

Preparación

**ADA BYRON**

**MADRID**

**GRANDMASTER**



**Competitive Programming**

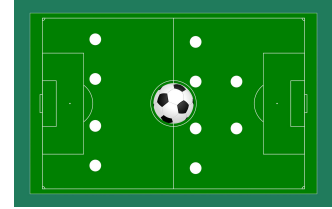
UPV

# Alineación de fútbol

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

<http://www.aceptaelreto.com/problem/statement.php?id=651>

Si dejamos de lado los partidos de fútbol en los patios de colegio donde todos los niños corren sin control detrás del balón, nos encontramos normalmente equipos organizados en donde cada jugador ocupa un puesto concreto. Aunque existen combinaciones imaginativas, lo habitual es hablar de tres líneas de jugadores (además del portero): defensas, medios y delanteros. Una de las labores del entrenador antes del partido es decidir cuántos irán a cada línea. Así, por ejemplo, se puede ver la distribución 4 4 2 en la que hay 4 defensas, 4 medios y 2 delanteros.



Los jugadores, por su parte, se sienten cómodos solo en algunas demarcaciones. Existen jugadores que son buenos solo cuando juegan como defensas, otros que valen como medios o delanteros y algunos que pueden colocarse en cualquiera de las tres líneas.

Teniendo en cuenta las habilidades de la plantilla, el entrenador podrá (o no) utilizar ciertas distribuciones concretas.

## Entrada

La entrada está compuesta de distintos casos de prueba, cada uno ocupando tres líneas.

La primera contiene tres números: el número de jugadores que se colocarán como defensas, como medios y como delanteros respectivamente. Algunos de esos números pueden ser cero pero la suma de todos ellos no excederá 10.

A continuación aparece una línea con el número de jugadores que hay en la plantilla (como mucho 25). La última línea contiene la descripción de cada uno de los jugadores, entendiendo ésta como las posiciones en las que ese jugador puede colocarse. La descripción consiste en una o más letras seguidas con el siguiente significado: F para defensa, C para centro y D para delantero.

Se garantiza que el número de jugadores en plantilla es mayor o igual que el número de jugadores que deben saltar al campo.

La entrada termina con una línea con tres ceros que no debe procesarse.

## Salida

Por cada caso de prueba se escribirá una línea independiente en la que SI indicará que existe al menos una asignación de jugadores a puestos en la que todos tengan una demarcación favorable y NO en caso contrario.

## Entrada de ejemplo

```
4 3 3
10
F C D FC DF DCF CD DC DC FD
3 3 3
11
F F F F C C C C D D F
0 0 0
```

## Salida de ejemplo

```
SI
NO
```

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

## Problema número 492

# ¡Mío!

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

<http://www.aceptaelreto.com/problem/statement.php?id=492>

Marta y Daniel están aprendiendo las tablas de multiplicar y juegan con su padre al siguiente juego. Cada uno de ellos elige un número (por ejemplo Marta elige el 3 y Daniel el 5) y su padre empieza a contar desde 1 hasta que se cansa. Cada vez que el padre dice un número múltiplo del número elegido por uno de los niños, ese niño debe gritar: *¡Mío!*

A continuación se muestra un ejemplo del juego:

```
Uno...
Dos...
Tres... Mio (Marta)
Cuatro...
Cinco... Mio (Daniel)
Seis... Mio (Marta)
Siete...
```

A los niños les hace mucha gracia cuando los dos gritan *¡Mío!* a la vez (por ejemplo con el 15 en el caso anterior), y se preguntan cuántas veces ocurrirá para valores arbitrarios de  $M$  (el número elegido por Marta),  $D$  (el número elegido por Daniel) y  $N$  (el número hasta el que cuenta el padre).

