



Concurso de Programación Concurso Final 2026

<https://urjc-cp.github.io/urjc-cp/>

Cuadernillo de problemas



Realizado en la **Escuela Técnica Superior de Ingeniería Informática (URJC)**
10 de abril de 2026

Índice

A El torneo mas corto	3
B Buscando oro	4
C La isla de los programadores	6
D La peor entrevista de...	7
E Números de Lychrel	9
F Evitando lag	10
G Plan de movilidad	11

Autores de los problemas:

- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Sergio Salazar Cárdenas (Universidad Rey Juan Carlos)

Tiempo: 1 segundo



El torneo mas corto

Tu amigo Fernando es un gran amante del balonmano y junto a su club ha decidido desarrollar un torneo a nivel nacional, de forma que cualquier club del país pueda apuntarse y poder conocer a grandes jugadores de todo el territorio.

Aunque en su cabeza no creía recibir a demasiados equipos, como todo el mundo sabe, el balonmano es un deporte especialmente popular, así que ahora le toca saber cuantos partidos debe organizar durante el torneo. Para resolver su problema ha decidido que el torneo se desarrolle a través del método de eliminación directa, es decir, en cada partido se enfrentan dos equipos, si pierdes quedas automáticamente descalificado. En caso de empate se jugara a balón de oro (el que meta gana de toda la vida), de esta forma siempre habrá un ganador.

Como si organizar un evento de estas magnitudes no fuese suficiente, desde el club están poniendo pegas a Fernando, pidiéndole que el torneo sea lo más corto posible tanto en partidos jugados, como en días de duración, ya que se necesita tiempo para otros eventos y los costes de arbitraje e instalación no son precisamente bajos.

Gracias a dios el pabellón donde se organiza el Torneo es suficientemente extenso como para poder jugar todos los partidos que se quieran a la vez, eso si, cada equipo solo jugará un partido al día (necesitan descansar). Bajo esta situación, ¿podrías indicarle cuántos partidos se deben jugar para un cierto número de equipos participantes?

Entrada

La primera contiene un entero C , el número de casos de prueba a evaluar. Las siguientes C líneas contienen un único entero N la cantidad de equipos que se han apuntado al torneo.

Salida

Por cada caso de prueba se deben imprimir un entero, el número de partidos a realizar.

Entrada de ejemplo

```
1
4
```

Salida de ejemplo

```
3
```

Límites

- $1 \leq C \leq 10000$
- $1 \leq N \leq 2^{60}$

Tiempo: 1 segundo

● B

Buscando oro

La fiebre del oro es el nombre de un período de migración de trabajadores hacia áreas rústicas, en las que se había producido un descubrimiento grande de oro. Uno de los más relevantes es el de California desde 1848 hasta 1960. Además, desde 2010 hay un programa con más de 258 capítulos sobre cómo algunas familias son capaces de encontrar oro utilizando grandes maquinarias enfrentándose a las adversidades del terreno, clima, etc. Parker es uno de los “buscadores de oro”, el cual conoce uno de los problemas más críticos. Las tierras ricas en oro se encuentran en zonas muy al norte de Canadá, como por ejemplo en Klondike, región de la ciudad de Dawson, Yukon, Canadá. Debido al invierno, en estas fechas es imposible llevar a cabo trabajos de minería, puesto que el terreno se congela. Parker ha decidido ir a Australia, donde suele ir cuando el invierno comienza en el Yukon, para poder continuar todo el año extrayendo oro, y necesita encontrar el mejor terreno. Para determinar si un terreno es prometedor o no, además de hablar con los habitantes, realizan una prospección del terreno, con una pequeña grava rica en oro, determinando si es prometedor o no. Durante su viaje, habla con diferentes habitantes que tienen algunos terrenos que están dispuestos a vender, pero esta tarea es muy tediosa y lenta. Con el fin de amortizar el tiempo, Parker ha decidido realizar un examen topológico a nivel de la región, en el cual puede observar la información que necesita de todos los terrenos. No todos los territorios le sirven, solo aquellos donde haya alguien dispuesto a vender. Parker conoce a todos los habitantes que están dispuestos a vender, pero no sabe muy bien como realizar este análisis, ya que son mapas muy grandes. Por esto, ha decidido contratar a una persona para desarrollar un programa que, dado el mapa de la región, determine cuál es el terreno más prometedor. Cada terreno vendrá limitado por tierras del gobierno que no pertenecen a la parcela. Un mismo terreno se define como todas aquellas parcelas de tierra u oro conectadas entre sí, limitadas por terrenos del gobierno. Parker ha venido con un buen lingote de oro para pagar a quien le ayude con esta tarea.

