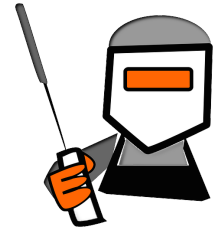


60

Varillas

Tenemos N varillas de longitudes l_1, \dots, l_N y precios c_1, \dots, c_N enteros, que no se pueden cortar. Se desea soldar algunas de ellas para obtener una varilla de longitud total L . Y estamos interesados en resolver cada uno de los siguientes problemas:



1. Indicar si es posible o no obtener la varilla deseada soldando algunas de las varillas dadas.
2. Calcular el número total de maneras de obtener la varilla deseada soldando algunas de las varillas dadas, sin que importe el orden de soldadura.
3. Calcular el número mínimo de varillas necesarias para obtener la varilla deseada.
4. Calcular el mínimo coste posible necesario para obtener la varilla deseada.

Entrada

La entrada consta de una serie de casos de prueba. Para cada caso, primero aparece el número N (entre 1 y 1.000) de varillas y la longitud L de la varilla a formar (entre 1 y 1.000). A continuación aparecen N líneas con la descripción de cada varilla: su longitud y su coste (todos ellos números entre 1 y 1.000).

Salida

Para cada caso de prueba, si es posible formar la varilla deseada, se escribirá **SI** seguido de las respuestas a los otros tres problemas: el número total de maneras de obtener tal varilla, el número mínimo de varillas a utilizar y el mínimo coste necesario. Si no es posible formar la varilla, se escribirá simplemente **NO**.

Entrada de ejemplo

```
4 50
10 25
20 15
30 40
40 75
3 10
1 30
2 60
3 90
```

Salida de ejemplo

```
SI 2 2 55
NO
```

Autor: Alberto Verdejo.

61

El cazatesoros

Marcos es cazatesoros. Acaba de localizar un barco pirata que se hundió hace siglos y repartió a su alrededor cofres llenos de monedas de oro. Un sofisticado sistema de sonar le ha permitido identificar la posición, la profundidad y la cantidad de oro de cada uno de estos cofres. Por desgracia, Marcos solamente dispone de una botella de aire comprimido para realizar las inmersiones de recuperación. Para más inri, olvidó traer su GPS por lo que no puede volver a puerto a por más botellas porque las posibilidades de volver a encontrar el pecio serían casi nulas.



Marcos quiere bucear y recuperar todo el oro que sea posible. Sabe que la botella le permitirá estar debajo del agua T segundos; que en cada inmersión solamente podrá subir uno de los cofres; y que el tiempo de descenso a una profundidad p es de p segundos mientras que el tiempo de ascenso es de $2p$ segundos.

¿Puedes ayudarle a decidir cuáles son los cofres que debe recoger para maximizar la cantidad de oro recuperada?

Entrada

La entrada está compuesta por diversos casos de prueba. Para cada uno, la primera línea contiene el valor T , los segundos que permite la botella estar debajo del agua (un entero entre 1 y 10.000). La segunda línea contiene el número N de cofres encontrados (un entero entre 1 y 100). A continuación, aparecen N líneas cada una con dos enteros, que representan la profundidad a la que se encuentra (un entero entre 1 y 500) y la cantidad de oro que contiene (un entero entre 1 y 10.000) cada uno de los cofres.

Salida

Para cada caso de prueba, primero se escribirá una línea con la máxima cantidad de oro que se puede recuperar. Después se escribirá otra línea con el número de cofres a recoger, seguida de una línea por cada uno de ellos con la información de cada uno: profundidad y cantidad de oro. Los cofres deben presentarse en el mismo orden que aparecen en la entrada. Se garantiza que la solución óptima es única.

Después de la salida de cada caso se escribirá ----.

Entrada de ejemplo

```
210
3
40 5
40 1
25 2
200
5
25 4
50 5
40 4
10 2
70 10
29
1
10 20
```

Salida de ejemplo

```
7
2
40 5
25 2
----
8
2
25 4
40 4
----
0
0
----
```

Autor: Alberto Verdejo.

