

Concurso de programación

# Las 12 UVas 2021

## Problemas



Premios patrocinados por



Ejercicios realizados por



Organizado desde la **Facultad de Informática (UCM)**  
31 de diciembre de 2021





## Listado de problemas

<b>A La otra página</b>	<b>3</b>
<b>B La media prometida</b>	<b>5</b>
<b>C El Joker de la Primitiva</b>	<b>7</b>
<b>D Doblando calcetines</b>	<b>9</b>
<b>E Triángulo de espumillón</b>	<b>11</b>
<b>F Plinio el Joven</b>	<b>13</b>
<b>G Plinio el Viejo</b>	<b>15</b>
<b>H Eligiendo uvas</b>	<b>17</b>
<b>I Villancicos al viento</b>	<b>19</b>
<b>J Torneo de pádel</b>	<b>21</b>
<b>K Ascensores con nivel</b>	<b>23</b>
<b>L Devorando el turrón</b>	<b>25</b>

Autores de los problemas (por orden alfabético de apellido):

- Marco Antonio Gómez Martín (Universidad Complutense de Madrid)
- Pedro Pablo Gómez Martín (Universidad Complutense de Madrid)
- Isabel Pita (Universidad Complutense de Madrid)
- Alberto Verdejo López (Universidad Complutense de Madrid)



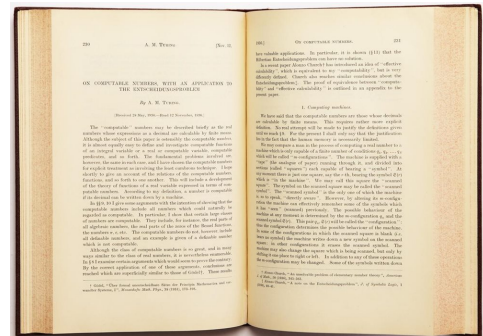
# A

## La otra página

Aunque hoy día estamos acostumbrados a que los libros tengan las páginas numeradas, no siempre ha sido así. De hecho antes de la aparición de la imprenta pocos libros eran los que incorporaban numeración. Esta numeración, además, no era por páginas sino por hojas en donde cada folio aparecía numerado únicamente por su parte delantera.

Incluso con la llegada de la imprenta, el uso de numeración tardó en adoptarse. Prueba de ello es que, aunque el nacimiento del libro impreso ocurrió en 1455 con la aparición de la primera Biblia de Gutenberg, el primer libro que salió de una imprenta con números (de hoja) no lo hizo hasta 1470, quince años después. Y hay que esperar hasta el siglo XVI para ver la adopción generalizada de la numeración por páginas.

Lo que parece que no generó dudas fue cómo numerar esas páginas. Desde el principio la hoja con numeración impar aparece a la derecha cuando se abre el libro completamente.



### Entrada

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

Cada caso de prueba consiste en un único número (entre 2 y 1000) con la numeración de una de las páginas que se ve en el libro abierto.

### Salida

Por cada caso de prueba se escribirá, en una línea independiente, el número de la otra página visible.

### Entrada de ejemplo

```
3
2
10
11
```

### Salida de ejemplo

```
3
11
10
```

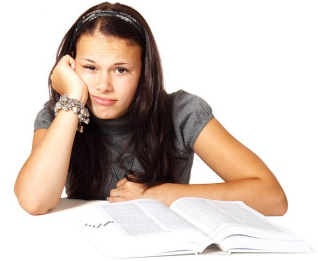




## La media prometida

Adolfina prometió a sus padres que, al final del trimestre, tendría un 7 en la asignatura que peor se le da. “¿Puedes prometerlo y lo prometes, o solo lo prometes y no vas a cumplirlo?”—le retó su padre. Ella contestó que podía prometerlo y que lo prometía: llevaría a casa un 7.

Pero ahora ya no está tan segura. Tiene varias notas de exámenes parciales y mañana tiene el último. La nota final será la media de todas las que ya tiene, junto con la que saque en el examen de mañana. Y ahora le preocupa más atinar con la calificación exacta que necesita para conseguir la media prometida que estudiar para el examen.