### Entrada

La entrada está formada por distintos casos de prueba, cada uno en una línea diferente. Cada caso de prueba consiste en tres números:  $M$ , el número elegido por Marta;  $D$ , el número elegido por Daniel; y  $N$ , el número hasta el que cuenta el padre (un entero positivo menor que  $10^9$ ). Los números  $M$  y  $D$  son enteros positivos menores o iguales que  $N$ . 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 escribirá una línea con el número de veces que los dos niños gritan a la vez *¡Mío!*

### Entrada de ejemplo

```
3 5 100
5 12 100
2 4 16
0 0 0
```

### Salida de ejemplo

```
6
1
4
```

**Autor:** Luis Fernando Lago Fernández.

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

# Coge el sobre y corre

Tiempo máximo: 2,000-6,000 s Memoria máxima: 10240 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=258>

Pedro Franqueza fue nombrado tesorero hace unos años. Desde entonces sus distintos *chanchullos* le han hecho una persona muy influyente. En la cajonera que tiene debajo de su mesa guarda una hilera de sobres, cada uno con una cantidad de dinero conseguido de manera dudosa.



Hoy tiene una cena de negocios en el bar *Cenás* a la que irá el presidente y otros nueve compañeros. Para mantenerlos contentos, justo antes de salir del despacho mete la mano en el cajón y coge diez sobres consecutivos para repartirlos allí mismo. No ha tenido tiempo de mirar cuánto dinero hay en cada sobre, así que habrá que repartirlos sobre la marcha. Eso sí, el presidente se quedará con el sobre que más dinero tenga.

En el coche yendo hacia el bar va pensando en cómo podría averiguar rápidamente cuánto dinero le correspondería al presidente en base al grupo de 10 sobres consecutivos seleccionados. Y como tiene aires de grandeza, se plantea si sería capaz de hacerlo si guardara hasta 500.000 sobres en su cajonera.

## Entrada

La entrada está compuesta por distintos casos de prueba que representan días distintos en la vida de Pedro Franqueza.

Cada caso aparece en dos líneas consecutivas. La primera de ellas contiene dos enteros, el primero con el número de sobres que guarda en el cajón ( $1 \leq n \leq 500.000$ ) y el segundo el número de sobres que tiene que coger para ir a la cena, o lo que es lo mismo, el número de comensales que se llevarán un sobre, incluido el presidente ( $1 \leq k \leq \min(n, 200.000)$ ). La segunda línea contiene  $n$  números mayores o iguales a 1 con el dinero que hay en cada uno de los sobres.

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

## Salida

Para cada caso de prueba se escribirá una única línea con  $n - k + 1$  números que indicarán la cantidad de dinero que se llevará esa noche el presidente dependiendo de qué sobres coja Pedro. En particular, el primer número será el dinero para el presidente si coge los  $k$  primeros sobres; el segundo si se salta el primer sobre y coge los  $k$  siguientes, etc. En general, el número  $i$ -ésimo indica la cantidad que se llevará el presidente si coge los  $k$  sobres desde el sobre  $i$  hasta el sobre  $i + k - 1$ .

## Entrada de ejemplo

```
6 1
3 8 5 12 15 9
6 3
3 8 5 12 15 9
0 0
```

## Salida de ejemplo

```
3 8 5 12 15 9
8 12 15 15
```

**Autor:** Marco Antonio Gómez Martín.

**Revisores:** Pedro Pablo Gómez Martín y Alberto Verdejo.

# Cargando el móvil

Tiempo máximo: 2,000-4,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=509>

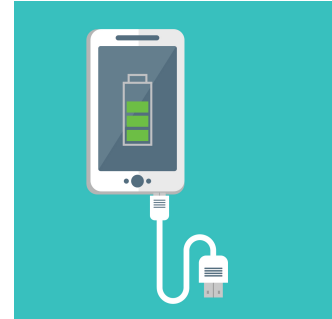
La dependencia que tenemos con los móviles nos hace cargarlos todas las noches hasta el nivel máximo de la batería para no correr el riesgo de quedarnos sin él a mitad del día siguiente. Hay incluso gente que cuando durante el día ve un enchufe libre lo pone a cargar por lo que pueda pasar.

