Concurso de Programación

Entrenamiento AdaByron Madrid 2023

http://www.ada-byron.es

Cuadernillo de problemas

UNIVERSIDAD
POLITÉCNICA
DE MADRID

Realizado en la **Escuela Técnica Superior de Ingenieros Informáticos de la Universidad Politécnica de Madrid**

10 de Marzo de 2023

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

10 de Marzo de 2023

Índice

Autores de los problemas:

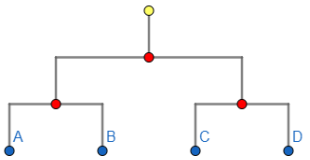
- Catalin Sorin Covaci (Universidad Politécnica de Madrid)
- Isaac Lozano Osorio (Universidad Rey Juan Carlos)
- Jakub Jan Luczyn (Universidad Rey Juan Carlos)

Prueba de los problemas

- Sergio Cavero Díaz (Universidad Rey Juan Carlos)

Tiempo: 1 segundo

A
El Torneo más 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 jugará 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 lineas 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

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

Notas

Con 4 equipos se juega el típico torneo de semi-finales, en la primera fase se jugarán dos partidos: A contra B y C contra D. En la fase final jugarán los ganadores de dichos enfrentamientos, por lo que la cantidad final es $2 + 1 = 3$ partidos. que se organizan en 2 días de competición.

Tiempo: 1 segundo

B
Artículo en Revisión

Aunque parezca mentira los profesores de la universidad también tienen que sacar tiempo para investigar, pues es uno de los requisitos para optar a nuevas plazas e incentivos. Pero... ¿qué es esto de investigar?

Investigar realmente lleva un gran trabajo por detrás: seleccionar una temática, estudiar todos los trabajos previos, analizar los resultados, proponer nuevas contribuciones, demostrarlas, realizar un documento científico y enviar a una revista con prestigio para que se valore. Según la Agencia Nacional de Evaluación de la Calidad y Acreditación (ANECA), el tiempo promedio de que un artículo de investigación sea aceptado (sin contar el trabajo que tiene por detrás) es de 142 días.

Las primeras personas que leen los artículos enviados a la revista son los editores. Estos editores están encargados de asociar revisores a los artículos que reciban. Los revisores reportarán un feedback y el editor notificará a los autores con el veredicto.

Los veredictos más comunes, ordenados de mejor a peor, son los siguientes: aceptado (AC), cambios menores (Cm), cambios mayores (CM) o rechazado (RJ).

Dado el gran volumen de artículos recibidos los tiempos de espera para obtener una respuesta por parte de los investigadores es realmente extenso. ¿Crees que podrías echar una mano a los editores de las revistas? Tú misión será, dado el veredicto de un conjunto de revisores, reportar el peor de todos ellos.

Entrada

Existirá un único caso de prueba. Este caso de prueba tendrá un número R con el total de revisores que han reportado feedback. Por cada revisor se mostrará en una nueva línea su propuesta: "AC","Cm","CM"o "RJ".

Salida

Para cada caso de prueba se mostrará el resultado de las propuestas de los revisores.

Entrada de ejemplo

```
4
AC
Cm
CM
RJ
```

Salida de ejemplo

```
RJ
```

Límites

- $1 \leq R \leq 4$

Tiempo: 1 segundo

C
Visitando el Mundo

Isaac esta harto de su monótona vida, ha decidido dejar todo y comenzar a viajar por el mundo. Para ello ha estado mirando todas las posibles rutas. Esta viendo que puede comenzar en Madrid y desde Madrid visitar Toledo desde Toledo puede ir a Cuenca, etc. O podría ir desde Madrid a Cáceres (obviamente no en tren), después a Mérida, Badajoz, etc. De esta manera Isaac tiene un listado de posibles rutas, pero lo que no sabe es el total de ciudades que va a visitar. Por lo que le da mucha pereza contarlas y quiere crearse un algoritmo que lo haga por el.

Entrada

El primer número T, corresponde al número de casos de prueba. Posteriormente tendremos otro numero N siendo el número de movimientos de una ciudad a otra. Finalmente tendremos N pares de números A y B los cuales marcan que desde la ciudad A se puede ir a la ciudad B y también desde B podemos ir hasta A.

Salida

Para cada caso de prueba imprimir dada la ruta el número de ciudades que Isaac visitará.

Entrada de ejemplo

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

Salida de ejemplo

```
6
4
```

Límites

- $1 \leq T \leq 2500$
- $0 \leq N \leq 10000$
- $1 \leq T \cdot N \leq 150000$
- $0 \leq A,B \leq 10^{18}$

Tiempo: 2 segundos

D
Que gane el meXOR

Alicia y Bobo están siempre molestando en clase. Para mantenerles calladitos y ocupados, el profesor les regala un string binario s (compuesto por ceros y unos) de longitud n y les presenta un juego. En cada turno, el jugador correspondiente elige el primer o el último carácter y lo elimina del string. Los jugadores se van turnando, y pierde aquel que tras su jugada haga que el XOR de todos los valores eliminados hasta ahora sea 0.