### Entrada

Cada caso de prueba comienza con un número indicando cuántos exámenes ha hecho Adolfina hasta el momento, entre 1 y 10. A continuación aparecen las calificaciones que ha conseguido, separadas por espacio, que serán siempre valores enteros entre 0 y 10. Tras la lista aparece, en una línea independiente, la nota media que ha prometido a sus padres conseguir.

La entrada termina con un 0.

### Salida

Por cada caso de prueba el programa escribirá la nota que debe sacar Adolfina en el último examen para conseguir la media que ha prometido. Si resulta imposible, se escribirá “IMPOSIBLE”.

En el examen solo puede sacar una nota entera entre 0 y 10.

### Entrada de ejemplo

```
3
6 6 6
6
2
10 10
7
3
6 6 7
8
0
```

### Salida de ejemplo

```
6
1
IMPOSIBLE
```





# C

## El Joker de la Primitiva

La *Lotería Primitiva* celebró su primer sorteo el 10 de diciembre de 1763, como un medio del Estado para recaudar dinero sin crear un nuevo impuesto. Originalmente denominada, simplemente, *Lotería por números*, se renombró a *Lotería primitiva* cuando se introdujo, en 1812, la que se llamó *Lotería moderna* que fue el germen de la que hoy se conoce como *Lotería Nacional de España*.



La celebración de la Lotería Primitiva se interrumpió en 1862, y volvió a recuperarse, más de un siglo después, en 1985, celebrándose desde entonces. En 2012 se introdujo la posibilidad de jugar, con el mismo boleto, a un sorteo asociado y opcional llamado *JOKER*.

En este sorteo secundario cada participante recibe un número aleatorio de 7 cifras asignado por la administración. Durante el sorteo, de un bombo que contiene bolas numeradas con los dígitos del 0 al 9 se extraen 7 números, reincorporando al bombo la bola elegida tras cada extracción. La persona ganadora es aquella cuyo número de 7 cifras coincida con la combinación formada por los números extraídos del bombo *en el mismo orden*.

Las malas lenguas dicen que, durante los primeros borradores de la especificación del sorteo, los creadores olvidaron especificar la coletilla “en el mismo orden”. Afortunadamente alguien se dio cuenta a tiempo y evitó que el sorteo fuera deficitario para las arcas del Estado.

### Entrada

La entrada recibida por el programa comienza con un número indicando cuántos casos de prueba deberán ser procesados.

Cada caso está compuesto por dos números de 7 dígitos indicando, respectivamente, los dígitos extraídos en el sorteo del JOKER y los del número de un boleto concreto.

### Salida

Por cada caso de prueba se escribirá **GANA** si el boleto resulta ganador con la definición errónea del sorteo, en la que se habría faltado la coletilla de “en el mismo orden”. En otro caso se escribirá **PIERDE**.

### Entrada de ejemplo

```
4
1234567 7654321
0011223 0112233
1234567 1234567
0123450 0012345
```

### Salida de ejemplo

```
GANA
PIERDE
GANA
GANA
```





# Doblando calcetines

Lo de doblar calcetines me supone una tensión fuera de lo común. Supongo que tiene que ver con el método que utilizo pero es el que he usado desde siempre y no soy muy dado a los cambios.

Mi tarde doblando empieza con un barreño lleno de calcetines limpios que acabo de quitar del tendedero. Del barreño los voy sacando de uno en uno sin rebuscar en el montón; si su pareja ya ha salido, los emparejo y los meto en el cajón. Si aún no ha salido, lo estiro y lo dejo cuidadosamente en la mesa para identificarlo rápidamente cuando encuentre el compañero. Cada vez que añado uno sin pareja a la mesa aumenta mi nerviosismo pues por un lado el hueco en la mesa es escaso y por otro no sé si podré emparejarlo o pasará a formar parte de los calcetines solteros que pueblan el mundo. Afortunadamente, siempre consigo agruparlos todos, pero algún día seguro que me quedo sin hueco en la mesa...