Yo soy uno de ellos.

Con los años he ido refinando mi técnica para no cargar la batería al máximo todas las noches. Con mis rutinas repetitivas perfectamente conocidas, tengo claro en qué momentos del día voy a poder hacer esas *recargas parciales* (y en cuántas unidades se incrementará la carga) y lo que me baja el nivel de batería entre cada una. Con esos datos, todas las noches calculo el nivel mínimo de batería con el que tengo que salir de casa al día siguiente para que éste nunca baje de mi umbral auto-impuesto de dos unidades.

Lo que aún no he conseguido dominar es ese mismo cálculo cuando salgo de la rutina establecida y tengo varias alternativas para hacer las cosas. Me sucede por ejemplo los días en los que estoy en ruta a otra ciudad a la que puedo llegar de distintas formas.

En mi primera aproximación al problema estoy suponiendo que quiero desplazarme por un “tablero de ajedrez”, desde la esquina superior izquierda a la esquina inferior derecha. En cada “celda” o bien puedo cargar el móvil una cantidad concreta o bien no puedo y su nivel de carga desciende. Teniendo en cuenta que quiero ir por un camino que atraviase el mínimo número de celdas posible, ¿cuál es la carga mínima con la que tengo que salir para que el móvil nunca baje de esas dos unidades de carga?



## Entrada

La entrada esta compuesta por distintos casos de prueba.

Cada caso de prueba comienza con una línea que contiene dos números: el número  $C$  de columnas y el número  $F$  de filas del tablero (ninguno de ellos supera las 100 unidades).

A continuación aparecen  $F$  líneas, una por cada fila. Cada una de ellas contiene  $C$  números, indicando la variación del nivel de carga que sufre el móvil al pasar por cada casilla de esa fila. Un número positivo indica cuánto puedo cargar en ella el móvil antes de tener que seguir mi camino, mientras que uno negativo indica que no hay enchufes y que la batería del móvil desciende su nivel de carga en esa cantidad. Se garantiza que las cantidades en las celdas origen y destino (esquina superior izquierda e inferior derecha) son siempre 0 y que los valores del resto no superan (en valor absoluto)  $10^6$ .

## Salida

Por cada caso de prueba se escribirá una línea con un único número: el nivel de batería con el que hay que salir para garantizar que podré llegar desde la esquina origen a la destino por el camino más corto que seleccionemos sin que el nivel de carga baje nunca de 2 unidades.

## Entrada de ejemplo

```
5 1
0 -1 -1 -1 0
5 1
0 1 -1 -1 0
5 1
0 -1 -1 1 0
2 2
0 -3
-5 0
```

## Salida de ejemplo

5
3
4
5

**Autor:** Marco Antonio Gómez Martín.

**Revisores:** Pedro Pablo Gómez Martín, Alberto Verdejo y Manuel Montenegro.

# La carrera hacia la Casa Blanca

Tiempo máximo: 1,000-4,000 s Memoria máxima: 8192 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=573>

A Donald se le ha metido en la cabeza ser presidente de Estados Unidos. Ha mirado las normas y las cumple: es ciudadano del país por nacimiento, tiene al menos 35 años (¡la primera vez que le pintaron fue en 1934!) y lleva siendo residente al menos 14 años. Así que ha elegido un partido político y se ha puesto a analizar sus posibilidades.

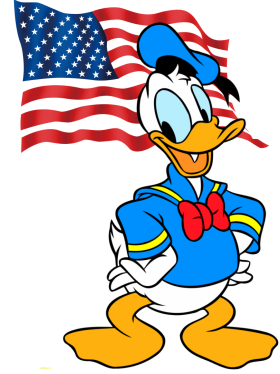
Las reglas son bastante conocidas. El candidato del partido es elegido en la convención nacional que tiene lugar el verano del año electoral. En esa convención, los *delegados* que vienen de los distintos estados votan entre los distintos aspirantes; el que tenga más votos se convierte en el candidato del partido.

