

Preparación

ADA BYRON

MADRID

NEWBIE



Competitive Programming

UPV

Diagoland

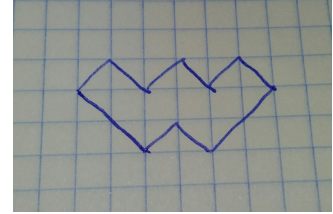
Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=690>

En *Diagoland* los niños en el colegio aprenden a escribir en *cursiva*, les cuesta mucho entender el concepto de *ángulo recto*, tienen mucha manía a Pitágoras y prefieren los rombos a los rectángulos.

Curiosamente, sin embargo, utilizan hojas de cuadros. Pero cuando hacen dibujos sobre ellos no utilizan líneas horizontales ni verticales, sino diagonales.

Dada una hoja de cuadros con varios polígonos (que no se tocan), ¿cuál es su área total?



Entrada

La entrada está compuesta por distintos casos de prueba terminados por una línea con dos ceros.

Cada caso de prueba comienza con una línea con dos números que indican el tamaño de la hoja de cuadros donde están los polígonos; el primero indica el ancho y el segundo el alto. Ninguno de los dos números será superior a 100.

A continuación aparece el dibujo. Los caracteres posibles son las dos barras diagonales que se utilizan para marcar los límites de cada polígono y el punto (“.”) para indicar cuadrados vacíos.

Los polígonos nunca comparten lados ni vértices. Se garantiza, además, que ningún polígono queda completa ni parcialmente contenido en otro.

Salida

Por cada caso de prueba se escribirá un único número con la suma del área de todos los polígonos del dibujo medida en número de cuadrados.

Entrada de ejemplo

```
6 3
/\ /\
\.../
.\ \/.
6 5
.. \..
/\ \ /\
\.\ \/.
.\ \ \/.
..\ \ \..
0 0
```

Salida de ejemplo

```
10
12
```

Autor: Marco Antonio Gómez Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

Computus Algorithmicus

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

<http://www.aceptaelreto.com/problem/statement.php?id=644>

Todos los años cuando se acerca la Semana Santa surge la misma pregunta. ¿Por qué la fecha exacta cambia de un año a otro?

Hace 1.500 años no existía la electricidad, ni la iluminación a gas, ni se habían inventado las farolas. Así que caminar por las calles en una noche sin luna podía resultar un tanto complicado. Por ello, el monje y erudito Dionisio el Exiguo (que era bajito, pero muy inteligente) convenció al Papa para que la Semana Santa se celebrase siempre en primavera y en tiempo de luna llena.

En concreto, en el año 525 Dionisio reflejó este cálculo, vigente hoy en día, en el llamado *Computus*: “el Domingo de Pascua será el primer domingo posterior a la primera luna llena posterior al equinoccio de primavera”.

Muchos siglos después, Carl Friedrich Gauss (que no era tan bajo, aunque nació en la Baja Sajonia) desarrolló un algoritmo para el cálculo del Computus. Sea A el año en cuestión, y M y N dos variables que van cambiando dependiendo del siglo (para el siglo XX fueron 24 y 5 respectivamente, coincidiendo, curiosamente, con los valores para el siglo XXI). El algoritmo de Gauss es el siguiente:

- Sea a el resto de la división $A/19$.
- Sea b el resto de la división $A/4$.
- Sea c el resto de la división $A/7$.
- Sea d el resto de la división $(19a + M)/30$.
- Sea e el resto de la división $(2b + 4c + 6d + N)/7$.
- Si $d + e$ es menor que 10, entonces el Domingo de Pascua es el $(d + e + 22)$ de marzo.
- En caso contrario, el Domingo de Pascua es el $(d + e - 9)$ de abril.

Hay que tener en cuenta, no obstante, que en ocasiones falla retrasando la fecha una semana. En concreto:

- Si la fecha obtenida es el 26 de abril, debe cambiarse por el 19 de abril.
- Si el resultado final conseguido es el 25 de abril con $d=28$, entonces la fecha correcta es el 18 de abril.

Dado un año entre 1900¹ y 2099, inclusive, debes calcular en qué día cae el Domingo de Pascua usando el algoritmo de Gauss.

Entrada

La entrada está compuesta por distintos casos de prueba, cada uno en una línea. Cada una de ellas tendrá un único número entero, el número A del año (entre 1900 y 2099).

Tras el último caso de prueba vendrá una línea con un 0 que no debe ser procesada.

Salida

Por cada caso de prueba se escribirá una línea indicando el día en que cae el Domingo de Pascua. La fecha se escribirá con el formato x de marzo o y de abril.

¹Para este problema, los siglos comienzan en el año terminado en 00.



Entrada de ejemplo

2019 2002 1992 0

Salida de ejemplo

21 de abril 31 de marzo 19 de abril

Autor: Ginés García Mateos.