## Entrada

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

La primera línea de cada caso contiene el número  $N$  de calcetines que tengo que doblar, siempre par y como mucho 1000. En la siguiente línea aparecen  $N$  números que simbolizan los calcetines que voy cogiendo. Cada calcetín viene determinado por su tipo (color, grosor, longitud, etc.) y es un número entre 1 y 100.

La entrada termina con una línea con un 0 que no debe procesarse.

## Salida

Por cada caso de prueba se escribirá una única línea con el número máximo de calcetines que tengo sin emparejar durante el proceso.

### Entrada de ejemplo

```
4
1 1 2 2
4
1 2 2 1
6
1 2 1 1 1 2
0
```

### Salida de ejemplo

```
1
2
2
```





## Triángulo de espumillón

Después de adornar el árbol de navidad, te han sobrado un montón de tiras de espumillón de distintas longitudes. Quieres aprovechar tres para rodear la base del árbol formando un triángulo para colocar dentro los regalos.

Tienes la esperanza de recibir muchos regalos, y no quieres que el hueco sea un problema. Por eso, quieres hacer el triángulo con el mayor perímetro posible. De modo que te preguntas, dadas las longitudes de las tiras de espumillón que tienes, ¿cuál es el triángulo de mayor perímetro que puedes crear utilizando una para cada uno de los tres lados?



### Entrada

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

Cada caso comienza con una línea indicando el número de tiras de espumillón que tienes (como mucho 3000). A continuación aparece una línea con las longitudes de cada una (como mucho 100.000).

### Salida

Para cada caso de prueba se escribirá en una línea independiente el perímetro máximo que puedes conseguir. Si no se puede formar ningún triángulo, se escribirá **NO HAY NINGUNO**.

### Entrada de ejemplo

```
2
5
1 2 2 1 2
3
1 1 10
```

### Salida de ejemplo

```
6
NO HAY NINGUNO
```



# ● F

## Plinio el Joven

Plinio el Joven fue un abogado, escritor y científico de la antigua Roma. Ilustre orador seguidor de Cicerón, su primera obra, una tragedia en griego, la escribió con solo 14 años. Aunque la mayoría de sus escritos se han perdido, destacó en poesía y es considerado por algunos críticos el inventor del género literario de la carta escrita para ser publicada.

Una de sus cartas más famosas es la que dirigió a Tácito, historiador romano, describiendo la erupción del monte Vesubio en la que murió su tío, Plinio el Viejo. Debido a ella, las erupciones de ese tipo se conocen hoy como *plinias*.

Plinio el Joven fue joven toda su vida porque, según cuenta la leyenda<sup>1</sup>, nació un 29 de febrero, por lo que solo cumplía años una vez de cada cuatro.

Dada una fecha, ¿podrías decirnos el siguiente cumpleaños de Plinio?



### Entrada

El programa deberá leer, de la entrada estándar, un primer número indicando cuántos casos de prueba deberán procesarse.

Cada caso de prueba está compuesto por una fecha en formato DD/MM/AAAA, indicando el día, mes y año de la fecha a consultar, que será siempre correcta. El año estará entre 60 y 1581.

### Salida

Por cada caso de prueba el programa escribirá, en el mismo formato, la fecha del siguiente cumpleaños de Plinio el Joven, es decir el siguiente 29 de febrero. Los años consultados son anteriores a la reforma Gregoriana ideada por Clavius e impulsada por el papa Gregorio XIII, por lo que se consideran bisiestos *todos* los años divisibles por 4.

Si la fecha de entrada es el cumpleaños de Plinio, se indicará el siguiente.

### Entrada de ejemplo

```
3
04/12/0061
29/02/1500
01/01/0444
```

### Salida de ejemplo

```
29/02/0064
29/02/1504
29/02/0444
```

---

<sup>1</sup>Una leyenda seguramente falsa, porque acabamos de inventarla.





# ●● G

## Plinio el Viejo

Plinio el Viejo fue un escritor y militar de la antigua Roma. Tío y padre adoptivo de Plinio el Joven, perteneció a la clase social de los caballeros romanos y luchó contra Germania durante 12 años. También fue procurador en Galia e Hispania.

