maldición que mataba a todo aquél que se acercaba a ese libro, caí en la cuenta de que me vería obligado a transcribir muchos más *dígitos*, que no páginas, que mi compañero de celda, razón que me llevó a mi enfrentamiento con el abad.

Éste, que consideró una mera lujuria del conocimiento que me hubiera planteado ni siquiera semejante hecho, me forzó a rezar en Laudes, Tercia y Vísperas durante todo un año el Salmo 30 para pedir protección contra las injusticias. Aun así, algo debió ver en mis ojos brillantes de muchacho, que concedió darme la bula de su castigo si le decía hasta qué página debía numerar Adso, y a partir de cuál debía numerar yo para que el reparto fuera justo, de forma que si Adso numerara una página más ya tendría que escribir más dígitos que yo.

En el pecado llevé mi penitencia, porque me tembló el entendimiento y ni siquiera hoy he podido descubrir la respuesta.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso son dos números, el primero indica la página inicial a numerar y el segundo la página final. Se garantiza que $1 \leq p_1 < p_2 \leq 1.000.000$. La entrada termina con dos valores a cero.

Salida

Para cada caso de prueba se indicará la última página que debería escribir Adso, el primer monje, de tal forma que ambos escriban la misma cantidad de dígitos. Si esto resulta imposible, Adso deberá numerar la mayor cantidad posible de páginas siempre que escriba menos *dígitos* que el segundo monje.

Entrada de ejemplo

```
1 200
99 100
99 101
97 103
0 0
```

Salida de ejemplo

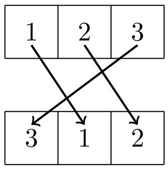
```
118
99
99
100
```

F

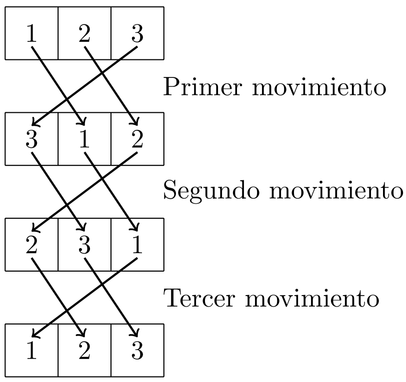
El teorema del punto fijo

En la familia de teoremas matemáticos llamados *del punto fijo*, Brouwer estudió uno que, parece ser, explicaba con la analogía de remover una taza de café con azúcar. Brouwer afirmaba que en todo momento hay un punto/molécula de la taza que no habrá cambiado de lugar.

Nosotros no estudiaremos ese teorema, pero nos quedamos con la idea de que, al mover la taza de café, sus moléculas intercambian sus lugares para terminar en sitios completamente distintos. Por poner un ejemplo reducido (en el que ni siquiera se cumple el teorema del punto fijo), la molécula 1 podría haber pasado a ocupar el lugar que tenía la molécula 2, la molécula 2 pasar donde estaba la molécula 3, y la molécula 3 ocupar el lugar que originalmente tenía la 1.



Si somos capaces de replicar el mismo movimiento una y otra vez, llega un momento en el que la taza de café vuelve a su estado original:



La pregunta que nos hacemos es, dada la descripción del intercambio de moléculas que conseguimos con el movimiento de la taza de café, ¿cuántas veces tendremos que repetirlo para que el estado de la taza vuelva a ser el mismo que al principio?

Entrada

La entrada estará compuesta por distintos casos de prueba. Cada uno de ellos consta de dos líneas, una primera con el número de moléculas en la taza de café ($1 \leq n \leq 100$) y otra indicando qué movimiento de moléculas se realizan cada vez que se mueve la taza.

El movimiento viene definido con n enteros que indican, para cada posición de una molécula, en qué posición termina la molécula que ocupaba ese lugar.

La entrada termina con una taza de café sin moléculas, que no debe procesarse.

Salida

Para cada caso de prueba se escribirá el número de veces que hay que mover la taza de café para que cada molécula recupere su lugar original. Se garantiza que el número no excederá 10^9.