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

La serie de Leibnitz

Tiempo máximo: 2,000-3,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=561>

La serie de Leibnitz es una serie infinita cuyo valor converge a $\pi/4$:

$$1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \frac{\pi}{4}$$

Aunque puede utilizarse para calcular el valor de π , en la práctica la convergencia es muy lenta y es necesario sumar muchos términos para lograr una precisión de unos pocos decimales. Por ejemplo, sumando el primer millón de términos de la serie se obtiene un valor de π igual a 3.1415916535..., que sólo es exacto hasta la quinta cifra decimal.

El objetivo de este ejercicio es calcular la suma de los primeros N términos de la serie de Leibnitz con un determinado número M de posiciones decimales. Es importante calcular cada uno de los términos de la suma con el número de decimales requerido antes de sumarlo. Nótese que el número obtenido de este modo puede ser distinto del resultado de truncar a M decimales la suma de los mismos N términos calculados con un número de cifras decimales mayor.

Por ejemplo, la suma de los 100 primeros términos de la serie, calculados con 10 decimales, da como resultado:

0.7828982264

En cambio sumando los mismos 100 términos, calculados con 50 decimales se obtiene:

0.78289822588963819107685355956923012910076359610164

Los dos últimos decimales del primer resultado (64) no son iguales a los de esa misma posición en el segundo (58).

Entrada

La entrada está formada por distintos casos de prueba, cada uno descrito por dos números enteros. El primer entero, N , es el número de términos de la serie que se quiere sumar. El segundo entero, M , es el número de cifras decimales que se desea tener. Se garantiza que $1 \leq N \leq 10.000$ y $0 \leq M \leq 500$.

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 valor de la suma de los N primeros términos de la serie de Leibnitz con M cifras decimales.

Entrada de ejemplo

```
100 10
100 50
100 0
1 12
2 7
0
```

Salida de ejemplo

```
0.7828982264
0.78289822588963819107685355956923012910076359610164
1.
1.000000000000
0.6666667
```

Autor: Luis Fernando Lago Fernández.

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

¡Me caso!

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

<http://www.aceptaelreto.com/problem/statement.php?id=380>

Cada vez que un familiar o un amigo me dice las temidas palabras “¡Me caso!”, mientras mi cara pone gesto de alegría mi cartera echa a temblar. Luego viene la hipocresía de la invitación. ¿Pero cómo que “*invitación*”!? La RAE lo deja bien claro... La segunda acepción de *invitar* es “Pagar el gasto que haga o haya hecho otra persona, por gentileza hacia ella”. Sí, es verdad... formalmente la comida o cena no la pago directamente; pero si no se hace un regalo digno de la celebración, te pondrán en la lista de los invitados poco populares.



Esta opinión dura hasta que... el que se casa eres tú. Y es que, creo que no lo he dicho, ¡me caso! Y desde que lo sé, soy más de la primera acepción de la RAE: “Llamar a alguien para un convite o para asistir a algún acto”. Que, digo yo, el que se casa soy yo, vale, pero todos lo vamos a celebrar, nos vamos a divertir, y vamos a engordar un par de kilos... no va a salir todo de mis costillas, ¿no?

Y es que esto de las bodas es muy bonito, un día inolvidable y todo lo que tú quieras. Pero por muy especial que sea y lo hagamos con toda la ilusión sin llevarnos nada a cambio, el que imprime las invitaciones, el que hace los obsequios para los invitados y el del banquete quieren su dinero, por mucho que nos feliciten. Así es que nos pasamos las noches en vela, sumando y sumando gastos, pensando en poner en la limusina el logo de algún banco que nos patrocine a ver si conseguimos no exprimir tanto a nuestros, ejem, *invitados*.

Entrada

Cada caso de prueba comienza con un número n indicando la cantidad de gastos que se sufren al organizar una boda (como mucho 50.000). A continuación aparecen, en otra línea, n números positivos con todos esos gastos.

La entrada termina con una boda sin gastos (obviamente imposible) que no debe procesarse.

Salida

Para cada caso de prueba se escribirá, en una línea independiente, el coste total de la boda. Todo está por las nubes, pero la suma de los gastos será menor que 10^9 .

Entrada de ejemplo

```
4
100 200 100 200
2
50 25
0
```

Salida de ejemplo

```
600
75
```

Autor: Pedro Pablo Gómez Martín.

Revisores: Marco Antonio Gómez Martín y Alberto Verdejo.

