

Preparación

ADA BYRON

MADRID

EXPERT



Competitive Programming

UPV

Navegando sin teclado

Tiempo máximo: 2,000-4,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=318>

Lo de ayer fue un drama. Con los años que llevas criticando a todos los que abusan del ratón en lugar de utilizar las teclas rápidas, te tuvo que pasar a ti. Maldito teclado, que se tuvo que romper. Y encima en día festivo, imposible comprar otro. Todo el día utilizando el ordenador a través del lento e incómodo ratón. Menos mal que lo único que hiciste fue navegar por internet y pudiste ir de una página a otra utilizando los enlaces entre ellas.

Una persona normal con el orgullo herido compraría un teclado de reserva. Pero tú no. Con un flamante teclado nuevo harás un programa que te ayude a decidir la manera más rápida de navegar si te vuelves a ver en la misma situación.

Para eso has recopilado todas las páginas web que visitas y has hecho una lista de los enlaces que tiene cada una de ellas. Teniendo en cuenta tu página de inicio del navegador quieres saber cuál es la forma más rápida de llegar a una página determinada utilizando únicamente esos enlaces. Como estás obsesionado por la velocidad, el programa tendrá en cuenta que el tiempo de carga de cada página puede ser distinto y que una vez cargada, encontrar y pinchar en cada enlace depende de lo lejos que esté del principio de la página y de lo oculto que esté. Venga, ¡manos a la obra!



Entrada

La entrada está compuesta por múltiples casos de prueba. Cada uno representa un escenario de navegación distinto, con la descripción del conjunto de páginas que deben considerarse, sus tiempos de carga y los enlaces que hay entre ellas.

La primera línea de cada caso contiene el número N de páginas manejadas ($1 < N \leq 1.000$). La página de inicio del navegador es la 1 y la página que queremos visitar la N . En la línea siguiente N números indican el tiempo de carga (en milisegundos) de cada una de ellas.

Aparece después una línea con un número M indicando el número de enlaces entre páginas que se han identificado, tras lo que aparecen M líneas cada una describiendo un enlace: página origen, página destino y el tiempo (en milisegundos) que se tarda desde que la página carga hasta que el usuario encuentra el enlace y lo pulsa.

Se garantiza que todos los tiempos serán menores a 10 segundos y que en caso de que una página tenga más de un enlace a la misma URL, únicamente se proporciona aquel que puede utilizarse antes.

Salida

Para cada caso de prueba se escribirá el número mínimo de milisegundos que se tardará en poder navegar desde la página 1 hasta la página N contando el tiempo de carga de todas las páginas y el tiempo de localización y pulsación de cada enlace. Si no se puede alcanzar la página, se escribirá IMPOSIBLE.

Entrada de ejemplo

```
4
10 5 15 8
4
1 2 10
1 3 20
2 4 20
3 4 20
3
10 5 15
2
1 2 10
3 2 10
```

Salida de ejemplo

53 IMPOSIBLE

Autores: Marco Antonio Gómez Martín y Alberto Verdejo.

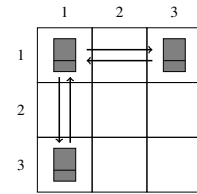
Revisor: Pedro Pablo Gómez Martín.

Repartidor de frigoríficos

Tiempo máximo: 1,000-4,000 s Memoria máxima: 16384 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=562>

Un repartidor de frigoríficos debe decidir en qué cruce de calles situar su local, de modo que la distancia total de todos los trasportes sea mínima. El repartidor trabaja en la ciudad de Novapolis donde todas las calles son en cuadrícula. Las distancias más cortas entre dos puntos se deben por tanto calcular siguiendo las calles y no como la distancia euclídea. La dificultad añadida es que en la furgoneta del repartidor solo cabe un frigorífico por lo que después de cada reparto debe volver al local a cargar el siguiente.



Por poner un ejemplo, en una cuadrícula de 3×3 en el que los lugares de reparto están en las posiciones (1, 1), (1, 3) y (3, 1), la mejor localización para el local del repartidor será el punto (1, 1).