Los delegados, eso sí, se rigen por unas normas claras: los  $K$  delegados de un estado concreto apoyarán al mismo candidato, aquel que resultó ganador en las votaciones *primarias* que se hicieron en su estado durante los primeros meses del año y en el que pueden votar todos los afiliados al partido.

El objetivo de Donald ahora está en las primarias de cada estado; tiene que hacer una buena campaña para garantizar que en la convención nacional terminará ganando al otro aspirante del partido.

En su análisis de la situación ha visto que en cada estado hay tres tipos de personas: aquellas que le votarán, aquellas que no le votarán y un grupo que aún está indeciso. Evidentemente la campaña de Donald debe centrarse en esos indecisos. ¿A cuántos debe convencer como mínimo para garantizarse que será el próximo candidato a la Casa Blanca?

Ten en cuenta que al ser un pato, en caso de empate siempre sale perdiendo. Si en un estado hay empate, los delegados votarán al otro candidato. Y si en la votación de la convención nacional hay empate a votos, también perderá la carrera hacia la Casa Blanca.



## Entrada

La entrada estará compuesta por distintos casos de prueba.

Cada caso comienza con una línea indicando el número  $N$  de estados del país ( $1 \leq N \leq 100$ ). A continuación vendrán  $N$  líneas describiendo cada estado con cuatro números: número de delegados que aporta (al menos uno), cantidad de gente que sabemos que votará a Donald, cantidad de gente que votará al otro aspirante y, por último, número de indecisos.

El número total de delegados que van a la convención nacional no supera los 5.000, y el número de votantes por estado nunca será mayor de diez millones.

## Salida

Por cada caso de prueba se escribirá una única línea indicando a cuántos indecisos debe convencer como mínimo Donald para asegurarse ser el candidato a la presidencia. Si los datos reflejan que no podrá ser el candidato por mucho que invierta, se escribirá **IMPOSIBLE**.

## Entrada de ejemplo

```
3
8 1000 500 0
8 500 1000 0
1 100 100 25
1
8 500 1000 500
1
8 0 0 100
```

## Salida de ejemplo

13
IMPOSIBLE
51

**Autor:** Marco Antonio Gómez Martín.

**Revisores:** Alberto Verdejo y Pedro Pablo Gómez Martín.



# Haciendo largos

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

<http://www.aceptaelreto.com/problem/statement.php?id=557>

El médico me ha recomendado nadar para cuidar mi espalda, así que desde esta semana estoy yendo a la piscina. Al ser mis primeros días estoy haciendo series cortas, de sólo dos largos, y descanso un rato entre series. Pero como estoy en muy baja forma, necesito hacer los descansos cada vez más largos. Así, en mi primer día de natación mi primer descanso fue de un minuto, el segundo de un minuto y medio, el tercero de dos minutos y así sucesivamente. Cada descanso duró 30 segundos más que el anterior y, como todos los largos los hice en el mismo tiempo (30 segundos), tardé 37 minutos en nadar 20 largos.



Así las cosas, me pregunto de manera general cuánto tardaría en hacer un determinado número de largos en función del tiempo por largo, el número de largos por serie y los tiempos de descanso entre series.

## Entrada

La primera línea de la entrada contiene un número que indica cuántos casos de prueba habrá que procesar.

Cada caso de prueba consiste en una línea con 5 números enteros:  $N$ ,  $T$ ,  $M$ ,  $D$  e  $I$ . El primer entero,  $N$ , es el número de largos que quiero nadar. El segundo número,  $T$ , es el tiempo en segundos que tardo en hacer un largo. El tercer entero,  $M$ , es el número de largos que hago en cada serie. Finalmente  $D$  indica el tiempo en segundos del primer descanso entre series, e  $I$  es la cantidad extra de descanso, en segundos, que necesito con cada serie adicional. Nótese que la última serie no tiene por qué estar completa.