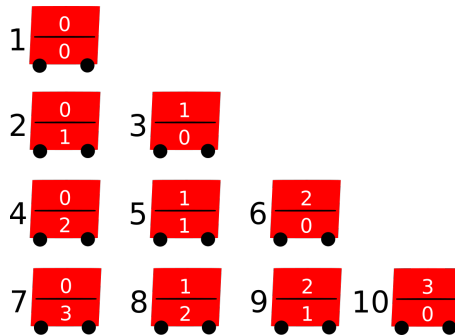
Double decker

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

<http://www.aceptaelreto.com/problem/statement.php?id=321>

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
```

Autor: Enrique Martín Martín.

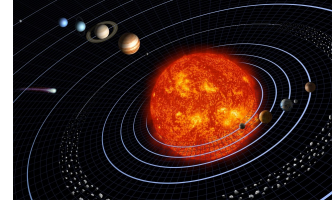
Revisores: Pedro Pablo Gómez Martín y Marco Antonio Gómez Martín.

Alineación planetaria

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

<http://www.aceptaelreto.com/problem/statement.php?id=381>

Cuando los planetas se alinean en el cielo, la asociación *Amantes del Día de la Alineación* (ADA) organiza la jornada más divertida en el observatorio del Monte Tajinaste. Los investigadores traen a sus familias, comen todos juntos, bailan... Después de la merienda, y tras cantar la canción “Adiós, planetas en línea” todo el mundo recoge y se marcha a casa. Pero siempre surge la pregunta ¿cuándo volverán a tener la excusa para reunirse? Normalmente esa pregunta la responde Calculón, pero ha conectado una baja médica, un permiso de paternidad y unos moscosos que le sobran y es muy posible que no se le vuelva a ver por el observatorio hasta Navidad. La de dentro de tres años.



En el sistema hay varios planetas y para cada uno se conoce su periodo exacto de traslación, es decir, cuántos días tarda en dar una vuelta completa alrededor del sol. Sabiendo que hoy todos los planetas estaban alineados, ¿podrías vaticinar cuándo se volverán a alinear en esa misma posición?

Entrada

Cada caso de prueba consiste en una línea con el número N de planetas en el sistema, seguido de una línea con N enteros con el periodo de traslación de cada planeta (en días) alrededor del sol. Nada impide que planetas diferentes tengan el mismo periodo de traslación. El periodo de traslación está entre 1 y 70 días, y siempre es un valor entero. Además, todos los sistemas tienen entre 2 y 5 planetas.

El último caso de prueba es un único 0, que no debe ser procesado.

Salida

Para cada caso de prueba se mostrará una línea con el mínimo número de días que tardarán los planetas en volver a alinearse en la misma posición.

Entrada de ejemplo

```
2
4 2
3
2 3 5
0
```

Salida de ejemplo

```
4
30
```

Autor: Enrique Martín Martín.

Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

RunnersApp

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

<http://www.aceptaelreto.com/problem/statement.php?id=653>

Los *runners* son una plaga. Son personas equipadas con ropa técnica profesional, relojes inteligentes, móviles de última generación y auriculares inalámbricos que nunca olvidan hacerse un *selfie* antes de salir de casa a dar una vuelta (lo de correr y sudar es opcional).

En este escenario, los desarrolladores de la famosa *RunnersApp* nos han pedido ayuda para implantar una nueva funcionalidad. En concreto, esta aplicación va a registrar el tiempo invertido por el “deportista” en recorrer intervalos de 100 metros de cada una de sus salidas y, al final de la misma, quieren que se indique qué tramo de un kilómetro se ha recorrido más rápidamente y en cuánto tiempo se ha hecho.



Entrada

La entrada está formada por distintos casos de prueba, cada uno ocupando dos líneas.

La primera línea de cada caso contiene el número de hectómetros (100 metros) recorridos por el *runner*. La segunda tiene tantos números enteros como hectómetros indicando el tiempo que ha invertido en recorrer cada uno de los tramos. En cada salida, los usuarios de la aplicación siempre corren al menos un kilómetro y nunca más de 10.000. Además, como mucho mínimo tardan 1 segundo y como máximo 30 en recorrer 100 metros.

Tras el último caso de prueba viene una línea con un 0 que no debe procesarse.

Salida

Por cada caso de prueba, se escribirá una línea que comenzará con dos números separados por guiones indicando el principio y fin del kilómetro que se ha recorrido en menos tiempo. A continuación se escribirá el tiempo necesario para recorrerlo (en formato `minutos:segundos`, los segundos siempre con dos dígitos).

En caso de existir varios tramos de 1 kilómetro en los que se haya invertido el mismo tiempo, siempre se elegirá el tramo que se haya realizado más tarde, ya que tiene más mérito hacer un buen tiempo cuando se está cansado que cuando se está fresco.

Entrada de ejemplo

```
15
10 10 1 2 1 2 1 2 1 2 1 2 10 10 10
15
10 10 1 1 1 1 1 1 1 1 1 1 1 1 10
15
30 30 6 6 6 6 6 6 6 6 6 6 7 30 30 30
0
```

Salida de ejemplo