62

Juego de tablero

Queremos jugar a un juego en el que tenemos un tablero $N \times N$ donde en cada casilla aparece un número natural. El juego consiste en elegir una casilla de la última fila (la N) y movernos desde ella hasta una casilla de la primera fila (la 1), donde los únicos movimientos legales consisten en moverse, sin salirse del tablero, de una casilla a una de las tres casillas de la fila superior alcanzables en vertical o en diagonal (a izquierda o derecha). La cantidad ganada será la suma de los valores en las casillas del tablero por las que pasamos en este recorrido.

Por ejemplo, para conseguir la mayor suma en el siguiente tablero, tendríamos que comenzar en la casilla (4,2), y después pasar por las casillas coloreadas, obteniendo una ganancia total de 30.

	1	2	3	4
1	2	5	8	3
2	1	4	2	9
3	9	2	8	5
4	3	5	2	1

Dado un tablero, queremos saber cuál es la máxima cantidad que se puede conseguir y cuál es la casilla por la que habría que comenzar.

Entrada

La entrada está formada por una serie de casos de prueba. Para cada caso, en la primera línea aparece un número N (entre 3 y 500) indicando la dimensión del tablero. Luego aparecen N líneas, cada una con N números (entre 1 y 10.000) separados por espacios, que representan el contenido del tablero.

Salida

Para cada caso de prueba, el programa escribirá una línea que contendrá el máximo valor que se puede obtener y la columna (numeradas de 1 a N) de la celda de la última fila desde donde hay que comenzar el recorrido. En caso de que haya varias preferimos la que esté más a la izquierda (con un número menor).

Entrada de ejemplo

```
4
2 5 8 3
1 4 2 9
9 2 8 5
3 5 2 1
3
1 1 1
1 1 1
1 1 1
```

Salida de ejemplo

```
30 2
3 1
```

Autor: Alberto Verdejo.

63

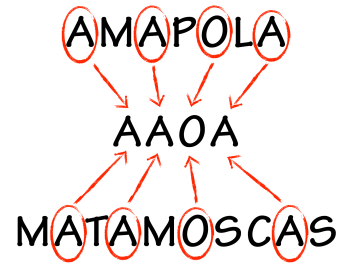
Subsecuencia común más larga

Si a una secuencia X de elementos (pongamos por ejemplo, caracteres) le quitamos algunos de ellos y dejamos los que quedan en el orden en el que aparecían originalmente tenemos lo que se llama una *subsecuencia* de X . Por ejemplo, **AAOA** es una subsecuencia de la secuencia **AMAPOLA**.

El término también se aplica cuando quitamos todos los elementos (es decir la secuencia vacía es siempre subsecuencia de cualquier secuencia) o cuando no quitamos ninguno (lo que significa que cualquier secuencia es siempre subsecuencia de sí misma).

Dadas dos secuencias X e Y , decimos que Z es una *subsecuencia común* de X e Y si Z es subsecuencia de X y de Y . Por ejemplo, si $X = \text{AMAPOLA}$ e $Y = \text{MATAMOSCAS}$, la secuencia **AAOA** es una de las subsecuencias comunes a X e Y *más larga*, con longitud 4, ya que no hay ninguna subsecuencia común a X e Y de longitud mayor que 4. También son subsecuencias comunes de longitud 4 **MAOA** o **AMOA**.

Lo que queremos es encontrar la longitud de las subsecuencias comunes más largas de dos secuencias de caracteres dadas.



Entrada

La entrada está compuesta por diversos casos de prueba, estando cada uno de ellos formado por una línea en la que aparecen dos cadenas no vacías (de no más de 1.000 caracteres de la A a la Z) separadas por un espacio.

Salida

Para cada caso de prueba se debe escribir la longitud de las subsecuencias comunes más largas de las dos cadenas leídas.