Entrada

La entrada está formada por distintos casos de prueba. Cada caso de prueba consiste en varias líneas donde la primera línea son tres números enteros: C , el número de calles en dirección vertical, F , el número de calles en dirección horizontal y N , el número de frigoríficos a repartir. A continuación aparecen N líneas con dos números enteros cada una que indican la posición (x, y) de cada sitio de reparto. Se satisface que $1 \leq x \leq C$, $1 \leq y \leq F$, $1 \leq F$, $C \leq 10.000$ y $1 \leq N \leq 50.000$. Además, no hay posiciones repetidas.

El final de la entrada se indica con una línea con tres ceros que no se debe procesar.

Salida

Para cada caso de prueba, se escribirán en una línea las coordenadas (x, y) de la posición óptima del local del repartidor. En caso de existir varias posiciones óptimas, se escribirán las coordenadas de aquella tal que la suma $x+y$ sea mínima. Si aún así hubiera varias opciones, se debe considerar la posición con menor x .

Entrada de ejemplo

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

Salida de ejemplo

```
1 1
2 2
```

Autor: Luis Fernando Lago Fernández.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

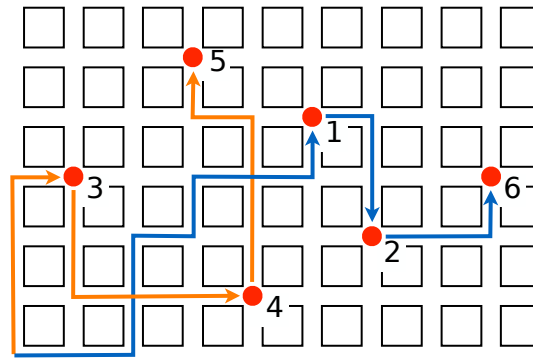
Yincana 2015

Tiempo máximo: 1,000-2,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=257>

En la zona residencial de *Concursolandia* organizan una yincana todos los años. Los concursantes, por parejas, recorren en orden una serie de puntos, repartidos por la urbanización, donde jueces que velan por el buen funcionamiento del concurso plantean una serie de acertijos a los concursantes, según van llegando. Como en muchas yincanas, los acertijos están encadenados: es imposible acertar uno si no se han recogido las pistas proporcionadas al resolver los acertijos planteados en todos los puntos anteriores (con un número menor).

El complejo residencial está formado por una serie de parcelas muy bien alineadas y separadas mediante avenidas horizontales (de este a oeste) o calles verticales (de norte a sur), como muestra el dibujo. Los puestos de control (círculos rojos) se encuentran en intersecciones entre calles. Los concursantes, que comienzan en la esquina inferior izquierda de la urbanización con pistas para resolver el primer acertijo, para alcanzar un punto tienen que caminar por estas calles, de la forma en la que decidan, pero no pueden atravesar por medio de parcelas. Los concursantes, para no perturbar en exceso a los vecinos, siempre van a la misma velocidad y tardan exactamente 1 minuto en ir de una intersección a la siguiente en horizontal o vertical. Gana el concurso el equipo que menos tiempo emplee en resolver todos los acertijos.



Mica y Dina forman pareja, pero tienen claro que ir siempre juntas les perjudica, y que podrían ganar tiempo si cada una sigue una ruta distinta, comunicándose por móvil las pistas según las vayan recibiendo para que la otra pueda avanzar. Al proponérselo a los jueces, estos no tienen claro si supone demasiada ventaja o si de permitirlo se llenaría el barrio de chicos pululando, por lo que imponen una restricción más: si optan por separarse, en todo momento solamente una de ellas podrá estar caminando y la otra deberá estar parada en un punto de control (o en el comienzo).

¿Cuál es el menor tiempo que necesitan Mica y Dina para resolver todos los acertijos respetando todas estas restricciones?