```
200-1200 0:15
400-1400 0:10
200-1200 1:01
```

Autores: Pablo Trinidad y Marco Antonio Gómez Martín.

Revisor: Pedro Pablo Gómez Martín.

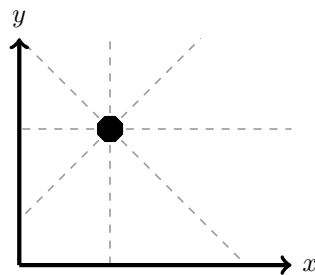
Espantaperros

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

<http://www.aceptaelreto.com/problem/statement.php?id=685>

El *Espantaperros* (también conocida como *Torre de la Atalaya*) es una torre construida en el siglo XII que se encuentra en uno de los extremos de la Alcazaba de Badajoz. Existen varias teorías sobre el origen de este nombre tan peculiar. Por un lado, se cree que el fuerte y agudo sonido que emitía la campana situada en lo alto de la torre hacía huir a los perros. Por otro lado, la tradición popular cuenta que esta campana servía para avisar a los cristianos del comienzo del culto, lo cual espantaba al resto de habitantes de la zona, a los que denominaban *perros* por no seguir la fe cristiana. En cualquier caso, esta torre también tenía una finalidad militar, ya que su elevada altura permitía divisar a potenciales intrusos.

Vista desde arriba, la torre del Espantaperros tiene forma de octógono regular. Los vigilantes se colocaban en lo alto de la torre, de modo que podían tener una vista completa de todo lo que ocurría alrededor. Sin embargo, cuando las condiciones meteorológicas obligaban a ello, los vigilantes tenían que bajar al interior de la torre y desde allí ponerse a vigilar a través de las estrechas ventanas que hay en cada pared. Ahí tenían mucho más difícil detectar a los intrusos porque la zona de visión del vigilante se limita a las ocho semirrectas que salen a través de las ventanas (ver figura). Suponemos que cada una de estas semirrectas es paralela a alguno de los ejes cartesianos, o bien forma un ángulo de 45° con alguno de ellos.



Entrada

La entrada consta de una serie de casos de prueba, cada uno en una línea. Cada caso consiste en cuatro números enteros comprendidos entre 0 y 10^9 . Los dos primeros indican la posición en la que se encuentra la torre; los dos últimos indican la posición de un determinado intruso. El primer número de cada posición indica la coordenada en el eje x , mientras que el segundo indica la coordenada en el eje y . Suponemos que la torre y el intruso se encuentran en posiciones distintas.

La entrada finaliza con una línea que contiene cuatro ceros, caso que no deberá procesarse.

Salida

Por cada caso de prueba debe escribirse una línea con el texto SI si un vigilante situado en el interior de la torre puede detectar al intruso, o con el texto NO en caso contrario.

Entrada de ejemplo

```
0 2 0 4
1 2 3 1
1 2 4 5
0 0 0 0
```

Salida de ejemplo

SI
NO
SI

Autores: Manuel Montenegro y Pedro Pablo Gómez Martín.

Revisores: Alberto Verdejo y Isabel Pita.

Anillos de plástico

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

<http://www.aceptaelreto.com/problem/statement.php?id=687>

Aunque poco a poco se van reduciendo, los anillos de plástico que se utilizan para mantener juntas las latas de refresco en grupos son una maldición para las aves y la vida marina. Confundiéndolos por alimento, muchos especímenes terminan enredándose en ellos, sufriendo de por vida las consecuencias.

Las asociaciones ecologistas llevan décadas intentando concienciar del problema. Aparte de la solución obvia de reducir su uso en las plantas de envasado, a los consumidores se nos pide que, al menos, cortemos los anillos antes de desechar el plástico en el cubo de reciclaje. De esa forma se evitarán los enredos aunque, naturalmente, se seguirá incurriendo en la contaminación causada por el propio material.

Siempre que me enfrento a un plástico de anillos, tijera en mano, me hago la misma pregunta. ¿Cuál es el menor número de cortes que tengo que hacer para eliminarlos todos?



Entrada

Cada caso de prueba comienza con un número $1 \leq n \leq 100$, indicando cuántos puntos de unión entre diferentes tiras de plástico hay en el conjunto. A continuación viene la información de esas tiras de plástico, indicando el punto de unión inicial y final que juntan (ambos números entre 1 y n).

La lista de tiras de plástico termina con dos ceros.

Ten en cuenta que a veces alguien ha empezado el trabajo por mí y ha realizado ya algunos cortes, aunque el plástico siempre formará una única pieza.

La entrada termina con un 0.

Salida

Por cada caso de prueba el programa escribirá el menor número de cortes que tenemos que realizar para evitar que la fauna se enrede en los anillos de plástico. Los puntos de unión se consideran despreciables, por lo que únicamente se pueden cortar las tiras.

Entrada de ejemplo

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

Salida de ejemplo

5
0

Autor: Pedro Pablo Gómez Martín.

Revisor: Alberto Verdejo.

Internet en el metro

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

<http://www.aceptaelreto.com/problem/statement.php?id=382>

Para fomentar el uso del transporte público, el alcalde de mi ciudad ha pedido a la empresa concesionaria del servicio de metro que instale antenas wifi en los túneles para dar acceso gratuito a Internet en toda la red. De esta forma, los ciudadanos no tendrán excusa para no utilizar este medio de transporte, ya que podrán aprovechar sus desplazamientos para navegar.

Para asegurarse de que los ciudadanos estarán satisfechos con el sistema, ha pedido a la empresa que le indique la localización de las antenas y la cobertura de cada una de ellas para poder comprobar que a nadie se le cortará la conexión.



Entrada

La entrada comienza con el número de casos de prueba. Para cada caso de prueba se indica la longitud del túnel seguida del número de antenas que se han instalado. En la línea siguiente se indica, para cada antena, la distancia a la que se encuentra del comienzo del túnel seguida de su radio de cobertura (es decir, la antena cubre esa distancia en cada uno de los sentidos).

Se garantiza que el número de antenas está entre 1 y 1.000 y tanto la longitud del túnel como los radios de cobertura están entre 1 y 10^9 , aunque ninguna antena cubrirá más allá del punto 10^9 . Además, las antenas aparecen en la entrada ordenadas por distancia al inicio del túnel.

Salida

Para cada caso de prueba el programa debe escribir en una línea distinta **SI** si el túnel está completamente cubierto o **NO** en caso contrario.

Entrada de ejemplo