Su obra *Historia natural* condensó en forma enciclopédica gran parte del conocimiento de su época en el Imperio Romano. Sus escritos sirvieron como modelo hasta el siglo XVII cuando empezaron a ser sustituidos por estudios que seguían el método científico. Murió en la erupción del monte Vesubio en el año 79, seguramente por los gases volcánicos del flujo piroclástico, cuando se acercó a estudiar el fenómeno y a ayudar a sus conocidos.

Cuenta la leyenda<sup>2</sup> que Plinio fue siempre viejo porque contaba sus años como *años lunares* compuestos por 12 meses de 28 días.



### Entrada

Cada caso de prueba está compuesto por una fecha en formato DD/MM/AAAA, indicando el día, mes y año de la fecha de nacimiento de una persona siguiendo nuestro calendario solar. El año estará entre 20 y 1581, por lo que los años múltiplos de 4 son *todos* bisiestos.

A continuación aparece la edad de dicha persona mediante tres números con los años, meses y días que tiene, respectivamente. Recuerda que cuenta su vida usando el calendario lunar, en el que los años están compuestos por 12 meses de 28 días cada uno.

La entrada termina con una fecha y edad con todos los valores a 0.

### Salida

Por cada caso de prueba el programa escribirá la fecha (en nuestro calendario solar) en la que la persona tenía esa edad. La fecha se escribirá con el mismo formato DD/MM/AAAA utilizado en la entrada.

Se garantiza que la salida no será posterior al 4 de octubre de 1582.

### Entrada de ejemplo

```
28/02/1500 0 0 2
01/12/0061 0 1 3
06/05/0432 1 1 1
00/00/0000 0 0 0
```

### Salida de ejemplo

```
01/03/1500
01/01/0062
06/05/0433
```

---

<sup>2</sup>La leyenda inventada para este problema, que es a todas luces falsa.



# ●● H

## Eligiendo uvas

La variedad de uvas es enorme. Las hay blancas, tintas, de mesa, para vino. . . Y cada una con su nombre o denominación de origen: aledo, moscatel, cariñena, loureira. . . ¡un mundo!

Incluso dentro del mismo tipo, luego en un racimo hay uvas con más o menos sabor, o con la piel más o menos gruesa. Noah L. Gido siempre terminaba cogiendo las peores uvas del racimo, y decidió que eso se tenía que acabar. Desde entonces, cuando coge su siguiente uva busca aquella que parezca la más sabrosa y, al mismo tiempo, la de piel más fina.

Pero en los racimos grandes tarda mucho tiempo en elegir y necesita ayuda para evitar que pase el año sin que haya escogido.



### Entrada

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

Cada uno está compuesto por la descripción de un número indeterminado de uvas. Cada uva está definida con dos números enteros entre 1 y  $10^{18}$  indicando, respectivamente, el sabor estimado y el grosor de la piel de dicha uva.

La lista de uvas termina con dos ceros.

### Salida

Por cada caso de prueba se escribirá, en una línea independiente, la descripción de la mejor uva disponible, es decir la que tiene un sabor al menos igual de bueno que todas las demás y, al mismo tiempo, una piel al menos tan fina como todas las demás.

Si no hay ninguna uva que sobresalga del resto en ambos aspectos, se escribirá **NO HAY MEJOR**.

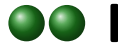
### Entrada de ejemplo

```
7 1 6 2 5 2 0 0
7 3 8 4 0 0
```

### Salida de ejemplo

```
7 1
NO HAY MEJOR
```





# Villancicos al viento

En Villa Cinco son muy aficionados a los villancicos. María Morena vive en una de sus calles, de casitas bajas unifamiliares, donde la mayoría de los vecinos ponen sus villancicos favoritos en las fechas señaladas.



Esta nochevieja sopla un fuerte viento del este, por lo que la música puesta en una casa es arrastrada por el aire y se escucha en varias de las casas calle abajo. Dependiendo del volumen, llegará más o menos lejos.

Dado el número de casas a las que llegan los villancicos puestos en cada vivienda de la calle, ¿cuántas oyen música de algún vecino?