Entrada

La entrada comienza con un entero que indica el número de casos de prueba que vendrán a continuación. Cada caso comienza con el número N (entre 1 y 500) de puntos de control que forman la yincana. A continuación aparecen N líneas cada una con dos enteros que indican la avenida y la calle en los que se encuentran cada uno de estos puntos, en el orden en el que deben ser visitados. Todos los puntos están en intersecciones diferentes, y ninguno está en el punto de salida. Tanto las avenidas como las calles están numeradas desde 0 hasta 10.000. Los concursantes parten siempre de la intersección (0,0), pero este punto no se contabiliza dentro de los N a visitar.

Salida

Para cada caso de prueba se escribirá el mínimo tiempo necesario para ganar la yincana, si se compete por parejas y se siguen las restricciones impuestas por los jueces.

Entrada de ejemplo

```
2
6
4 5
2 6
3 1
1 4
5 3
3 8
2
4 4
6 6
```

Salida de ejemplo

```
29
12
```

Autor: Alberto Verdejo.

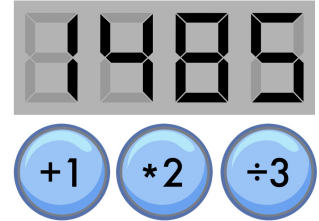
Revisores: Pedro Pablo Gómez Martín y Marco Antonio Gómez Martín.

La máquina calculadora

Tiempo máximo: 1,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=319>

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 $\div 3$, 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
```

Autor: Alberto Verdejo.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Esquiando en Alaska

Tiempo máximo: 1,000-3,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=254>

Para celebrar el aniversario de la exitosa y extravagante serie televisiva de los noventa *Doctor en Alaska* se ha organizado una competición invernal en el lejano y maravilloso pueblo de Cicely, Alaska, donde participarán sus estrafalarios vecinos y el médico neoyorquino, judío y urbanita, Dr. Joel Fleischman.

Los productores, cumpliendo con ciertos compromisos publicitarios, han recibido unos esquís que deben repartir entre los participantes, teniendo en cuenta que se esquía mejor cuando la longitud de los esquís es acorde con la altura del esquiador.

En concreto, el responsable de la competición recomienda minimizar la suma de las diferencias (en valor absoluto) entre la altura de cada esquiador y la longitud de los esquís que le han sido asignados. Pero los productores no saben cómo conseguirlo, por lo que te han contratado, poniendo en tus manos el éxito del evento.



Entrada

La entrada consta de una serie de casos de prueba. Para cada caso, primero aparece el número N de esquiadores y esquís que hay que emparejar (entre 1 y 200.000). A continuación aparecen dos líneas con N enteros cada una, la primera con las alturas de los esquiadores y la segunda con las longitudes de los esquís (todos números entre 1 y 1.000.000).

La entrada termina con un caso sin esquiadores.

Salida

Para cada caso de prueba se escribirá una línea con la mínima suma de diferencias entre cada esquiador y sus esquís. Se garantiza que el resultado nunca será mayor que 10^9 .

Entrada de ejemplo