Formalmente, sea $x := 0$ inicialmente. En un turno, un jugador debe hacer lo siguiente:

1. Elige el carácter s_1 o el carácter s_n del string. Sea c su valor.
2. Actualiza $x := x \oplus c$, donde $\oplus$ es el XOR. Elimina el carácter del string, por lo que se reduce su longitud ($n := n - 1$).
3. Si x es 0, el jugador actual pierde y acaba la partida. En caso contrario, pasa el turno al otro jugador y se vuelve al paso 1, mientras el string no esté vacío ($n > 0$).

En el primer turno empieza Alicia. Determina quién gana si ambos juegan de manera óptima.

Nota: El XOR de dos bits, denotado $\oplus$, es 0 si son iguales, y 1 si son distintos, es decir, $1 \oplus 1 = 0 \oplus 0 = 0$ y $0 \oplus 1 = 1 \oplus 0 = 1$.

Entrada

La entrada consiste en varios casos de prueba. La primera línea contiene un único entero t ($1 \leq t \leq 2 \cdot 10^5$) — el número de casos de prueba. Cada caso de prueba está formado por dos líneas.

La primera línea contiene un único entero n ($1 \leq n \leq 2 \cdot 10^5$) — la longitud del string binario s.

La segunda línea contiene el string $s_1 s_2 \ldots s_n$ ($0 \leq s_i \leq 1$) — los n caracteres binarios.

Se garantiza que la suma de n sobre todos los casos de prueba no excederá $2 \cdot 10^5$.

Salida

Para cada caso de prueba, imprime "`ALICIA`" o "`BOBO`", según quién sea el ganador, o "`EMPATE`", si el string se queda vacío antes de decidirse un ganador.

Entrada de ejemplo

```
2
1
0
2
01
```

Salida de ejemplo

```
BOBO
EMPATE
```

Notas

En el primer ejemplo, Alicia no tiene más remedio que elegir el único carácter, que es un 0, así que pierde.

En el segundo ejemplo, Alicia está obligada a elegir el 1 para no perder en la primera jugada, tras lo cual Bobo elige el 0 sin más opciones, y el XOR sigue siendo 1, así que nadie gana.

Tiempo: 1 segundo

E
A la vuelta de la esquina

Unos amigos están buscando piso para vivir mientras estudian en la universidad. Pero, como ya sabemos todos, los estudios no son su prioridad. En concreto a estos amigos les gusta quedar en algún bar y ponerse al día de las hazañas que ha conseguido cada uno. Ahora que pueden elegir ellos donde van a vivir, quieren un piso que tenga el mayor número de bares *a la vuelta de la esquina*.
Tienen la suerte de que la zona en la que están buscando el piso tiene las calles dispuestas siguiendo una cuadrícula, por lo que un bar está *a la vuelta de la esquina* respecto a una calle si, caminando por esta en uno de sus dos sentidos, al llegar a la primera intersección uno puede verlo a su izquierda o a su derecha. Ten en cuenta que si tienes que cruzar otra intersección para llegar al bar está demasiado lejos para estar *a la vuelta de la esquina*. Ahora lo único que queda es encontrar en la ciudad los mejores sitios para alquilar el piso. ¿Puedes ayudarles con esa tarea?

Entrada

La entrada estará compuesta por un único caso de prueba. En la primera línea vendrán dos números, N y M, indicando la altura y la anchura del plano de la ciudad. A continuación vendrán N líneas con M caracteres cada una. "#" indica un edificio, "-" indica una calle vacía ,"*" indica una calle con un bar y "+" indica una intersección.
Se sabe que tanto en la esquina superior izquierda e inferior derecha del plano se encuentra un edificio, seguirán el mismo patrón que el ejemplo que se muestra a continuación y que ninguna de las 4 casillas vecinas de una calle es un bar. También se sabe que nunca va a haber un bar en una intersección.

Salida

Tienes que imprimir el plano de la cuidad que has leído, sustituyendo los caracteres "-" y "*" por el número de bares *a la vuelta de la esquina* que hay en esas posiciones. Se garantiza que este número será menor que 10 (si no el plano quedaría un poco feo).

Entrada de ejemplo

```
5 5
#-#*#
-+-+-
#-#*#
-+-+*
#*#*#
```

Salida de ejemplo

```
#0#0#
0+2+2
#0#1#
1+3+2
#0#1#
```

Límites

- $1 \leq N, M \leq 999$