Entrada

La entrada comienza por un número T , que contiene el número de casos de prueba. Por cada caso la primera línea contiene dos números N y M que denotan el tamaño de la cuadrícula del mapa que han examinado. Seguidamente aparecen N líneas con M columnas donde cada uno de estos datos es un carácter C , que puede ser “O” si hay oro, “T” si hay tierra y “G” si es un terreno del gobierno. A continuación tenemos un número L que indica el número de contactos con habitantes que tiene Parker. Por cada uno de los habitantes L viene una línea en la cual se muestra el nombre del habitante NL , el identificador de la tierra ID y un punto del terreno representados por un primer número con la coordenada X y la coordenada Y .

Salida

La salida será el nombre del habitante con el identificador del terreno y el total de oro que Parker podría sacar en la campaña, cada uno de ellos en una línea diferente. En caso de empate en oro con un terreno, priorizará el primer contacto que tiene y al menos una porción de terreno será tierra u oro.

Entrada de ejemplo

```
1
6 6
GTGGGG
GOOGGG
GGGGGG
GTGOOG
GTOTOG
GGOOOG
2
Hoffman 0126 0 1
Patrik 12 4 4
```

Salida de ejemplo

```
Patrik
12
7
```

Límites

- $1 \leq T \leq 100$
- $1 \leq N, M \leq 300$
- C puede tener los valores “O” denotando oro, “T” tierra y “G” terreno del gobierno.
- $1 \leq L \leq N$
- $length(NL)$ tendrá menos de 10 caracteres del alfabeto inglés en mayúsculas y minúsculas.
- $1 \leq ID \leq 10000$
- $1 \leq X \leq N$
- $1 \leq Y \leq M$

Tiempo: 1 segundo



La isla de los programadores

Inspirado en el famoso programa *La isla de las tentaciones*, *La isla de los programadores* se trata de un reality en el que varios programadores viajan al Caribe con sus parejas para poner a prueba su amor. Una vez allí, se separan y conviven durante 1 mes con tentadores que harán temblar los cimientos de cada relación. Al llegar, todos los participantes se sientan en un sofá para ir conociéndose. Cada tentador puede utilizar su encanto con parejas que están a un número de sitios máximo del sofá, que será el alcance de su encanto. Además, cada tentador solo puede romper una pareja como máximo. La organización del programa quiere calcular cuál es el máximo número de parejas que podrán romperse dada una disposición de tentadores y parejas en el sofá.

Entrada

La primera línea contiene 2 enteros N y K , que indican el número de participantes en el reality y el alcance del encanto de los tentadores. Las siguientes N líneas contienen una cadena de caracteres que indica el tipo de participante que se sitúa en cada posición del sofá en el orden en el que se sientan. Esta cadena puede ser "TENTADOR" o "PAREJA".

Salida

Se debe imprimir por consola el número máximo de parejas que podrán romperse dada la disposición del sofá.

Entrada de ejemplo

```
6 2
PAREJA
PAREJA
TENTADOR
TENTADOR
PAREJA
TENTADOR
```

Salida de ejemplo

```
3
```

Límites

- $5 \leq N \leq 10000$
- $1 \leq K \leq 3$

Tiempo: 1 segundo

● D

La peor entrevista de...

GomezRM es un canal de YouTube que tiene como objetivo sacar de contexto entrevistas, con un estilo propio, creando situaciones “graciosas” y ridículas con fines meramente “humorísticos”. En ningún momento se busca ofender ni molestar a ningún creador, artista, etc. y por lo tanto no se debe tomar en serio. El éxito de sus entrevistas sacadas de contexto bajo el título “*La peor entrevista de...*” le ha convertido en tendencia con cada vídeo que publicaba consiguiendo millones de seguidores entre todas sus redes sociales. Nosotros no queremos quedarnos atrás y queremos desarrollar un programa que sea capaz de generar entrevistas fuera de contexto de la misma manera que lo haría GomezRM.

Entrada

La entrada comienza con una línea que contiene 2 números: P y F . P representa el número de preguntas de la entrevista real y F representa el número de frases que se van a utilizar de esa entrevista real para crear nuestra entrevista sacada de contexto.

Como se trata de una entrevista, cada una de las preguntas tiene siempre una respuesta. Por lo tanto, tras esta primera línea vienen $2 \cdot P$ líneas de texto, de las cuales, la primera línea es una pregunta y la segunda es una respuesta. Cabe recalcar que tanto las preguntas como las respuestas tendrán una longitud L y contienen letras del alfabeto inglés en mayúsculas o minúsculas junto a espacios.