```
4
1500 2
500 500 1000 500
50 2
10 10 40 10
50 3
10 5 30 5 30 25
100 3
30 30 70 5 75 25
```

Salida de ejemplo

```
SI
NO
NO
SI
```

Autor: Isabel Pita.

Revisores: Pedro Pablo Gómez Martín y Alberto Verdejo.

Temperaturas extremas

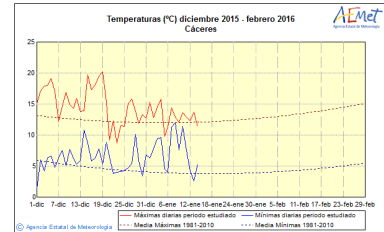
Tiempo máximo: 1,000-2,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=314>

La Agencia Estatal de Meteorología (AEMET) actualiza diariamente el Banco Nacional de Datos Climatológicos, en el que se almacenan las observaciones climatológicas (precipitación y temperatura) realizadas en España desde hace unos 150 años.

Nuria está estudiando la relación entre la variabilidad de la temperatura y el estado hídrico del suelo en una región de secano en España. Para ello, acude al Banco y solicita los registros de temperaturas en dicha zona año a año desde que existen registros.

En la primera fase de su estudio pretende determinar el número de picos y valles en las temperaturas en un determinado periodo de tiempo. Una temperatura se considera pico (resp. valle) en una secuencia cuando la anterior y la siguiente en la secuencia son estrictamente menores (resp. mayores).



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 dos líneas. La primera de ellas tiene un único entero con el número de temperaturas registradas (mayor que 0 y menor o igual que 10.000), mientras que la segunda línea contiene la secuencia de temperaturas (números enteros entre -50 y 60 grados centígrados).

Salida

Para cada caso de prueba se escribirá el número de picos y de valles, separados por un espacio y seguidos de un salto de línea.

Entrada de ejemplo

```
4
5
7 5 3 8 9
8
8 9 7 6 5 3 4 2
2
3 -5
8
4 -1 5 3 7 7 6 8
```

Salida de ejemplo

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

Autor: Clara Segura.

Revisores: Pedro Pablo Gómez Martín y Alberto Verdejo.

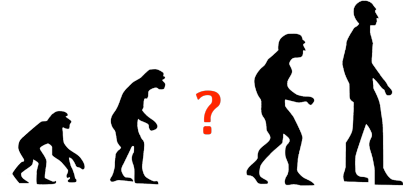
Eslabones perdidos

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

<http://www.aceptaelreto.com/problem/statement.php?id=453>

Según la teoría de la evolución, todas las especies de seres vivos han surgido como resultado de las mutaciones progresivas, a lo largo de millones de años, de las primeras bacterias. Se forma así un “árbol de especies” con el que se podría seguir la pista de todos los pasos intermedios por los que una especie ha pasado hasta llegar a ser como es.

Los registros fósiles dan muestra de muchas de esas especies. Sin embargo, hay muchas *formas transicionales* que no se han encontrado, y que a menudo son llamadas *eslabones perdidos*.



Entrada

El programa deberá procesar múltiples casos de prueba recibidos por la entrada estándar. Un caso de prueba está compuesto de una primera línea indicando el número e de especies conocidas (entre 1 y 10.000) y el número r de relaciones (diferentes) entre ellas.

A continuación vendrán r líneas, cada una con dos números distintos $1 \leq o, d \leq e$ indicando que la especie número o es considerada la predecesora directa de la especie d .

Se garantiza que cada especie tendrá, como mucho, una especie que la precede, que no se formarán ciclos y que cada relación aparecerá en la entrada una única vez.

La entrada termina con un 0.

Salida

Para cada caso de prueba, el programa escribirá “TODAS” si se conocen todas las especies de modo que es posible seguir el rastro desde cada una hasta la primera bacteria y “FALTA ALGUNA” en otro caso.

Entrada de ejemplo

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

Salida de ejemplo

