

Concurso de Programación Ada Byron

Fase local - UCM



Cuadernillo de problemas



Facultad
de
Informática



UNIVERSIDAD COMPLUTENSE
MADRID

13 de enero de 2017

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

Ada Byron

Listado de problemas

A	¡Me caso!	3
B	Alineación planetaria	5
C	Internet en el metro	7
D	El alcance de las historias	9
E	El juego de la linterna	11
F	La ronda de la noche	13
G	Clúster de microondas	15

Autores de los problemas:

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Enrique Martín Martín (Universidad Complutense de Madrid)
- Isabel Pita Andreu (Universidad Complutense de Madrid)
- Antonio Sánchez Ruiz-Granados (Universidad Complutense de Madrid)
- Clara Segura Díaz (Universidad Complutense de Madrid)
- Alberto Verdejo López (Universidad Complutense de Madrid)

● A

¡Me caso!

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
```

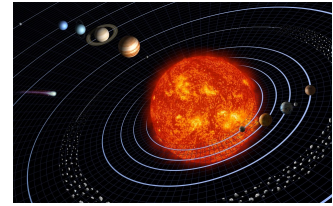
Salida de ejemplo

```
600
75
```




Alineación planetaria

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
```




Internet en el metro

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
```




El alcance de las historias

A David le encantan los libros de la colección “Elige tu propia aventura”, en los que hay páginas con opciones para continuar la historia de formas distintas. Además, en un mismo libro puede haber varias páginas de comienzo de historias. En estos libros hay contenidas varias historias posibles y le encanta explorar todas y cada una de ellas. Cuando lee este tipo de libros escribe en un cuaderno la secuencia de las páginas que ha leído desde una de las posibles páginas de comienzo hasta completar una historia y le fascina contemplar los saltos que ha dado por el libro, sobre todo los que se producen hacia el principio del libro. Como le gusta jugar con los números, para cada secuencia de páginas calcula la diferencia mayor entre cada página de la secuencia y cualquier otra página más adelante en la secuencia. Es lo que él llama el “alcance” de la historia. En un libro normal de n páginas, que empezase en la página 1 y en el que se avanzase de uno en uno hasta el final, el alcance sería -1 . Si se leyese de final al principio el alcance sería en cambio $n-1$. Ayuda a David a calcular el alcance de sus historias.



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 páginas leídas (entero mayor o igual que 2 y menor o igual que 300.000), mientras que la segunda línea contiene la secuencia de páginas (enteros mayores o iguales que 1 y menores o iguales que 300.000).

Salida

Para cada caso de prueba se escribirá el alcance de la historia, es decir, la mayor diferencia entre una página de la secuencia y las que le siguen en la secuencia.

Entrada de ejemplo

```
4
8
1 2 3 8 9 4 5 7
7
10 3 7 8 1 2 11
6
1 2 5 6 2 10
5
1 3 5 7 9
```

Salida de ejemplo