Finalmente, la última línea del programa es una secuencia de F números que hacen referencia a las frases del diálogo anterior que se van a usar en el vídeo. Por lo tanto, el número 1 se corresponde con la primera pregunta de la entrevista real; el número 2, con la respuesta a la pregunta 1; el número 3, con la segunda pregunta en la entrevista real; el número 4, a la respuesta a la pregunta anterior, etc. En esta secuencia los números podrán estar en cualquier orden y podrán estar repetidos.

Salida

La salida serán F líneas. Cada línea se corresponde con una de las frases de la entrevista real, y estarán ordenadas en función de la secuencia recibida por entrada de F números.

Entrada de ejemplo

```
5 7
Como esta Leo Messi
Muy bien
y tu familia
Bien tambien
Has jugado este fin de semana
No
Por
Me sancionaron por agresion
Vaya
Esta muy mal
1 10 3 10 7 8 9
```

Salida de ejemplo

```
Como esta Leo Messi
Esta muy mal
y tu familia
Esta muy mal
Por
Me sancionaron por agresion
Vaya
```

Límites

- $1 \leq P, F \leq 50$
- $1 \leq L \leq 70$

Tiempo: 1 segundo



Números de Lychrel

Álex es profesor de matemáticas en el instituto de la ciudad, y quiere intentar ganarse el respeto de los estudiantes enseñándoles un truco de magia. Al terminar la clase de hoy, les ha enseñado el truco de convertir cualquier número en capicúa. Para convertir un número a capicúa, se tiene que revertir ese número y sumarlo con el original. Por ejemplo, $72 + 27$ da 99, así que podemos convertir tanto al 72 como al 27 en capicúas. Manuel, que no es tonto, ha levantado la mano para afirmar que hay números como el 19 donde eso no funciona. $19 + 91$ es 110, y 110 no es capicúa. Ante esa observación, Álex ha explicado que no basta con hacerlo una vez y que puede ser necesario repetir el proceso varias veces. Por ejemplo, si se hace lo mismo con 110 tenemos que $110 + 11$ es 121, ¡que sí que es capicúa! Además, Álex ha afirmado que como máximo hacen falta 25 repeticiones. Sin más preguntas, Álex ha mandado a los alumnos la tarea para el día siguiente, ha dado a cada alumno varios números distintos, y quiere que calculen cuántas veces es necesario sumar el número al revés para llegar a un capicúa. Lo que no saben los alumnos, es que sí que hay números a los que parece imposible llegar a un capicúa, y el truco de magia falla. Álex ha dado uno de esos números a cada alumno, con la esperanza de que dejen de dedicarse a los trabajos de tecnología en el tiempo de su clase. Esos números son llamados números de Lychrel, y hay numerosos ejemplos de ellos. No obstante, los matemáticos siempre hablan de los mismos 4 números, las únicas semillas de Lychrel, a partir de las cuales es posible obtener los demás: Los números 196, el 879, el 1997 y 7059. *Nota: Un número capicúa es aquel que leído al revés es igual que leído del derecho, es decir, un número simétrico.*

Entrada

La entrada comenzará con un número C indicando la cantidad de números que ha mandado Álex en la lista. A continuación, acompañarán C_i números, cada uno en una línea, los números de los que se debe calcular cuántos pasos hacen falta para conseguir un capicúa.

Salida

Para cada número de la entrada se deberá imprimir el número de pasos que hacen falta para conseguir un capicúa, o, alternativamente, un -1, si no es posible conseguir un capicúa a partir de ese número.

Entrada de ejemplo

```
5
784
196
62
55
218
```

Salida de ejemplo

```
2
-1
1
0
2
```

Límites

- $1 \leq C, C_i \leq 10000$

Tiempo: 1 segundo



Evitando lag

Desde que se ha popularizado el trabajo remoto, el estrés de las redes de comunicaciones se ha incrementado substancialmente. Como nuevo becario de la popular empresa Naranjito, te han asignado la tarea de verificar el estado de la red. Disponemos de una red formada exclusivamente por diferentes ordenadores, conectados mediante enlaces. No todos los ordenadores tienen un enlace directo con todos los ordenadores, por lo que es normal establecer rutas a través de las cuales lleguemos a los diferentes ordenadores de la red. Definimos como ruta de calidad una secuencia de saltos entre dos ordenadores en los que la latencia total entre ambos ordenadores es estrictamente menor a 10000. ¿Podrás encontrar los nodos de la red que no son capaces de establecer rutas de calidad entre ellos?