## Entrada

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

La primera línea de cada caso contiene el número  $1 \leq n \leq 500.000$  de casas de la calle. La segunda línea contiene la distancia a la que llegan, hacia la izquierda, los villancicos puestos por cada una de las  $n$  casas. Por ejemplo, un 1 indica que los villancicos de esa vivienda son oídos por la situada inmediatamente antes. Ninguna casa pone la música tan alta como para que sea escuchada más allá de 500.000 casas.

La entrada termina con un caso de prueba sin viviendas, que no deberá ser procesado.

## Salida

Por cada caso de prueba, el programa escribirá cuántas casas de la calle escuchan música de alguno de los vecinos calle arriba.

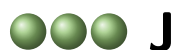
## Entrada de ejemplo

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

## Salida de ejemplo

```
1
3
```





# Torneo de pádel

Poco después de que terminara la obra de construcción de mi urbanización y nos dieran las llaves a los flamantes nuevos propietarios, aquello se llenó de camiones de mudanza de contentos vecinos deseosos de estrenar su nueva casa. En general reinaba un ambiente de felicidad propicio para que muchos de esos desconocidos entablaran una rápida amistad.

Con ese caldo de cultivo pronto hubo quien propuso organizar un torneo de pádel en la, aún por estrenar, pista comunitaria. Dicho y hecho, ese entusiasta grupo de vecinos organizó el que llamaron Primer Torneo de Pádel “*Lo importante es la gente*” y, ni cortos ni perezosos, nos apuntaron a todos los demás sin preguntar.

El torneo fue en modo campeonato. Aprovechando que el número de viviendas es una potencia de dos, lo organizaron con distintas eliminatorias a un único partido. Si ganabas el partido pasabas a la siguiente ronda y si perdías quedabas descalificado.

No es de extrañar, claro, que muchos de nosotros no fuéramos a los partidos. Afortunadamente la normativa estaba preparada para eso. Si a un partido faltaba uno de los jugadores, el otro ganaba automáticamente sin jugar. Si faltaban los dos, no se jugaba tampoco el partido de la siguiente ronda, clasificándose automáticamente el otro jugador (si lo había).

A la vista de la intención de cada vecino de participar o no en el torneo, ¿cuántos partidos se jugaron en las primeras rondas?



## Entrada

La entrada está compuesta de distintos casos de prueba, cada uno en dos líneas.

La primera línea de cada caso de prueba indica el número de jugadores (siempre una potencia de dos entre 2 y  $2^{18}$ ) seguido del número de rondas sobre las que se quiere conocer los partidos jugados.

A continuación aparece una línea con tantos números como jugadores. Un 1 indica que ese jugador se presentará a todos sus partidos, mientras que un 0 indica que el vecino no tiene intención de jugar. El orden de los jugadores está dado de forma que en la primera ronda se enfrentan el primer jugador contra el segundo, el tercero contra el cuarto y así sucesivamente. Además están colocados de forma que en la final se enfrentan el ganador de la primera mitad de jugadores con el ganador de la segunda mitad y este esquema se repite para conocer los ganadores de ambas mitades.

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

## Salida

Por cada caso de prueba se escribirá una única línea con el número de partidos jugados en las rondas indicadas.

## Entrada de ejemplo

```
4 2
1111
4 1
1111
4 2
0110
8 2
10101010
0 0
```

### Salida de ejemplo

3
2
1
2





# Ascensores con nivel

Kevin acaba de empezar a trabajar en la multinacional *PorNiveles*, cuyas oficinas se encuentran en uno de los rascacielos más altos de la ciudad. Por supuesto, Kevin ha comenzado como simple becario y su puesto de trabajo se encuentra en un despacho compartido en la primera planta. Su jefe directo tampoco es muy importante y su oficina se encuentra en la segunda planta. Normalmente Kevin solo se mueve entre esas dos plantas, cuyas ventanas tienen unas vistas nada llamativas. A él le gustaría subir al último piso desde donde le han dicho que se puede ver toda la ciudad y más allá.