Entrada de ejemplo

```
AMAPOLA MATAMOSCAS
AAA BBBB
ATAMOS MATAMOSCAS
```

Salida de ejemplo

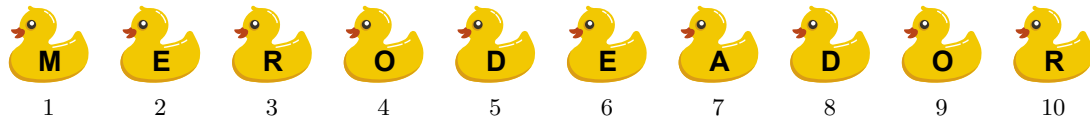
```
4
0
6
```

Autores: Alberto Verdejo y Marco Antonio Gómez Martín.

64

Tiro al patíndromo

Ya es tradición que cuando en el pueblo de Juan Filloy¹ 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
```

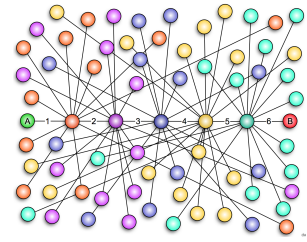
Autor: Alberto Verdejo.

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

65

Seis grados de separación

¿Sabías que entre tú y Will Smith solamente hay cinco personas que de alguna manera te relacionan con él? Al menos eso es lo que dice la *Teoría de los seis grados de separación*. Según esta teoría cualquier persona en la Tierra puede estar conectada con cualquier otra a través de una cadena de conocidos que no tiene más de cinco intermediarios (conectando a ambas personas con solo seis enlaces), algo que se ve representado en la popular frase “*el mundo es un pañuelo*”. La teoría fue inicialmente propuesta en 1930 por el escritor húngaro Frigyes Karinthy en un cuento llamado *Cadenas*.



El concepto está basado en la idea de que el número de conocidos crece exponencialmente con el número de enlaces en la cadena, y solo se necesita un pequeño número de enlaces para que el conjunto de conocidos se convierta en la población humana entera.

Con la aparición de las redes sociales en Internet y los conocidos virtuales este grado de separación se ha ido reduciendo. Un estudio realizado por Facebook en 2011 concluyó que prácticamente la totalidad de pares de sus usuarios (un diez por cien de la población mundial) estaba conectado con cinco grados de separación.

Nosotros queremos averiguar el *grado de separación* de una red de personas. Para dos personas cualesquiera, su grado de separación es el número mínimo de relaciones que hay que atravesar para conectarlas. Para una red, el grado de separación es el máximo grado de separación entre dos personas cualesquiera de la red. Si existe un par de personas que no están conectadas por una cadena de relaciones decimos que la red está desconectada.

Entrada

La entrada consiste en una serie de casos de prueba que representan redes de personas. La primera línea de cada caso contiene dos números: el número P de personas en la red ($1 \leq P \leq 500$) y el número R de relaciones directas ($0 \leq R \leq 10.000$). Las siguientes R líneas contienen una relación cada una, formada por el nombre de dos personas en la red. Los nombres de personas son únicos y están formados por como mucho 10 letras del alfabeto inglés (sin espacios). Pero el nombre de una persona puede aparecer en varias relaciones, indicando que esa persona está relacionada con varias otras.

Salida

Para cada red de personas el programa debe mostrar el grado de separación de la red en una línea, a no ser que la red esté desconectada, en cuyo caso se mostrará la palabra **DESCONECTADA** en una línea.

Entrada de ejemplo

```
4 4
Marta Mario
Mario Roberto
Roberto Raquel
Mario Raquel
4 2
Roberto Mario
Marta Raquel
```

Salida de ejemplo