```
3
10 15 20
16 12 23
2
175 200
140 150
0
```

Salida de ejemplo

```
6
85
```

Autor: Alberto Verdejo.

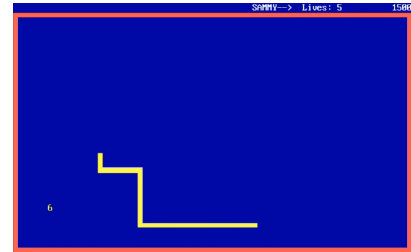
Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Snake

Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=552>

Snake es un videojuego clásico que se creó en los años 70 y se popularizó algunos años más tarde. Consiste en controlar a una serpiente evitando que choque contra las paredes y contra su propia cola. La serpiente siempre está en movimiento, pero el jugador puede controlar la dirección en la que su cabeza avanza (izquierda, derecha, arriba o abajo). El cuerpo de la serpiente sigue exactamente el movimiento de la cabeza. Para complicar las cosas, la cola de la serpiente va creciendo y cada vez queda menos espacio disponible. El juego termina cuando la cabeza de la serpiente choca, o bien contra una pared, o bien contra su propio cuerpo.



Suponiendo un panel de juego cuadrado de tamaño 21×21 casillas, nuestro objetivo es determinar en qué instante acaba el juego dada la secuencia de movimientos realizados por el jugador.

El juego comienza en el instante $t=0$ con una serpiente de longitud 10, alineada en dirección vertical, con la cabeza en el centro del tablero y moviéndose hacia arriba. En ese momento tiene por delante 10 casillas en blanco hasta el borde superior.

La serpiente siempre se mueve a razón de una casilla por unidad de tiempo en base a la dirección que lleva, por lo que en el instante $t=1$ estará una unidad más cerca del borde superior. Si no hay movimientos, en $t=10$ la cabeza de la serpiente estará tocando el borde del tablero y al intentar avanzar otra vez no podrá y colisionará. Eso sí, si en $t=10$ el usuario se mueve a la derecha, salvará la situación y la serpiente se moverá pegada al borde superior hasta el borde derecho, que alcanzará en $t=20$. Si no hay otros movimientos, la partida terminará ahí.

Como hemos dicho, la serpiente va haciéndose más grande con el paso del tiempo. En concreto cada 10 unidades de tiempo crece una unidad. Eso significa, por ejemplo, que en $t=10$ la serpiente tiene las 10 unidades de longitud iniciales pero en $t=11$ pasa a ocupar una unidad más. Ten en cuenta que la serpiente *crece por la cola*, por lo que si el final de la serpiente en $t=10$ estaba en la posición (x,y) , esa posición seguirá ocupada en $t=11$ y, por lo tanto, si la cabeza iba a colocarse en esa posición, habrá colisión y terminará la partida.

Entrada

La entrada está formada por distintos casos de prueba, cada uno en una línea distinta.

En cada línea aparece en primer lugar un número entero N ($1 \leq N \leq 1.000$), que indica el número de movimientos que siguen a continuación. Tras esto la línea contiene una lista con la descripción de los N movimientos. Un movimiento viene descrito por la pareja (t_i, l_i) , $i = 1, \dots, N$, con t_i un número entero y l_i una letra mayúscula. El número, $1 \leq t_i \leq 2.000$, indica el instante en el que se realiza el movimiento (será visible en t_i+1). La letra indica la dirección del mismo de acuerdo al siguiente esquema: U = arriba (*up*), D = abajo (*down*), L = izquierda (*left*), R = derecha (*right*). Se garantiza que los movimientos aparecen siempre ordenados por el instante en que se realizan.

El final de la entrada se indica con una línea con un único 0 que no se debe procesar.

Salida

Para cada caso de prueba, se escribirá en una línea el último instante de tiempo en el que la serpiente pudo avanzar, justo antes de la primera colisión. Este instante siempre existirá, pues después del último movimiento, en el supuesto de que se llegue a este punto, la serpiente continúa en línea recta hasta chocar. Por otra parte es posible que en algunos casos el juego termine antes de que se realicen todos los movimientos.

Entrada de ejemplo