Se satisfacen los siguientes límites:  $1 \leq N \leq 100.000.000$ ,  $20 \leq T \leq 40$ ,  $1 \leq M \leq N$  y  $0 \leq D, I \leq 60$ .

## Salida

Para cada caso de prueba, se escribirá una línea con el tiempo en segundos necesario para hacer los largos.

## Entrada de ejemplo

```
2
20 30 2 60 30
20 30 2 60 60
```

## Salida de ejemplo

```
2220
3300
```

**Autor:** Luis Fernando Lago Fernández.

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

# Haciendo trampas en Serpientes y Escaleras

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

<http://www.aceptaelreto.com/problem/statement.php?id=253>

*Serpientes y Escaleras* es un juego clásico, originario de la India, donde ya se jugaba en el siglo XVI. El tablero está formado por una cuadrícula de  $N \times N$  casillas numeradas de forma consecutiva desde 1 hasta  $N^2$ , comenzando por la esquina inferior izquierda y continuando fila por fila de abajo a arriba, alternando en cada fila el ir hacia la izquierda o hacia la derecha, como aparece en el dibujo. Algunos pares de casillas, siempre en filas diferentes, pueden estar conectados mediante *serpientes* (que bajan, naranjas en el dibujo) o *escaleras* (que suben, azules en el dibujo). Cada casilla puede ser extremo de como mucho una serpiente o una escalera. La primera y la última casilla nunca son extremos de una serpiente o escalera.

100	99	98	97	96	95	94	93	92	91
81	82	83	84	85	86	87	88	89	90
80	79	78	77	76	75	74	73	72	71
61	62	63	64	65	66	67	68	69	70
60	59	58	57	56	55	54	53	52	51
41	42	43	44	45	46	47	48	49	50
40	39	38	37	36	35	34	33	32	31
21	22	23	24	25	26	27	28	29	30
20	19	18	17	16	15	14	13	12	11
1	2	3	4	5	6	7	8	9	10

A *Serpientes y Escaleras* pueden jugar cualquier número de jugadores correspondiéndole a cada uno una ficha. Todas las fichas comienzan en la casilla número 1. Los jugadores van alternándose para mover su ficha. Para ello, tiran un dado con  $K$  caras numeradas desde 1 hasta  $K$ , y avanzan su ficha siguiendo la numeración del tablero tantas casillas como indique el dado. Si la ficha termina en el extremo superior de una serpiente, se deslizará hasta su extremo inferior. En cambio, si termina en el extremo inferior de una escalera, ascenderá hasta su extremo superior. Gana la partida el jugador que antes alcance la última casilla.

El juego así planteado no requiere ninguna destreza. Pero supongamos que has trucado el dado y tienes el poder de elegir la cara que saldrá cada vez que lo tiras. ¿Sabes cuántos movimientos tendrías que hacer para ganar la partida si comienzas moviendo tú?

## Entrada

La entrada consta de una serie de casos de prueba. En la primera línea de cada caso aparecen cuatro números: la dimensión  $N$  del tablero, el número  $K$  de caras del dado ( $K \leq N$ ), el número  $S$  de serpientes y el número  $E$  de escaleras. Las siguientes  $S + E$  líneas contienen cada una dos números, indicando la casilla inicial y la casilla final de una serpiente (las  $S$  primeras) o una escalera (las  $E$  siguientes). Tanto  $N$  como  $K$  son números entre 1 y 100.

La entrada termina con 0 0 0 0.

## Salida

Para cada caso de prueba se escribirá el menor número de movimientos necesarios para ganar la partida. Está garantizado que la casilla final es alcanzable desde la inicial en todos los casos.