```
2
DESCONECTADA
```

Autor: Alberto Verdejo.

66

El carpintero Ebanisto

El carpintero *Ebanisto* ha recibido el encargo de cortar un tablón en varios trozos que han sido previamente marcados sobre la madera. El esfuerzo de cortar un tablón de madera en dos es el doble de su longitud.

Ebanisto se ha dado cuenta de que el orden en el que realice los cortes en el tablón influye en el esfuerzo empleado. Por ejemplo, supongamos que un tablón de 10 metros de longitud tiene que cortarse a 3, 6 y 8 metros de uno de los extremos. Una posibilidad sería cortar primero por la marca de los 3 metros, luego por la marca de los 6 metros y finalmente por la de 8 metros, lo que le costaría a Ebanisto un esfuerzo total de $2 * 10 + 2 * 7 + 2 * 4 = 42$. Sin embargo, si corta primero por la marca del 6, después por la del 3 y finalmente por la del 8, entonces le costaría un esfuerzo de $2 * 10 + 2 * 6 + 2 * 4 = 40$.

¿Puedes ayudar a Ebanisto a averiguar en qué orden cortar el tablón por las marcas para minimizar el esfuerzo realizado?



Entrada

La entrada constará de varios casos de prueba. La primera línea de cada caso contendrá dos números positivos: L (entre 10 y 1.000.000), que representa la longitud del tablón que debemos cortar; y N (entre 1 y 500), que indica el número de cortes que se deben realizar. La siguiente línea contendrá N números positivos c_i ($0 < c_i < L$), que determinan los puntos en los que se deben realizar los cortes, dados en orden creciente.

La entrada termina con 0 0.

Salida

Para cada caso de prueba se debe escribir el mínimo esfuerzo que debe realizar Ebanisto para realizar todos los cortes.

Entrada de ejemplo

```
10 3
3 6 8
20 4
8 10 15 17
0 0
```

Salida de ejemplo

```
40
88
```

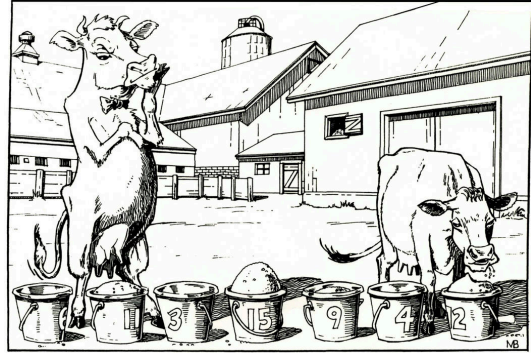
Autor: Alberto Verdejo.

67

Las vacas pensantes

Los humanos no tienen ni idea de en qué pensamos las vacas. Nos veis descansando a la sombra de un árbol, sin preocuparnos de los problemas del mundo y suponéis que no tenemos nada entre oreja y oreja. Pero estáis equivocados, pensamos mucho en los problemas que nos importan, como cuándo será la siguiente comida.

A mi jefe, el granjero Sancho, le gusta ponernos retos. Antes de ordeñarnos, coloca una hilera de cubos con diferentes cantidades de comida y elige dos vacas. Estas deben alternarse para comer y cada una, en su turno, debe elegir uno de los cubos de los extremos, comerse su contenido y retirarlo. Así hasta que se acaban los cubos.



Cuando me emparejan con mi compañera *Devoradora* ella siempre sigue la misma estrategia: comer el cubo de los extremos que tenga mayor cantidad. Yo solía elegir al azar, pero con ella suelo quedarme hambrienta. Y aunque el hambre puede sacar lo mejor de nosotros, no quiero que vuelva a ocurrir. ¿Puedes ayudarme a calcular cuánto puedo comer como máximo si me toca empezar a comer y me enfrento a *Devoradora*?

Entrada

La entrada estará compuesta por varios casos de prueba, cada uno ocupando dos líneas. La primera línea contiene el número N (entre 1 y 1.000) de cubos. La segunda línea contiene N enteros diferentes entre 1 y 10.000, que representan las cantidades de comida en los N cubos según están colocados de izquierda a derecha.

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

Salida

Para cada caso de prueba se escribirá una línea con la cantidad máxima que puede comer nuestra amiga si le toca empezar a comer y se enfrenta a *Devoradora*.

Entrada de ejemplo