Entrada de ejemplo

```
3
1 2 3
3
2 3 1
4
2 1 4 3
0
```

Salida de ejemplo

```
1
3
2
```

G
Helados de cucurucho

Alba y Blanca tienen opiniones enfrentadas respecto a los cucuruchos de helado, y eso que las dos comparten a la vainilla y al chocolate como sus sabores preferidos. Cuando sus padres tienen un buen día y deciden invitarlas a un helado de dos bolas, siempre surge entre ellas la misma discusión. Al colocarla, el tendero aprieta la primera bola hacia dentro del cucurucho para poder colocar la segunda, lo que significa que el segundo sabor que coloca es irremediablemente el primero que se come.

Alba prefiere poner el chocolate arriba. Es su sabor preferido, y quiere comérselo el primero, porque cuando llega a la vanilla tiene la lengua demasiado fría y apenas percibe el sabor con claridad. A Blanca también le gusta más el chocolate que la vanilla, pero tiene una teoría diferente. Es más seguro poner el chocolate abajo, porque la bola de arriba es la que más tendencia tiene a terminar en el suelo si se comen el helado mientras caminan.

Sus padres no sirven para desempatar, porque ellos... se compran los helados de tres bolas, y también cada uno las pide en un orden diferente. De hecho, cuantas más bolas tiene un helado, más formas distintas hay de colocarlas... Así no hay quién se ponga de acuerdo en nada.

Entrada

La entrada comienza por una línea indicando el número de casos de prueba que deberán procesarse. Cada caso de prueba es una pareja de números indicando el número de bolas de helado de chocolate y de vainilla que se usarán para un cucurucho. No habrá helados sin bolas; además hay algunos cucuruchos enormes que pueden llegar a soportar hasta 15 bolas.

Salida

Para cada caso de prueba se escribirán todas las formas posibles de colocar las bolas de helado. Cada configuración de un helado se escribirá como una sucesión de letras `C` y `V` para chocolate y vainilla, respectivamente. Las diferentes configuraciones se escribirán por orden alfabético, separadas por un espacio. No se pondrá espacio tras la última.

Entrada de ejemplo

```
2
1 1
2 1
```

Salida de ejemplo

```
CV VC
CCV CVC VCC
```

H

La ardilla viajera

La leyenda popular dice que en el libro *Geografía* escrito en el siglo I A.C., el geógrafo romano Estrabón dijo de la península ibérica que era tan frondosa que una ardilla podía cruzarla de sur a norte saltando de árbol en árbol sin tocar el suelo.

Parece ser que, en realidad, el bueno de Estrabón nunca aseguró tal cosa en su libro y, de hecho, hoy en día se cree que ni siquiera en aquellos tiempos la azaña de la ardilla pudo ser cierta.

No obstante dejemos volar la imaginación por un momento y pensemos que en algún instante del pasado el número de árboles en el territorio de la península permitiera la aventura. Dado que hoy día no es posible realizarla, está claro que en algún momento del pasado se cortó el árbol que provocó la separación de la región del norte y la del sur en lo que a saltos de rama en rama se refiere.

Para simplificar un poco el problema, asumamos que el territorio es una región cuadrada de tamaño $N \times M$ en la que se sitúan árboles (que consideraremos de grosor 0) en posiciones (x, y). La tarea de la ardilla consiste en ir desde el árbol situado en la posición $(0, 0)$ hasta la posición (N, M) (en ambas posiciones hay siempre un árbol). La ardilla es capaz de saltar de un árbol a otro si la distancia entre ambos no supera K unidades.

Los datos que tenemos del territorio son las posiciones de todos los árboles al principio de los tiempos y el orden en el que fueron cortados y debemos determinar la posición del árbol que, al ser cortado, hizo que la ardilla no pudiera atravesar la península de parte a parte sin pisar el suelo.

Entrada

Cada caso de prueba se compone de varias líneas. La primera de ellas tiene los números N, M, K y n ($1 \leq N, M \leq 1.000$, $1 \leq K \leq 10$, $1 \leq n \leq 100.000$), donde los tres primeros tienen el significado descrito más arriba y el último indica el número de árboles en el territorio (sin contar los situados en la esquina origen y destino de la ardilla que siempre están presentes).