```
5
9
4
-2
```




El juego de la linterna

Jimmy tiene muchos sobrinos a los que tiene que entretener cada vez que van a su casa de visita. Hace tiempo que encontró la fórmula para conseguirlo. En realidad para él fue una sorpresa descubrir que aquel medio juego improvisado que empezó como una manera de tenerles tranquilos cinco minutos se ha convertido en una tradición que le piden fin de semana tras fin de semana.

El “juego” (por llamarlo de alguna forma) consiste en lo siguiente: Jimmy deja a todos los sobrinos en la habitación. Mientras él espera en el pasillo, los niños se colocan al azar uno al lado del otro formando una hilera de personajitos de distintas edades y alturas. Cuando ya están todos colocados, se apagan las luces y el tío Jimmy entra a oscuras y se coloca frente a la fila. Y entonces es cuando empieza la parte “divertida” del juego.



Jimmy saca una linterna de su bolsillo e ilumina la cara de uno de sus sobrinos al azar, que da un paso adelante. Después se desplaza a la derecha saltándose a un número de sobrinos al azar e ilumina a otro de los niños que avanza también saliéndose de la fila. El proceso lo repite una última vez, dejando a tres niños elegidos aleatoriamente e iluminados por la luz de la “linterna mágica”.

En ese momento, se da la luz y los tres sobrinos que han sido elegidos son premiados con cualquier cosa; unas veces son caramelos, otras veces es alguna moneda que los niños gastarán en su próxima visita a la tienda de chucherías. . .

Eso sí, el tío Jimmy pone una única condición para el premio: cuando los tres sobrinos que han dado un paso adelante salen de la fila, sus alturas son variopintas. Dependiendo de a quién haya elegido, el primero puede ser el más bajito, el más alto o el del centro. Los tres niños pueden, por tanto, quedar en orden creciente de altura, en orden decreciente o de cualquier otra forma. Pues bien, los niños se quedarán *sin premio* si el primer sobrino elegido es el más bajito de los tres, el segundo es el más alto y el tercero es el mediano.

Albertito, el primo mayor, ha decidido dirigir a todos sus primos y en el momento de colocarse supervisa si la colocación final puede llevar o no a que el tío Jimmy elija a tres niños de forma que no haya premio. Y en ese caso les pide a todos que se recolquen.

Por ejemplo, si ese día hay 5 niños y se colocan aleatoriamente de forma que las alturas quedan 5 1 4 3 2, el tío Jimmy podría tener la “puntería” de sacar al de altura 1, al de 4 y al de 2 y se quedarían sin premio. Sin embargo, en una colocación como 5 1 2 3 4 los sobrinos elegidos nunca se quedarán sin premio.

Entrada

La entrada estará compuesta por distintos casos de prueba, cada uno ocupando dos líneas. La primera línea contiene un número indicando el número de sobrinos que hay ese día en casa (como mínimo 3 y hasta 500.000). La segunda línea contendrá las alturas de cada uno de ellos en la configuración aleatoria en la que se han colocado. Las alturas no se repiten, y serán números entre 1 y 10^9 .

Salida

Para cada caso de prueba se indicará el veredicto que debe dar Albertito ante esa colocación. Se escribirá SIEMPRE PREMIO si el tío Jimmy nunca podrá sacar una configuración de sobrinos sin premio, o ELEGIR OTRA si el tío puede sacar a tres sobrinos de forma que se queden sin premio.

Entrada de ejemplo

5
5 1 4 3 2
5
5 1 2 3 4

Salida de ejemplo

ELEGIR OTRA SIEMPRE PREMIO



La ronda de la noche

En Ciudad Decoro les preocupa que los jóvenes se reúnan con sus parejas en la noche, a espaldas de sus padres. Por ello en la última Ordenanza Municipal se ha obligado a que todas las casas tengan un jardín-laberinto para acceder a su puerta principal y que (opcionalmente) se instalen sensores de movimiento en los mismos. Conscientes de la posible dificultad de implantación de la Ordenanza, se ha decidido que todos los jardines-laberinto estén organizados en cuadrículas donde cada casilla está libre, tiene un muro o alberga un sensor de movimiento.

Como no podría ser de otra forma, la plataforma activista *#FreeLove* va a hacer todo lo posible para evitar que estas nuevas medidas supongan un obstáculo real para el amor. Durante las últimas semanas ha ido investigando y realizando mapas de todos los jardines-laberinto, anotando la posición de los sensores de movimiento. *#FreeLove* ha detectado que únicamente hay 10 tipos diferentes de sensores de movimiento, etiquetados como *CAT-k* donde k es un número natural entre 0 y 9. Los sensores captan movimiento en línea recta en las 4 direcciones (norte, sur, este y oeste) y la k de su categoría indica el número de casillas que cubren en cada dirección. De esta manera, un sensor *CAT-0* únicamente capta movimiento en la casilla en la que está instalado, mientras que un sensor *CAT-2* capta movimiento en su casilla y en 2 casillas en cada una de las 4 direcciones (en total cubre 9 casillas). Sin embargo, lo más interesante que ha descubierto *#FreeLove* es que los sensores no pueden traspasar los muros del laberinto, así que en ocasiones su alcance en alguna dirección se ve limitado.

									P
		2							
E						1			

#FreeLove necesita descubrir cuáles de los jardines-laberinto permiten que un amante vaya de la entrada del jardín a la puerta principal de la casa sin ser descubierto por ningún sensor. ¿Podrías ayudarles a diferenciar los jardines-laberinto impenetrables de aquellos favorables al amor, y en esos casos calcular el mínimo número de casillas que hay que atravesar para llegar de la entrada del jardín a la puerta principal?

Entrada

La entrada comienza con una línea conteniendo el número de jardines a analizar. Cada jardín comienza con dos números $0 < ancho, alto \leq 1000$ con el ancho y alto del jardín-laberinto en una línea. Le sigue la descripción del jardín en *alto* líneas de *ancho* caracteres cada una. Estos caracteres son:

- '#': Una pared.
- '.': Una casilla libre.
- 'E': La casilla donde está la entrada al jardín-laberinto.
- 'P': La casilla donde está la puerta principal de la casa.
- k , con $0 \leq k \leq 9$: Casilla que alberga un sensor *CAT-k*.

La entrada al jardín y la puerta principal de la casa pueden estar en cualquier parte del jardín, incluidas casillas interiores. Además, nada impide que algún sensor vigile la casilla donde está la entrada al jardín o la puerta principal de la casa, por lo que en esos casos no será posible recorrer el jardín sin ser descubiertos.

Salida

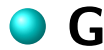
Por cada jardín-laberinto la salida será una línea con el mínimo número de casillas del jardín que hay que atravesar para llegar de la entrada a la puerta principal de la casa sin ser descubierto, o la palabra NO en caso de que sea imposible.

Entrada de ejemplo