### Entrada de ejemplo

```
10 6 5 6
50 13
68 55
81 16
93 43
98 36
3 60
6 47
32 53
45 86
51 94
61 83
0 0 0 0
```

### Salida de ejemplo

```
3
```

**Autor:** Alberto Verdejo.

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

# La galería de tiro

Tiempo máximo: 1,000-2,000 s Memoria máxima: 10240 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=550>

La federación de tiro con pistola de Ruritania nos ha encargado el diseño de una galería de tiro donde la distribución de los tiradores se ha de hacer de la siguiente manera:

- El primer tirador que llegue ha de situarse en uno de los dos extremos, izquierdo o derecho, de la galería.
- Los siguientes tiradores han de situarse de manera que maximicen el número de puestos de tiro vacíos hasta el tirador más cercano.
- No se admiten más tiradores si no es posible tener al menos un puesto vacío entre dos tiradores.

Por ejemplo, si hay 6 puestos, los dos primeros tiradores se situarán en los extremos mientras que un tercero podrá ponerse en el tercer o en el cuarto puesto. Tras esto no será posible acomodar a más tiradores. Si hay 8 puestos de tiro, se puede acomodar a 4 tiradores: dos en los extremos y, por ejemplo, uno en el cuarto puesto y otro en el sexto. En una galería con 9 puestos podrían disparar hasta 5 tiradores simultáneamente.

La federación no está segura de su presupuesto, por lo que nos pide determinar, para un cierto número  $N$  de puestos de tiro, cuál va a ser el número máximo de tiradores a acomodar.

## Entrada

La entrada está formada por distintos casos de prueba, uno por línea. Cada línea contiene un único número entero  $1 \leq N \leq 1.000.000$  que representa el número de puestos en la galería.

El final de las entradas se indica con una línea con un 0, que no se debe procesar.

## Salida

Para cada caso de prueba se escribirá una línea con el número máximo de tiradores que se pueden situar en la galería.

## Entrada de ejemplo

```
2
5
7
10
0
```

## Salida de ejemplo

```
1
3
3
5
```

**Autor:** Luis Fernando Lago Fernández.

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

# Construyendo dianas

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

<http://www.aceptaelreto.com/problem/statement.php?id=648>

Del juego de los dardos hay muchas variantes. Todas consisten en tirar los dardos a una diana dividida en sectores de diferentes colores o que tienen asociada una puntuación distinta y conseguir cierto objetivo.



Queremos proponer una variante a la liga de dardos local: conseguir sumar cierto valor a partir de las puntuaciones obtenidas con los dardos pero con el menor número de tiros que sea posible. Por ahora tenemos una serie de dianas a las que hemos asignado puntuaciones diferentes a sus sectores y estamos interesados en conocer si ciertos valores pueden ser conseguidos tirando dardos a esas dianas y cuántos dardos como mínimo son necesarios en cada ocasión.

## Entrada

En la entrada aparecerán diferentes configuraciones de dianas y objetivos. Cada una ocupa dos líneas. En la primera aparecen dos números: el valor (entre 1 y 500) que hay que conseguir sumar tirando dardos a una diana y el número  $S$  de sectores (entre 1 y 50) en los que está dividida la diana. En la segunda línea aparecen, en orden estrictamente creciente, las  $S$  puntuaciones asociadas a esos sectores (valores entre 1 y 500).

## Salida

Para cada caso se escribirá el menor número de dardos necesarios para conseguir la cantidad, separado por dos puntos de las puntuaciones que permiten conseguir ese valor, ordenadas de mayor a menor y separadas por espacios.

Si hay varias soluciones, se escribirá aquella cuya mayor puntuación sea la más alta; si aún siguen existiendo varias soluciones, aquella cuya segunda mayor puntuación sea la más alta; y así sucesivamente.

Si es imposible conseguir el objetivo con las puntuaciones asignadas a los sectores de la diana, se escribirá **Imposible**.