A continuación aparece una línea por cada árbol con dos enteros x, y con la posición de cada uno. El orden en el que se dan es el mismo en el que después fueron cortados. Se garantiza que todas las posiciones están dentro del territorio y que no hay dos árboles en la misma ubicación.

Salida

Por cada caso de prueba se escribirá una única línea con la posición dentro del territorio del árbol que, cuando se cortó, provocó que las dos esquinas del territorio no fueran alcanzables para la ardilla.

Si la ardilla nunco pudo atravesar la península de parte a parte, se escribirá `NUNCA SE PUDO`.

Entrada de ejemplo

```
3 3 2 4
1 1
2 2
2 0
3 1
3 3 2 2
3 0
0 3
```

Salida de ejemplo

```
2 0
NUNCA SE PUDO
```

El acertijo del mercader

Un grupo de peregrinos, reunidos camino a Santiago, decidieron proponerse acertijos entre sí para hacer más amena la marcha. Entre ellos iba un mercader, hombre pensativo, organizado y que manejaba los números con solvencia.

Cuando le llegó su turno, les hizo ver que, en total, formaban un grupo de 12 caminantes y, por tanto, podían ir andando en fila india, o agrupados de dos en dos, de tres en tres, de cuatro en cuatro, de seis en seis o formando una muralla humana de 12 personas. Además, les explicó, si fueran menos sería imposible poder agruparse de 6 formas diferentes.

“Bien sé que estos pedregosos caminos — les dijo — son estrechos en muchos tramos pero, dejando eso a un lado, ¿cuál es el menor tamaño que debería tener nuestro docto grupo para poder caminar exactamente en 64 formaciones distintas?”

Sólo cuando todos consiguieron la Compostelana decidió el mercader, como regalo, desvelarles la respuesta.

Entrada

El programa deberá leer, de la entrada estándar, un conjunto de casos de prueba. Cada uno estará formado por un único número $1 \leq n \leq 1.000.000.000$.

La entrada acabará con un `0`, que no deberá procesarse.

Salida

Para cada caso de prueba el programa escribirá el menor número de peregrinos que debe formar el grupo para que se puedan estructurar en exactamente n formaciones diferentes.

Se considera que no tiene sentido formaciones de más de 1.000.000.000 de personas, por lo que si la respuesta supera ese número se escribirá “`+INF`” en su lugar.

Entrada de ejemplo

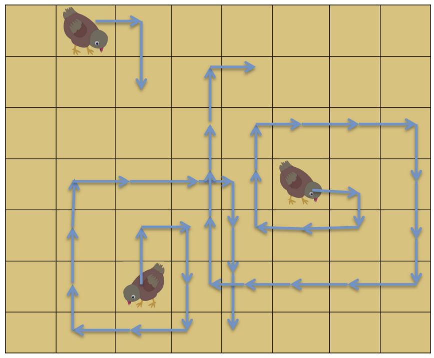
```
6
37
64
0
```

Salida de ejemplo

```
12
+INF
7560
```

● J

La comida de los pollitos



Los pollitos pasan todo el día picoteando el suelo del gallinero para comer los granos que van encontrando. Has estado varios días observando sus movimientos y has descubierto que todos ellos siguen un curioso patrón basado en las baldosas que hay en el suelo. Se despiertan mirando en una dirección (norte, sur, este, oeste) y comienzan a andar en esa dirección siguiendo un movimiento en espiral en el sentido de las agujas del reloj. El paseo termina cuando se cansan (cada pollito tiene un aguante distinto) o en su recorrido se topan con el borde del gallinero, momento en el que quedan aturdidos y se duermen hasta el día siguiente.

Además, en cada baldosa que pisan (incluída la que ocupan al despertarse) miran si hay pienso y si lo hay, comen un grano antes de dar el siguiente paso. Si no hay, simplemente siguen avanzando. Como son pequeños, cuando coinciden en un punto comen al mismo tiempo sin molestarse y a veces incluso duermen en el mismo sitio.