```
FALTA ALGUNA
TODAS
```

Autor: Pedro Pablo Gómez Martín.

Revisores: Alberto Verdejo y Marco Antonio Gómez Martín.

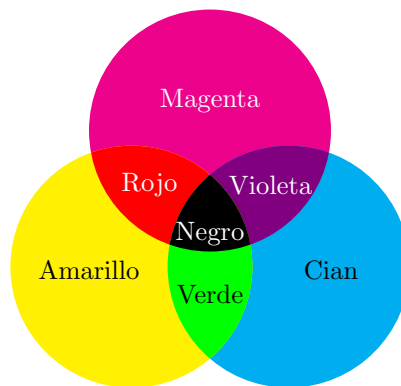
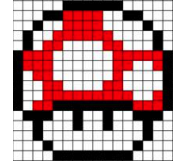
Pixel Art

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

<http://www.aceptaelreto.com/problem/statement.php?id=390>

Pixel Art es una técnica que consiste en crear imágenes a partir de pequeños cuadrados o *píxeles* que se rellenan de un único color. Se hizo muy popular en la década de los 80 debido a las limitaciones que tenían los ordenadores para representar imágenes, pero hoy en día sigue siendo popular entre los amantes de lo *retro*.

Reproducir este tipo de imágenes es muy fácil cuando se dispone de todos los colores necesarios. El reto es hacerlo usando sólo los colores primarios: magenta, amarillo y cian. Afortunadamente, la teoría del color nos dice que el resto de colores se puede obtener mezclando otros diferentes según el siguiente esquema:



Por ejemplo, el color rojo se obtiene mezclando magenta y amarillo; el negro, mezclando los 3 colores básicos; y el blanco, dejando el papel sin pintar.

Entrada

El programa deberá leer, por la entrada estándar, el número de casos de prueba que vendrán a continuación, cada uno en una línea.

Para cada caso, los primeros 3 números indican la cantidad de pintura disponible de cada color básico en este orden: magenta, amarillo y cian. Cada unidad de pintura permite pintar un píxel de ese color. Ten en cuenta que al mezclar colores se gastan unidades de todos los colores usados. Por ejemplo, pintar un píxel de rojo requiere gastar una unidad magenta y otra amarilla.

A continuación aparece la secuencia de píxeles que componen la imagen codificados con letras: magenta (M), amarillo (A), cian (C), rojo (R), negro (N), verde (V), violeta (L) y blanco (B). Todas las imágenes tendrán al menos 1 píxel, y ninguna tendrá más de 100.000.

Salida

Para cada caso de prueba el programa escribirá una línea. Si la imagen se puede completar con las pinturas disponibles, se escribirá SI y las unidades restantes de pintura magenta, amarillo y cian, en ese orden. Si la imagen no se puede completar, se escribirá NO.

Entrada de ejemplo

```
3
1 1 1 MCA
3 2 1 MABBCR
1 1 1 RL
```

Salida de ejemplo

SI	0	0	0
SI	1	0	0
NO			

Autor: Antonio Sánchez Ruiz-Granados.

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

Marketing de zapatos

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

<http://www.aceptaelreto.com/problem/statement.php?id=510>

En la zapatería donde trabajo para sacarme unas perrillas tenemos dos modelos de zapatos distintos (uno para hombre y otro para mujer) a los que prestamos especial atención porque se venden muy bien en estas fechas. Para intentar cautivar a los clientes, la responsable de *escaparatismo* (o “*visual merchandising*”, como le gusta decir a ella que es muy *cool*) ha decidido colocar un montón de ejemplares de esos dos tipos de zapatos a lo largo de la pared principal de la tienda en una fila enorme.



Hay que reconocer que tiene buen gusto, porque la forma en la que ha alternado los dos modelos crea una mezcla de color muy atrayente y las ventas han subido aún más. Pero se nota que no es ella quien se encarga de atender a los clientes, porque los zapatos no están ordenados por talla, y cada vez que tenemos que buscar una nos toca pasearnos de un lado a otro por toda la tienda.

Los vendedores queremos reordenar la fila de modo que la alternancia entre los dos modelos de zapatos se mantenga, pero, por separado, cada modelo esté ordenado por talla. Así no se perjudica al *visual merchandising*, pero son más fáciles de buscar.

Entrada

La entrada consiste en múltiples casos de prueba. Cada uno comienza con un número $1 \leq N \leq 1.000$ indicando el número de zapatos puestos en la fila de la tienda. La línea siguiente contiene N números con las tallas de todos ellos en su disposición original.

Para diferenciar entre los dos modelos, las tallas de uno de ellos se proporcionan con números positivos, y las tallas del otro con números negativos. Somos una zapatería que está dentro de la iniciativa *perfect fit* de ajuste perfecto, por lo que, en valor absoluto, las tallas son números entre 1 y 1.000.

La entrada termina con un caso sin zapatos, que no debe procesarse.

Salida

Por cada caso de prueba se escribirá la lista de tallas de zapatos reordenada de menor a mayor. Las tallas se escribirán también con valores positivos y negativos como en la entrada. Por el especial significado del signo, en la salida los números positivos estarán ordenados de menor a mayor, y los negativos de mayor a menor.

Entrada de ejemplo