## Entrada de ejemplo

```
100 5
10 15 20 25 30
8 3
1 4 6
25 2
8 12
```

## Salida de ejemplo

```
4: 30 30 30 10
2: 4 4
Imposible
```

**Autor:** Alberto Verdejo.

**Revisores:** Isabel Pita, Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

# Accidentes aéreos

Tiempo máximo: 2,000-3,000 s Memoria máxima: 16384 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=571>

Estamos programando un *bot* de Telegram que enviará noticias a sus seguidores y ahora nos toca tratar el escabroso tema de los accidentes de aviación. Cuando se produce uno de estos accidentes, suelen hacerse comentarios del estilo “*Este es el accidente más grave desde febrero de 1995*”.



Nos han pasado un listado (ordenado cronológicamente) de accidentes ocurridos en el pasado y queremos estar preparados para que, cuando ocurra el siguiente accidente, podamos producir un comentario como el anterior.

De hecho, para probar esta funcionalidad, queremos saber cuál habría sido el comentario cuando se produjo cada uno de los accidentes conocidos.

## Entrada

La entrada está formada por una serie de casos. Cada caso comienza con el número  $N$  de accidentes conocidos (un número entre 1 y 300.000). A continuación aparecen  $N$  líneas con la descripción de cada uno de ellos: una fecha (con formato DD/MM/AAAA) y el número de víctimas en ese accidente (no superior a  $10^6$ ). Todas las fechas son distintas y el listado está ordenado cronológicamente del accidente más antiguo al más reciente.

## Salida

Para cada caso se escribirán  $N$  líneas. La  $i$ -ésima línea contendrá la fecha del último accidente anterior que tuvo (estrictamente) más víctimas que el accidente  $i$ -ésimo. Si no existe tal accidente (en particular, eso ocurre siempre para el primero), se escribirá **NO HAY** en su lugar.

Después de cada caso se escribirá una línea con tres guiones (---).

## Entrada de ejemplo

```
6
19/12/1990 50
01/02/2000 80
10/05/2001 30
20/10/2005 10
08/07/2007 60
10/07/2007 40
```

## Salida de ejemplo

```
NO HAY
NO HAY
01/02/2000
10/05/2001
01/02/2000
08/07/2007
---
```

**Autor:** Alberto Verdejo.

**Revisores:** Isabel Pita, Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

# La fiesta del verano

**Tiempo máximo: 1,000-3,000 s    Memoria máxima: 4096 KiB**<http://www.aceptaelreto.com/problem/statement.php?id=312>

El segundo domingo de agosto se celebra la fiesta de mi pueblo. El ayuntamiento coloca grandes carpas en la playa y pone mesas para que comamos todos los vecinos. Las mesas se organizan por edades para evitar que los chavales se aburran con la conversación de sus padres y estos a su vez no tengan que aguantar el jaleo de los más pequeños. Las mesas son todas del mismo tamaño colocándose varias para un mismo tramo de edad si es necesario.

Cada año el ayuntamiento dedica un gran esfuerzo a calcular el número de mesas necesarias, de forma que sean lo más grandes posible, no sobre ningún hueco, y al mismo tiempo sean todas ellas del mismo tamaño. Este año han decidido informatizar el proceso. ¿Puedes ayudarles?



## Entrada

La entrada está formada por varios casos de prueba. Cada caso consiste en una lista de valores con el número de personas en cada tramo de edad, acabada con el valor cero. La última lista tendrá solamente el valor cero, y no debe ser procesada.

Cada tramo estará formado por al menos una persona y el número total de personas nunca será mayor que  $10^{18}$ .

## Salida

Para cada caso de prueba se indicará el número de mesas a utilizar. Recuerda que se desea que las mesas tengan el mayor tamaño posible sin que sobre ningún hueco.