Para conseguir que tus pollitos crezcan lo más rápidamente posible sin gastar de más, has decidido distribuir los granos en el gallinero de forma que en cada paso cada pollito se encuentre un grano y pueda comerlo. Conoces la dirección en que se despiertan los pollitos, y el número de pasos que pueden dar en la espiral antes de cansarse y parar hasta el día siguiente. Ahora debes averiguar los granos que tienes que colocar en cada punto para que, al acabar el día, no quede ninguno y todos los pollitos hayan comido el máximo posible.

Entrada

La entrada comienza con una línea con el número de casos de prueba que deberán procesarse. Cada caso comienza con una línea con tres números, f, c y n indicando, respectivamente, el tamaño del gallinero en la dirección norte-sur, el tamaño en la dirección oeste-este y el número de pollitos ($1 \leq f, c \leq 50$, $0 \leq n \leq 500$). Las n líneas siguientes contienen la información de cada pollito. El primer número, v, indica la posición en la dirección norte-sur ($1 \leq v \leq f$), el segundo, h, la posición en la dirección oeste-este ($1 \leq h \leq c$), después se indica la dirección en la que empieza a moverse el pollito (`N`, `S`, `E`, `W`) y finalmente el número máximo de pasos que aguanta antes de dormirse a descansar (al menos uno).

Salida

Para cada caso de prueba se escribirán f líneas. En cada una se escribirán c valores, separados por un espacio en blanco, con el número de granos que hay que poner en cada punto. Tras cada caso de prueba se escribirá una línea con tres guiones (`---`).

Entrada de ejemplo

```
2
7 8 3
1 2 E 2
6 3 N 25
4 6 E 21
4 3 2
1 1 N 3
4 2 W 3
```

Salida de ejemplo

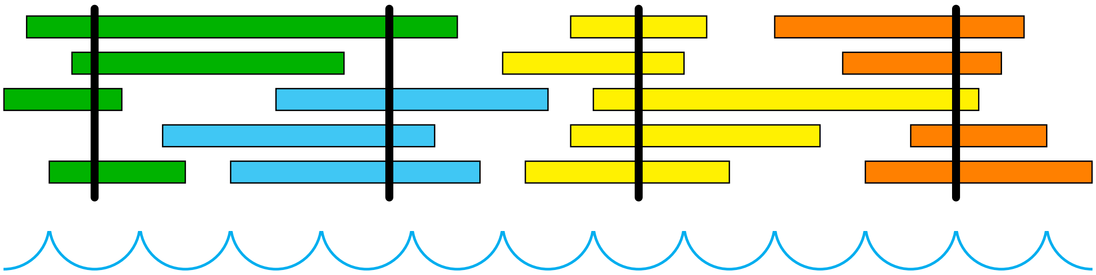
```
0 1 1 0 0 0 0 0
0 0 1 1 1 0 0 0
0 0 0 1 1 1 1 1
0 1 1 2 2 1 1 1
0 1 1 2 2 1 1 1
0 1 1 2 2 1 1 1
0 1 1 1 1 0 0 0
---
1 0 0
0 0 0
1 1 0
1 1 0
---
```

K

¡En primera línea de playa!

Ya nadie se cree, cuando un apartamento veraniego es anunciado con un gran *¡En primera línea de playa!*, que vaya a ser cierto. Por eso, los dueños de varios edificios de apartamentos (paralelos a la playa pero no en primera línea) han decidido construir pasadizos subterraneos (perpendiculares a la playa) que conecten todos los edificios con la arena. Así creen que los clientes estarán más safisfechos.

Como construir estos pasadizos no es barato, primero quieren saber cuántos túneles como mínimo serían necesarios. Por ejemplo, para la configuración de edificios de la figura (donde se han omitido los edificios en primera línea) son necesarios 4 túneles.