```
4
8 7 1 4
4
2 4 15 5
7
6 11 3 15 9 4 12
0
```

Salida de ejemplo

```
12
17
35
```

Autor: Alberto Verdejo.

Mejor no llevar muchas monedas

Mario tiene un cofre lleno de monedas que le han ido dando sus abuelos cuando les visita. Hace poco se ha pasado por su tienda favorita de juguetes y ha visto un coche teledirigido que le ha encantado. Preguntó el precio y ahora quiere saber si tiene monedas suficientes para poder comprárselo. Como tendrá que llevar las monedas en el bolsillo y en el camino pasará por una zona poco recomendable, quiere que se note lo menos posible que lleva ahí el dinero, por lo que quiere pagar con el menor número de monedas.



Mario ha clasificado las monedas por su valor y ha contado cuántas monedas tiene de cada tipo. ¿Puedes ayudarlo a averiguar si puede pagar *de forma exacta* el precio del coche y cuántas monedas necesita llevar como mínimo?

Entrada

La entrada consta de una serie de casos de prueba. Para cada caso, primero aparece el número N (entre 1 y 100) de tipos diferentes de monedas que Mario tiene. A continuación aparecen dos líneas con N enteros cada una: la primera con los valores de las monedas de cada tipo y la segunda con la cantidad de monedas que tiene de cada uno de esos tipos, en el mismo orden (todos ellos números entre 1 y 1.000). Por último, aparece una línea con el precio del coche (un número entre 1 y 10.000).

Salida

Para cada caso de prueba se escribirá una línea que comience por **SI** seguido del mínimo número de monedas necesarias, si es posible pagar el precio del coche; o que contenga la palabra **NO**, en caso contrario.

Entrada de ejemplo

```
4
1 5 10 50
10 2 5 4
260
3
1 10 100
3 2 2
114
3
10 15 5
2 2 2
20
```

Salida de ejemplo

```
SI 11
NO
SI 2
```

Autor: Alberto Verdejo.

Cine romántico a raudales

A *Dynamique Cinema* no le gustan las películas de terror tanto como a su hermana Deborah; ella prefiere el apasionado cine romántico. Y está de suerte, porque aprovechando el comienzo de la primavera, la filmoteca ha organizado un maratón de cine romántico: durante 24 horas se proyectarán películas (todas diferentes) en las diversas salas disponibles.

Dynamique ya se ha hecho con el folleto con la programación completa donde aparecen todas las películas que se van a proyectar durante el maratón; junto con el título, nombre del director, sala de proyección y otros datos de interés, se indica la hora de comienzo y duración de la película.

¿Puedes ayudar a Dynamique a planificar su maratón de cine, teniendo en cuenta que su único objetivo es estar viendo películas durante el mayor tiempo posible?



Entrada

La entrada consta de una serie de casos de prueba. Cada uno comienza con una línea con el número N de películas que se proyectarán ($0 < N \leq 1.000$). A continuación aparecerán N líneas con la información de cada película: la hora de comienzo dentro del día de proyección, en el formato HH:MM, y la duración en minutos de la película. Ninguna película acabará más allá de las 12 de la noche.

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

Salida

Para cada caso de prueba se escribirá una línea con el máximo tiempo, medido en minutos, que Dynamique Cinema puede estar viendo películas, suponiendo que siempre necesita 10 minutos libres (para comprar palomitas, cambiar de sala, etc.) entre película y película.

Entrada de ejemplo

```
3
11:00 90
12:30 90
12:45 60
3
11:00 90
12:45 60
11:00 180
2
12:00 80
20:00 80
0
```

Salida de ejemplo

```
150
180
160
```

Autor: Alberto Verdejo.