```
1 100 R
1 10 R
1 1 D
3 1 R 2 D 6 L
0
```

Salida de ejemplo

```
10
20
1
17
```

Autor: Luis Fernando Lago Fernández.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Evitando tropiezos

Tiempo máximo: 2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=688>

Aunque las escaleras son para niños y ancianos un reto que puede llegar a intimidarles, para la mayoría de la gente sin problemas de movilidad constituyen solo un mero obstáculo que puede ser superado mientras se habla, se corre o incluso se mira el móvil.

Pero para tener garantías de que esto es así, los arquitectos dedican parte de su tiempo a decidir el tamaño de la huella (la parte plana de un peldaño), de la contrahuella (la altura de un escalón) y la relación entre ellas. La fórmula de François Blondel, por ejemplo, les dice, desde 1675, el rango en el que debe moverse la suma de dos contrahuellas y una huella para que la escalera final sea armoniosa y amigable de ascender.



Una consideración mucho menos conocida dice que, para evitar tropiezos, la diferencia entre las contrahuellas de dos peldaños consecutivos no debe superar un determinado umbral. Al fin y al cabo, cuando levantamos la pierna para superar el siguiente escalón, nuestro subconsciente lo hace en relación al esfuerzo realizado para subir el último, y una variación grande puede hacernos tropezar.

Además, para que la escalera no quede demasiado irregular, también se suele imponer una diferencia máxima entre las contrahuellas de cualquier pareja de escalones. De otro modo, podríamos terminar teniendo peldaños muy grandes y muy pequeños, que generan una sensación de inestabilidad al mirar a la escalera en conjunto y la hacen, además, poco ergonómica.

Entrada

Cada caso de prueba comienza con tres números, c , m y n . El primero indica la diferencia máxima permitida entre las contrahuellas de dos peldaños consecutivos. Por su parte, el segundo indica la diferencia máxima permitida entre la contrahuella de dos peldaños cualesquiera. Se cumple que $0 \leq c \leq m \leq 10^6$. El último, $2 \leq n \leq 200.000$, indica el número de peldaños en la escalera.

A continuación aparece, en otra línea, un primer número indicando la altura del inicio de la escalera, respecto a algún punto de referencia indeterminado (la calle, el nivel del mar...). Después, vendrán n números más, con la altura a la que se encuentra cada peldaño, incluido el último que marca el fin de la escalera. Se garantiza que los números serán siempre crecientes y no mayores que 10^9 .

Salida

Por cada caso de prueba se escribirá “Ok” si la escalera es correcta y “Tropiezo” si no lo es.

Entrada de ejemplo

```
1 10 4
100 120 141 161 180
1 3 5
0 1 3 6 10 15
```

Salida de ejemplo

```
Ok
Tropiezo
```

Autor: Pedro Pablo Gómez Martín.

Revisores: Marco Antonio Gómez Martín y Alberto Verdejo.

Las perlas de la condesa

Tiempo máximo: 1,000-3,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=399>

El collar de perlas de la condesa es conocido por la calidad y la cantidad de sus perlas. Lo luce en las grandes fiestas, dándole tres o cuatro vueltas alrededor de su cuello. La perla del centro es la de mayor tamaño, y según nos desplazamos hacia los extremos son estrictamente más pequeñas. Las perlas están perfectamente seleccionadas y colocadas, lo que hace al collar completamente simétrico respecto a la perla central.



Durante el último baile, se ha roto el cierre y las perlas han rodado por el suelo. Los invitados han intentado recuperarlas todas ensartándolas en un nuevo cordel según las encontraban. Al día siguiente, la condesa ha llamado a su joyero para que las engarce de nuevo en el orden adecuado. Éste mide el diámetro de cada una de ellas y procede a rehacer el collar colocando la perla central y a continuación las siguientes en tamaño hasta llegar a los extremos. A la condesa no le importa si se ha perdido alguna perla, siempre y cuando el collar siga siendo completamente simétrico.

Entrada

Cada línea de la entrada forma un caso de prueba, que consiste en una lista de números positivos separados por espacios. Cada uno representa el diámetro de una de las perlas en el orden en el que las fueron recogiendo los invitados. La perfección del collar es legendaria, por lo que el diámetro se mide con una unidad de medida que muchos consideran infinitesimal, aunque el diámetro será siempre menor que 2^{31} . Cada lista tiene $0 < \text{perlas} < 1.000$ y acaba siempre con un cero.

El último caso de prueba, que no deberá procesarse, contiene un collar vacío, representado por una lista con un único cero.

Salida

Para cada caso de prueba, el programa escribirá "NO" si no es posible formar un collar simétrico con todas y cada una de las perlas encontradas de forma que la perla de mayor tamaño quede en el centro. En otro caso, se escribirán, separados por espacios, los diámetros de las perlas tal y como han quedado ordenadas en el collar, desde un extremo al otro.

Entrada de ejemplo

```
2 3 2 3 5 7 5 0
2 2 7 10 0
10 0
0
```

Salida de ejemplo

```
2 3 5 7 5 3 2
NO
10
```

Autor: Isabel Pita.

Revisores: Marco Antonio Gómez Martín, Alberto Verdejo y Pedro Pablo Gómez Martín.

Helados de cucurucho

Tiempo máximo: 1,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=325>

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 vainilla tiene la lengua demasiado fría y apenas percibe el sabor con claridad. A Blanca también le gusta más el chocolate que la vainilla, 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
```

Autor: Pedro Pablo Gómez Martín.