Entrada

La entrada consta de una serie de casos de prueba. Cada uno comienza con una línea con el número N de edificios ($1 \leq N \leq 100.000$). A continuación aparecen N líneas cada una con dos enteros que representan el extremo más occidental (W_i) y el más oriental (E_i) de cada edificio, con $W_i < E_i$, medidos en metros desde el extremo más occidental de la playa. Todas estas medidas son números enteros entre 0 y 10^9.

La entrada terminará con un caso sin edificios, que no debe procesarse.

Salida

Para cada caso de prueba se escribirá una línea con el mínimo número de pasadizos que es necesario construir. Los pasadizos deben ser de 1 metro de ancho y para ser útiles a un edificio deben estar completamente debajo de él cuando lo atraviesan.

Entrada de ejemplo

```
4
1 4
6 15
2 10
12 20
2
1 4
4 8
2
1 4
3 8
0
```

Salida de ejemplo

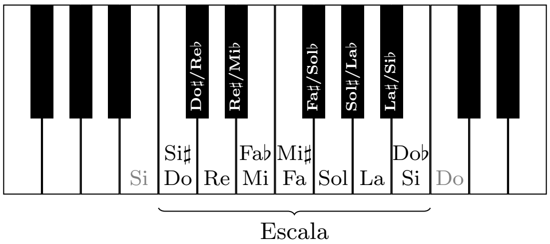
```
2
2
1
```

L
Teclas del piano

¡Qué ilusionado está Iker con su nuevo piano! Por fin va a poder tocar todas las canciones que le gustan. El vendedor le ha dicho que es un piano de muy buena calidad que le durará mucho tiempo, pero Iker no se fía del todo y ha decidido llevar la cuenta de cuántas veces toca cada tecla.

El teclado del piano está compuesto por 7 octavas[2] y cada octava consta de 12 notas (7 blancas y 5 negras) cada una a medio tono de distancia de la siguiente. Las teclas blancas corresponden a las notas Do, Re, Mi, Fa, Sol, La y Si (y a continuación vendrá el Do de la siguiente octava). Entre las notas Do y Re hay un tono de distancia por lo que hay una tecla negra en medio, mientras que entre Mi y Fa sólo hay medio tono de distancia. Las teclas negras corresponden precisamente a esos medios tonos y se nombran usando dos símbolos especiales: sostenido (♯) y bemol (♭). El sostenido aumenta medio tono mientras que el bemol disminuye medio tono.

De esta forma, la primera tecla negra de la octava corresponde a la nota Do ♯ pero también a Re ♭. Para liar un poco más las cosas la nota Mi ♯ está en la misma tecla que Fa, y un Do ♭ es lo mismo que el Si de la octava anterior.



¿Puedes ayudar a Iker a contar cuántas veces se pulsa cada tecla?

Entrada

El programa deberá leer, por la entrada estándar, múltiples canciones, cada una descrita mediante dos líneas. La primera línea indica el número de notas de la canción, y la segunda línea contiene las notas que la componen. Las notas aparecen separadas por espacios y siempre tienen el mismo formato: nombre de la nota, alteración (`#`, `b` o nada), y el número de octava. La octava más grave es la 1 y la más aguda la 7.

La entrada termina con una canción con 0 notas que no se debe procesar.

Salida

El programa deberá escribir una línea por cada canción que aparezca en la entrada, indicando cuántas veces se pulsó cada tecla del piano ordenadas desde la nota más grave hasta la más aguda. El primer número corresponderá al número de pulsaciones de la tecla más grave que aparece en la canción y el último a la más aguda. Es decir, la solución no debe empezar ni terminar por ceros.

Entrada de ejemplo

```
6
Do4 Do4 Re4 Do4 Fa4 Mi4
9
Mi5 Re#5 Mi5 Re#5 Mi5 Si4 Re5 Do5 La4
10
Do4 Do#4 Reb4 Re4 Re#4 Mib4 Mi4 Fab4 Mi#4 Fa4
0
```

[2] En realidad los pianos tienen algunas teclas más...

Salida de ejemplo

```
3 0 1 0 1 1
1 0 1 1 0 1 2 3
1 2 1 2 2 2
```