```
4
32 -32 -30 30
6
50 40 -70 50 -30 -70
0
```

Salida de ejemplo

```
30 -30 -32 32
40 50 -30 50 -70 -70
```

Autor: Pedro Pablo Gómez Martín.

Revisores: Marco Antonio Gómez Martín y Alberto Verdejo.

Tensión descompensada

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

<http://www.aceptaelreto.com/problem/statement.php?id=506>

La hipertensión arterial es un mal de “nuestro tiempo” que afecta a gran parte de la población, especialmente los mayores. Y eso para mí es malo, muy malo. Como es una enfermedad muy extendida, cuando les cuento a los médicos lo mal que me siento desde que soy hipertenso no me hacen demasiado caso. Me dicen que me tranquilice, que siga el tratamiento y la dieta y que no me preocupe más.

Pero yo estoy muy alarmado con el asunto, que tan mayor no soy. Para poder controlarme la tensión regularmente, tengo un tensiómetro en cada habitación. Así, me pille donde me pille la siguiente crisis, puedo medir mi presión arterial en cuestión de segundos y ver si tengo que salir corriendo al hospital o no.

La gente que tengo alrededor dice que estoy exagerando. Incluso los médicos de urgencias ya me conocen y me dejan horas en la sala de espera antes de llamarme. Y cuando me atienden noto cierto desdén en sus palabras supuestamente tranquilizadoras.

Lo peor de todo es que mi enfermedad va a peor. He leído en “*El rincón del hipocondríaco*” que es muy malo tener la tensión descompensada, algo que ocurre cuando la “*mínima*” y la “*máxima*” están demasiado cerca. Tras leerlo me he tomado la tensión y el susto ha sido mayúsculo, tanto que he salido disparado al hospital. Y la verdad es que me he enfadado bastante al ver cómo se han reído de mí en urgencias cuando les he dicho que la tenía tan descompensada que la *mínima* estaba por encima de la *máxima*. Me han dicho algo de que igual he leído los números al revés.



Entrada

La entrada comienza con una línea que contiene el número de casos de prueba que vendrán a continuación.

Cada caso de prueba es la lectura de los dos valores de presión dadas por el tensiómetro, separadas por una barra (“/”) entre dos espacios. Ambas cifras serán mayores que cero y no superarán 250.

Salida

Por cada caso de prueba se escribirá una única línea, indicando si la lectura es correcta (BIEN) o está dada la vuelta (MAL).

Se considera correcta si el primer valor es la *máxima* y el segundo la *mínima*.

Entrada de ejemplo

```
3
120 / 70
70 / 120
120 / 120
```

Salida de ejemplo

```
BIEN
MAL
BIEN
```

Autor: Marco Antonio Gómez Martín.

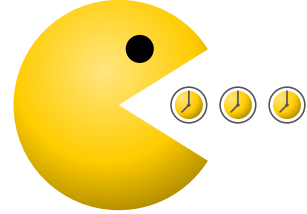
Revisores: Alberto Verdejo y Pedro Pablo Gómez Martín.

Una partidita y nos vamos a clase

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

<http://www.aceptaelreto.com/problem/statement.php?id=569>

Cuando estudiaba en la Universidad, una frase clásica que oía a menudo era “Venga, una partidita y nos ponemos a estudiar”. Este es el caso de Ladi. Ladi es un alumno responsable y, como cualquier otro, es un adicto a los videojuegos. Hasta hace poco podía aprovechar sus ratos libres para echarse unas partiditas en red mientras esperaba al comienzo de la siguiente clase. Por desgracia, recientemente ha habido cambios bastante grandes en la organización docente de su Universidad y se ha encontrado con un horario de clase mucho más ajustado que antes. No quiere perder clase, pero tampoco dejar de jugar, por lo que necesita una planificación muy meticulosa.



Como está muy ocupado preparándose los exámenes y leyendo sobre los últimos lanzamientos, te ha pedido ayuda. Llega a la Universidad a las 8:00 de la mañana, y coge el autobús de vuelta a casa a las 14:00 de la tarde. Conociendo su horario de clases, ¿cuánto tiempo puede jugar?

Entrada

La entrada está compuesta de una serie de casos de prueba. La primera línea indica cuántos hay, y la sigue la descripción de cada uno de esos casos.

Un caso se especifica con un primer número L mayor que 0, que indica la duración mínima, en minutos, de una partida del juego al que Ladi está enganchado en este momento. Si entre clase y clase no tiene al menos L minutos, no dará tiempo a terminar la partida y no tiene sentido jugar. Pero si hay tiempo suficiente, la partida puede alargarse todo el tiempo disponible.

El siguiente número N indica el número de clases, tutorías y seminarios a las que Ladi debe asistir a lo largo de un determinado día. A continuación siguen N líneas, donde se describe cada una, indicando la hora de inicio en formato HH:MM (horas y minutos) y la duración D en minutos. Se garantiza que las horas son correctas, están ordenadas cronológicamente, no hay dos actividades que se solapen y la última permite siempre coger el autobús, no terminando después de las 14:00.

Salida

Por cada caso de prueba se escribirá cuántos minutos puede dedicar Ladi a jugar ese día.

Entrada de ejemplo