Pero la empresa está fuertemente jerarquizada y todos los empleados se distribuyen entre una serie de *niveles* que influyen en todos los aspectos, desde el sueldo a determinados privilegios, como el piso del rascacielos hasta el que uno puede subir.

El enorme y altísimo edificio tiene un intrincado sistema de ascensores. No todos van desde la planta baja a la azotea, sino que cada ascensor se mueve entre dos plantas concretas, *A* y *B*, permitiendo viajes entre cualesquiera plantas entre esas dos. Pero si uno quiere subir por encima de *B* o bajar por debajo de *A* tendría que cambiar de ascensor. Además el acceso a los ascensores está restringido. Cada ascensor también tiene asignado un nivel y solamente los empleados con ese nivel o superior pueden utilizarlo.

Kevin ha comprobado ya que con su nivel de becario no puede subir muy alto y se pregunta qué nivel es el que hace falta para alcanzar la última planta.

## Entrada

La entrada está formada por diferentes casos de prueba. La primera línea de cada caso contiene el número *N* de ascensores en el edificio ( $1 \leq N \leq 100.000$ ). A continuación aparecen *N* líneas, cada una describiendo un ascensor con tres números, la planta más baja y la más alta que conecta el ascensor y el nivel necesario para poder utilizarlo. Las plantas están numeradas empezando por la 0 (la planta baja) y un edificio nunca tiene más de un millón de plantas. Los niveles varían entre 1 y 1.000.000. Se garantiza que siempre es posible (con el nivel apropiado) subir en ascensor desde la planta 0 a la más alta.

## Salida

Para cada caso de prueba se escribirá cuál es el nivel mínimo necesario para poder subir en ascensor desde la planta 0 hasta la última planta del edificio.

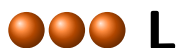
## Entrada de ejemplo

```
5
0 3 3
3 10 11
0 4 2
2 8 6
8 10 8
```

## Salida de ejemplo

```
8
```





## Devorando el turrón

Parecía que no iba a llegar nunca pero por fin te encuentras en la sobremesa de la cena de nochevieja. Y es que hacer de anfitrión y que venga todo el mundo a cenar a casa es agotador. Te pasas el día preparando entrantes, primer plato, plato principal, postre y uvas. Tienes que encargarte de que las bebidas estén frías, la mesa en perfecto estado y la casa ordenada. Y todo eso, además, teniendo en cuenta que la mañana la has pasado ocupada liado en el ordenador.

Y lo peor de todo es que ni siquiera has cenado bien. No has hecho más que ir y venir de la cocina al salón, preocupándote de que no le faltara nada a tus comensales y ahora te das cuenta de que ellos ya están atiborrados mientras a ti los turrones que cuidadosamente colocaste ayer en la bandeja (sabías que hoy no te daría tiempo) te están llamando a gritos.

En la bandeja hay dos hileras con pedazos de turrón de distintos tipos. Por la posición que ocupas llegas a coger turrones de ambas hileras así que comerás de ambas, aunque por educación irás cogiendo solo de los dos extremos más cercanos, haciendo que las hileras vayan siendo cada vez más cortas.

Si tienes una estimación del tiempo que tardas en comer cada pedazo, ¿cuántos trozos de turrón podrás comerte como mucho en el tiempo que queda de fiesta?



### Entrada

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

La primera línea de cada caso de prueba comienza con dos números,  $n$  y  $m$ , representando el número de trozos de turrón que hay en cada hilera (entre 0 y 200.000) seguidos del tiempo que tienes para comer turrón (entre 0 y  $10^9$ ).

Las dos líneas siguientes contienen  $n$  y  $m$  números respectivamente con el tiempo que se tarda en comer cada pedazo de turrón (entre 1 y 2000). Los pedazos que tienes más cerca (que son por los que empiezas comiendo) corresponden a los últimos números de cada línea.

### Salida

Por cada caso de prueba se escribirá una única línea con el número de pedazos de turrón que, como mucho, podrás comer en el tiempo disponible.

### Entrada de ejemplo

```
3 2 2
2 2 1
2 1
3 5 5
1 3 3
2 1 1 1 1
```

### Salida de ejemplo

```
2
4
```