Revisores: Alberto Verdejo y Marco Antonio Gómez Martín.

El burro y las alforjas

Tiempo máximo: 1,000-3,000 s Memoria máxima: 10240 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=249>

El burro es un animal actualmente en peligro de extinción, debido principalmente a que ya no es necesario para el trabajo en el campo. No ocurría así antiguamente, cuando este animal, noble, fuerte, pero sobre todo testarudo, ayudaba a los hombres en el transporte de sus cosechas y enseres. Utilizado mucho en el antiguo Egipto, era normal ver caravanas de pollinos avanzar con sus dos alforjas por los caminos.



Yafeu, el joven administrador de las propiedades de Amenophis a orillas del Nilo, debe cargar la cosecha de trigo en su caravana de asnos para trasladarla a la capital. Los campesinos han recogido la cosecha en sacos, cuyos pesos no son necesariamente iguales. Cada burro lleva un par de alforjas, colgadas una a cada lado del lomo. En cada alforja se lleva un saco, que puede ser de cualquier peso aunque es importante que las dos alforjas lleven el mismo peso para que el burrito vaya equilibrado y no se caiga.

Yafeu tiene mucho trabajo, y le aburre dedicar todo un día a emparejar los sacos. Este año le han recomendado un estudiante asegurándole que le ahorrará mucho tiempo. Está dispuesto a darle una buena recompensa si eso es cierto.

Entrada

La entrada comienza con un entero que indica el número de casos de prueba que vendrán a continuación. Cada caso consta de dos líneas. La primera contiene el número de burros disponibles y el número de sacos que hay que cargar (ambos entre 1 y 1.000.000). En la línea siguiente aparecen los pesos en gramos de cada saco separados por espacios. Cada uno pesa como mucho 20 Kg (valores entre 1 y 20.000).

Salida

Para cada caso de prueba se indicará el máximo número de burros que se pueden cargar.

Entrada de ejemplo

```
3
6 10
1 2 3 1 2 3 1 2 3 1
2 6
3 3 2 2 1 1
3 7
3 14 15 92 65 35 89
```

Salida de ejemplo

```
4
2
0
```

Autor: Isabel Pita.

Revisores: Marco Antonio Gómez Martín, Alberto Verdejo y Pedro Pablo Gómez Martín.

Durmiendo en albergues

Tiempo máximo: 1,000-2,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=400>

Como cualquier buen peregrino sabe, los albergues que uno encuentra por los caminos suelen tener grandes habitaciones donde hileras de camas unas junto a las otras esperan a ser ocupadas.

Aunque esas camas han sido usadas por infinidad de gentes antes, el cansancio hace que a uno no le importe tumbarse en colchones raídos por el tiempo con tal de dormir un puñado de horas seguidas y coger fuerzas para la jornada siguiente.

De hecho, para maximizar las posibilidades de dormir bien, lo más importante no es la salubridad del colchón, sino la distancia con tu vecino peregrino más cercano. Y es que una noche con un desconocido en la cama de al lado roncando puede arruinarte la noche mucho más que un puñado de ácaros.

Cuando entras en una de estas habitaciones, pues, el objetivo es encontrar la cama que maximice la distancia con el vecino más cercano (hacia cualquiera de los lados).



Entrada

La entrada está formada por distintos casos de prueba representando la ocupación de una hilera de camas una noche del camino.

Cada hilera aparece en una única línea que contiene una secuencia de . y X (hasta 500.000 caracteres). Los puntos representan camas libres, mientras que las X representan camas ocupadas. Se garantiza que habrá siempre al menos una cama libre y otra ocupada.

Salida

Por cada caso de prueba se escribirá una línea con el número de camas vacías más grande posible que se puede conseguir entre la cama seleccionada y tu vecino más cercano.

Entrada de ejemplo

```
.X.X.  
.X...X.  
.X....X.  
...X
```

Salida de ejemplo

```
0  
1  
1  
2
```

Autor: Marco Antonio Gómez Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

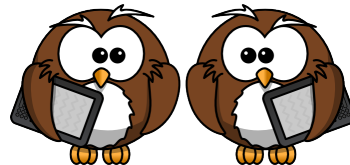
¿Acaso hubo búhos acá?