```
2
30 3
08:30 60
09:30 60
10:30 95
45 3
08:30 30
09:45 46
11:15 90
```

Salida de ejemplo

```
145
120
```

Autores: Vladyslav Lyeuta y Pedro Pablo Gómez Martín.

Revisor: Alberto Verdejo.

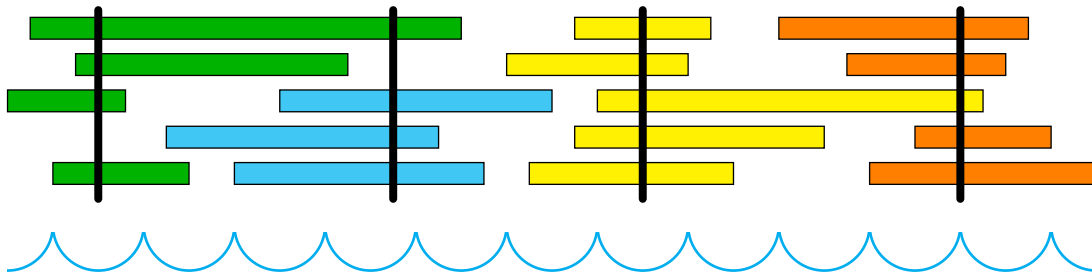
¡En primera línea de playa!

Tiempo máximo: 2,000-3,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=329>

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 subterráneos (perpendiculares a la playa) que conecten todos los edificios con la arena. Así creen que los clientes estarán más satisfechos.

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

Autor: Alberto Verdejo.

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

Recorriendo el archipiélago en bicicleta

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

<http://www.aceptaelreto.com/problem/statement.php?id=645>

Al archipiélago *Milislotes* han llegado las bicicletas. Sus habitantes están estusiasmados y no se bajan de la bici en todo el día, yendo de un lado para otro sin descanso. Pero las islas son tan pequeñas que terminan todos mareados de dar vueltas sin parar, por lo que se están planteando crear una red de puentes rectos que les permita ir en bici de cualquier isla a cualquier otra.



Han pedido presupuesto al arquitecto, que ha confeccionado una lista con todos los puentes que podrían construirse entre islas del archipiélago y cuánto costaría construir cada uno de esos puentes. ¿Podrías ayudarles a decidir qué puentes construir de tal forma que se pueda ir en bici desde cualquier isla a cualquier otra y el coste total de la obra sea lo mínimo posible?

Entrada

La entrada está formada por varios casos de prueba. Cada uno ocupa varias líneas: en la primera aparece el número I de islas en el archipiélago (entre 1 y 1.000); en la siguiente aparece el número P de puentes presupuestados (entre 0 y 10.000); y a continuación aparece una línea por cada uno de estos puentes con tres enteros, las islas que une (numeradas entre 1 e I) y el coste de construir ese puente (un valor entre 1 y 100.000). Los puentes siempre van de una isla a otra distinta, son transitables en bici en ambos sentidos, y no se ha presupuestado más de un puente entre un mismo par de islas.

Salida

Para cada caso de prueba se escribirá, en una línea, el coste mínimo de construir los puentes necesarios para unir todas las islas por bici. Si no hubiera suficientes puentes presupuestados para lograrlo, se escribirá `No hay puentes suficientes`.

Entrada de ejemplo

```
4
5
1 2 5
1 3 10
2 4 7
1 4 8
3 4 2
4
3
1 2 3
2 4 5
4 1 8
```

Salida de ejemplo

```
14
No hay puentes suficientes
```

Autor: Alberto Verdejo.

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