Entrada

La entrada comienza con una línea con dos números: N , el número de ordenadores a continuación, y M , el número de enlaces que existen en nuestra red. En cada una de las siguientes líneas M , aparece la información de cada enlace, de la siguiente forma: 3 números, A , B , L , que representan, respectivamente, nodo origen, nodo destino, latencia entre ambos nodos. Los enlaces son bidireccionales y su latencia es la misma en ambos sentidos. Nota: Se garantiza que existe al menos una ruta entre cada par de ordenadores.

Salida

Para cada caso de prueba, se pide escribir el número de rutas de calidad que faltan para que todos los ordenadores puedan comunicarse con todos. En el caso de ejemplo, 0 -¿1 -¿2, falta un enlace de calidad entre 0 y 2 al ser la menor latencia entre ellos 10000.

Entrada de ejemplo

```
3 2
0 1 5000
1 2 5000
```

Salida de ejemplo

```
1
```

Límites

- $1 \leq N \leq 100$
- $1 \leq M \leq 500000$
- $1 \leq L \leq 2^{30} - 1$

Tiempo: 1 segundo



Plan de movilidad

El nuevo alcalde de Byron, Thous, ha ideado un nuevo plan de movilidad para la ciudad. Uno de los principales problemas que se ha encontrado en sus largos paseos por las calles de su amada ciudad es que los coches y las motos son, en muchas ocasiones, incapaces de convivir: adelantamientos ilegales por parte de las motos, coches que se cambian de carril sin mirar y hacen daño a los motoristas, insultos e improperios... El alcalde tiene una idea para evitar que esto pase: establecer que en determinadas calles solo puedan circular motos y, en otras, solo puedan circular coches. La estructura de Byron es como la de cualquier ciudad: hay calles de un solo sentido (unidireccionales) y calles de dos sentidos (bi-direccionales). En una calle de un solo sentido que conecte los lugares A y B , se podrá circular de A a B pero no de B a A , mientras que en las calles de doble sentido, conectar A a B siempre implica que B también está conectado con A .

Hasta ahora, todo parecía un plan perfecto para el alcalde Thous, hasta que se ha dado cuenta de un pequeño detalle: su novia, Ada, presidenta de Byron, solo tiene coche, mientras que él solo dispone de moto. Esto supone que, para encontrarse, deben encontrar una ruta válida para ambos. Además, como quieren practicar la austeridad, pretenden minimizar el gasto que hacen en gasolina. El gasto en gasolina se mide en base a la longitud de la calle. Por simplicidad, asumimos que el gasto de pasar por una calle es equivalente a su longitud. Tu objetivo es encontrar el lugar de Byron en el que podrán reunirse Thous y Ada minimizando su gasto en gasolina.

Entrada

La entrada contiene la descripción de varios barrios de la ciudad, y comienza con el número de barrios N que se van a describir. Cada descripción comienza con un entero M , denotando el número de calles que habrá en un determinado barrio. A continuación, vendrán N líneas que contendrán:

- La palabra “coche” o la palabra “moto”, en función de si esa calle es transitable por una moto o por un coche.
- Un entero D que indicará si la calle es unidireccional ($D = 1$) o bidireccional ($D = 2$).
- Dos letras, A y B que indican que existe una conexión de A a B (las letras pertenecen al alfabeto inglés). Recuerda que si la calle es de dos sentidos, la conexión existe entre A y B y entre B y A .
- Un entero L que denota la longitud de la calle. Si Thous y Ada se encuentran en el mismo lugar, podemos asumir que se pueden ver con coste 0.

Finalmente, se recibirá una línea con dos caracteres T y A que indican, respectivamente, la posición inicial de Thous y Ada en el mapa.

Salida

Para cada descripción de un barrio, se debe imprimir el lugar donde se encontrarán Thous y Ada y el coste de llegar hasta ese lugar, separados por un espacio. Si existe más de un lugar en el que podrían encontrarse, se deben imprimir en orden lexicográfico, en la misma línea, separados por un espacio antes del coste asociado. Si nunca podrán llegar a encontrarse, deberás escribir “Amor imposible”.

Entrada de ejemplo

```
2
4
moto 2 A D 2
moto 2 A B 2
coche 2 C D 2
coche 2 C B 2
A C
2
moto 1 A B 10
coche 1 C D 20
A D
```

Salida de ejemplo

```
B D 4
Amor imposible
```

Límites

- $1 \leq D \leq 2$
- $1 \leq N \leq 1000$
- $1 \leq M \leq 20$
- $0 \leq L \leq 500$