Tiempo máximo: 1,000 s Memoria máxima: 2048 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=252>

Juan Filloy, un escritor argentino nacido en 1894, se autoproclamó el “*recordman mundial de palíndromía*” pues gracias a él hoy conocemos más de 8.000 palíndromos españoles.

En justicia, sin embargo, debemos decir que ese título hoy día debería llevárselo Victor Carbajo, un músico y compositor español que en su web¹ mantiene una colección de más de 100.001 palíndromos que amplía regularmente.



Un palíndromo es una palabra o frase que, tras unificar mayúsculas y quitarle tildes, espacios y signos de puntuación, se lee igual de izquierda a derecha que de derecha a izquierda. Un ejemplo es el título de este problema, uno de los palíndromos más famosos de Juan Filloy.

Asociada a los palíndromos, existe la pregunta casi filosófica de si éstos se *inventan* o se *descubren*. Nosotros nos quedaremos con la segunda opción e intentaremos descubrir, dada una frase, si es o no palíndroma.

Entrada

La entrada estará compuesta por múltiples casos de prueba. Cada caso de prueba es una única línea con una palabra o frase de no más de 100 caracteres. En ella puede haber tanto letras mayúsculas como minúsculas del alfabeto inglés y uno o varios espacios separando palabras (eso sí, las líneas empezarán y terminarán siempre con letra, nunca con espacios). A riesgo de comprometer la ortografía y la semántica, las palabras no contendrán tildes y los signos de puntuación se omiten.

El último caso de prueba va seguido de una línea con XXX que marca el final y no debe ser procesada.

Salida

Por cada caso de prueba se escribirá SI si la palabra o frase es palíndroma y NO en caso contrario.

Entrada de ejemplo

```
Acaso hubo buhos aca
Querido muerto esta tarde llegamos
XXX
```

Salida de ejemplo

```
SI
NO
```

Notas

El segundo caso de prueba, claramente no palíndromo, es una modificación (cambiando Señor por Querido para evitar caracteres fuera del alfabeto inglés) de un ejemplo famoso en el que se pone de manifiesto la importancia de los signos de puntuación y las tildes: “Señor muerto, esta tarde llegamos.” frente a “¡Señor! ¡Muerto está! ¡Tarde llegamos!”.

Autores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

Revisores: Alberto Verdejo y Isabel Pita.

¹<http://www.carbajo.net/varios/pal.html>

Traduciendo los apuntes

Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=553>

El profesor de programación tiene mucho trabajo traduciendo los apuntes al inglés, así que ha pedido a sus compañeros que le ayuden con tan apasionante tarea. Se han repartido el trabajo de tal modo que cada uno tiene que traducir una parte del temario. Pero, como son un poco despistados, a algunos se les ha olvidado la parte que les tocaba traducir y al final cada uno ha traducido las páginas que le ha dado la gana. Como resultado tienen un conjunto de intervalos de páginas traducidas que en algunos casos solapan, mientras que otras páginas se han quedado sin traducir. Así las cosas, el profesor se pregunta cuántas páginas están traducidas al inglés.



Entrada

La entrada está formada por varios casos de prueba, cada uno en una línea.

Cada línea comienza con un número N ($1 \leq N \leq 2.000$), que indica el número de profesores que colaboran en la traducción. A continuación aparecen N parejas de números (x_i, y_i) , $i=1, \dots, N$, que indican la primera y la última página traducidas por cada uno de los profesores. Siempre se cumple que $1 \leq x_i \leq y_i \leq 50.000.000$.

El final de la entrada se indica con una línea con un único 0 que no debe procesarse.

Salida

Para cada caso de prueba, se escribirá en una línea el número total de páginas traducidas.

Entrada de ejemplo

```
3 10 20 30 40 15 25
2 12 24 3 12
0
```

Salida de ejemplo

```
27
22
```

Autor: Luis Fernando Lago Fernández.

Revisores: Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.