## Entrada de ejemplo

```
10 25 30 0
60 45 0
7 0
0
```

## Salida de ejemplo

```
13
7
1
```

**Autores:** Isabel Pita y Alberto Verdejo.

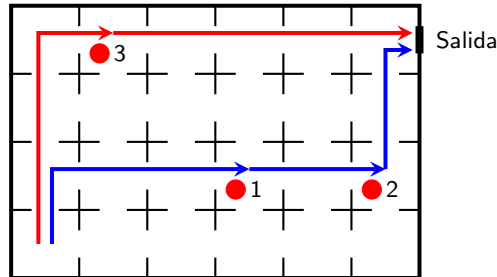
**Revisores:** Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.

# Más listos que el hambre

Tiempo máximo: 2,000-3,000 s Memoria máxima: 16384 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=398>

En el laboratorio estamos realizando una serie de experimentos para estudiar la inteligencia de una especie de ratones, los *ratones coloraos*. En los experimentos colocamos a unos ratones en una caja rectangular dividida en celdas como la de la figura. Los ratones se colocan en la esquina inferior izquierda y deben encontrar la salida situada en la esquina superior derecha. Para ello pueden moverse libremente por todas las celdas, pero de cada celda solamente pueden pasar a las colindantes en horizontal o vertical.



Para estos ratones eso está chupado, por lo que hemos ido complicando el experimento. Primero pusimos una serie de botones (círculos rojos en la figura) en algunas celdas de tal forma que los ratones tuvieran que pulsar todos esos botones para que se abriera la puerta de la salida. Pronto aprendieron a hacerlo, y además de la forma más rápida posible.

Para potenciar el trabajo en equipo, ahora hemos numerado los botones del 1 al  $N$ , y además han sido electrificados<sup>1</sup>. Para que el botón  $i$  sirva para abrir la puerta, y además no suelte una descarga eléctrica, deben haberse pulsado antes los botones con un número menor. En el experimento se colocan dos ratones en la celda inicial y ambos deben colaborar pulsando los botones en orden para lograr salir del laberinto en el menor tiempo posible. No hay más restricciones sobre qué botones pulsa cada ratón, salvo que si un ratón pulsa el botón  $i$ , antes él o su compañero deben haber pulsado el botón  $i - 1$ . Si lo logran encontrarán un succulento queso en la salida. Si no, solamente tendrán pan duro.

Nosotros somos mucho menos listos que los ratones y no sabemos si lo están haciendo en el mejor tiempo. ¿Nos puedes ayudar a calcularlo suponiendo que un ratón siempre tarda 1 segundo en pasar de una celda a otra colindante? El tiempo que tarda un ratón en pulsar un botón es despreciable, por lo que en el mismo segundo un ratón podría pulsar el botón  $i$  y el otro el botón  $i + 1$ . Ten en cuenta que la presencia de un botón en una celda no impide que un ratón pueda pasar por ella sin pulsarlo.

## Entrada

La entrada está compuesta por una serie de casos. Para cada caso, la primera línea contiene el número  $F$  de filas y el número  $C$  de columnas del entramado de celdas que contiene la caja (ambos números pueden estar entre 1 y 50). La siguiente línea contiene el número  $N$  de botones (entre 0 y 100) que deben pulsarse. En las siguientes  $N$  líneas se dan las posiciones de estos botones (una fila entre 1 y  $F$  y una columna entre 1 y  $C$ ) en el orden en que tienen que ser pulsados.

## Salida

Para cada caso se escribirá el tiempo mínimo que necesitan los dos ratones situados inicialmente en la celda (1,1) para salir de la caja habiendo pulsado todos los botones sin haber recibido ninguna descarga. Una vez alcanzada la celda  $(F,C)$ , si la puerta de salida está abierta, los ratones pueden salir con 1 segundo más.

<sup>1</sup>En la elaboración de este problema ningún animal ha sufrido daños. Se ha tenido un escrupuloso cuidado en respetar los derechos de los animales y las leyes contra el maltrato animal.



### Entrada de ejemplo

4	6
3	
2	4
2	6
4	2
3	3
4	
2	1
1	2
3	1
1	3

### Salida de ejemplo

11
5

**Autor:** Alberto Verdejo.

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