```
3
9 4
.....P
..2#.....
E.##..1..
.....
6 4
....#P
..#...
.####.
E....1
10 5
E.....
.....
....2.....
.....
.....P
```

Salida de ejemplo

```
12
10
NO
```

Clúster de microondas

Ante los retrasos continuos en la hora de llegada de los alumnos, los profesores que dan clase a primera hora de la tarde de cierta Facultad han decidido tomar medidas y no dejar entrar a nadie que llegue más de cinco minutos tarde. ¡Se acabaron las interrupciones!

Desde Delegación de Alumnos han contraatacado, culpando al Decanato de la situación. A medio día, argumentan, se forman unas grandes colas en los microondas disponibles para uso libre, lo que hace que haya que esperar mucho tiempo hasta conseguir tener la comida caliente. Y con lo aburridas que son algunas clases, aseguran, no todas las asignaturas se merecen renunciar por ellas a comer los macarrones calientes.

Antes de que llegue la sangre al río ante semejantes comentarios, desde Decanato han pensado que igual habría que hacer una ampliación de *hardware* en el clúster de microondas de la Facultad para evitar males mayores. Desde Asuntos Económicos miran con preocupación la medida, de modo que han pedido que se haga un estudio del número de microondas que se necesitaría tener disponibles para mantener la espera de sus usuarios dentro de unos márgenes razonables.

Y el Decanato ha querido involucrar a Delegación en él, para evitar críticas futuras... y les ha pedido que hagan ellos la estimación.



Entrada

El programa deberá leer de la entrada estándar múltiples casos de prueba.

Cada uno comienza con dos números enteros, n y t , indicando respectivamente cuánta gente utiliza los microondas a lo largo de un día cualquiera (como mucho 50.000), y cuál es el tiempo de espera máximo permitido.

Después aparecerá una línea con n parejas de números, una por cada hambriento usuario. El primer número de cada pareja indica el instante de tiempo en el que llega a hacer uso de un microondas, y el segundo el tiempo total (mayor que 0) que tiene que utilizarlo para calentar completamente su comida. Los usuarios estarán ordenados por orden de llegada. Hay veces que llegan en grupo, por lo que tendrán el mismo instante de llegada; en ese caso usarán los microondas en el orden en el que aparecen en la entrada.

La entrada terminará con un caso, que no deberá procesarse, en el que no hay nadie que quiera usar los microondas ($n = 0$).

Por simplicidad, los tiempos son números naturales carentes de unidad.

Salida

Por cada caso de prueba se escribirá el mínimo número de microondas que tiene que haber en el clúster para que nadie espere más tiempo del máximo permitido antes de empezar a calentar su comida. Se asume que hay “fila única”, y en cuanto un microondas queda libre (sea cual sea), la siguiente persona en la cola comienza a utilizarlo sin dilación en un tiempo despreciable.

Entrada de ejemplo

```
2 5
0 5 0 3
3 5
0 6 0 3 10 4
0 0
```

Salida de ejemplo

1
2