

Torneo Chileno de Programación 2017

12 de Agosto de 2017

Sesión de Contest

Este cuadernillo contiene 11 problemas; las páginas están numeradas de 1 a 12.

Este conjunto de problemas se utiliza simultáneamente en las competencias efectuadas en:

Iquique (UNAP), Antofagasta (UCN), Coquimbo (UCN), Valparaíso (UTFSM), Santiago (UTEM, USACH),
Curicó (UTalca), Concepción (UdeC), Temuco (UFro), y Valdivia (UACH)

Informaciones generales

Las siguientes condiciones son válidas para todos los problemas, salvo que sea explícitamente indicado.

Nombre del programa

1. Su solución debe tener el nombre `codename.c`, `codename.cpp` or `codename.java`, `codename.py2`, `codename.py3`, donde `codename` es la letra mayúscula que identifica al problema.

Entrada (Input)

1. La entrada debe ser leída desde la entrada estándar (*standard input*).
2. La entrada contiene un único caso de prueba, el cual se describe utilizando una cantidad de líneas que depende del problema. No hay más datos en la entrada que lo indicado en el enunciado.
3. Cuando una línea de datos contiene varios valores, éstos se separan utilizando exactamente *un* espacio entre ellos. Ningún otro espacio aparece en la entrada. La entrada no contiene líneas en blanco.
4. Para efectos de leer caracteres alfabéticos, se utiliza el alfabeto inglés. No se utilizan caracteres con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ã, é, Ì, ô, Ü, ç, etcétera).
5. Todas las líneas, incluyendo la última, terminan con la marca usual de fin de línea.

Salida (Output)

1. La salida debe ser escrita en la salida estándar (*standard output*).
2. El resultado del caso de prueba debe aparecer en la salida utilizando una cantidad de líneas que depende del problema. No se deben escribir más datos en la salida que los indicados por el problema.
3. Cuando una línea de resultados contiene varios valores, éstos deben ser separados utilizando exactamente *un* espacio entre ellos. Ningún otro espacio debe aparecer en la salida. No debe haber líneas en blanco en la salida.
4. Para efectos de escribir caracteres alfabéticos, se utiliza el alfabeto inglés. No se utilizan caracteres con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Todas las líneas, incluyendo la última, deben terminar con la marca usual de fin de línea.
6. Al momento de escribir números reales, éstos deben ser redondeados al racional más cercano usando la cantidad de dígitos posterior al punto decimal que se especifica en el enunciado. Cada caso de prueba se ha diseñado de manera que no haya empates en el redondeo.

Tiempo límite

1. El tiempo límite informado para cada problema corresponde a la entrada descrita en el enunciado, y *no* a múltiples instancias de ella.

Comité de problemas

Las siguientes personas participaron en la elaboración del conjunto de problema mediante la creación y mejora de enunciados, soluciones, casos de pruebas y verificaciones de entrada y salida:

Sebastián Barbieri
Rodrigo Basoalto
Juan Pablo Canepa
Nico Lehmann
Andrés Letelier
Federico Meza
Francisco Rojas
Cristian Ruz

Problem A – Alianzas de aniversario

Autor: Federico Meza

El Liceo Técnico Computín de Pelotillehue, también conocido como Liceo TCP, ha dividido sus cursos superiores en tres alianzas que competirán durante las celebraciones de aniversario. Para ganar deben conseguir la mayor cantidad de puntos en una actividad de beneficencia, a través de la cual entregarán alimentos a un albergue.

Cada alianza cuenta con un presupuesto para la compra de los alimentos. Este dinero fue recaudado con mucho esfuerzo, a través de diversas actividades que han desarrollado en las últimas semanas. Sin embargo, dado que los distintos tipos de alimentos entregan distintos puntajes en la competencia, la alianza con más presupuesto no tiene garantizada la victoria, pues otra alianza con menos dinero podría lograr una mejor mezcla de alimentos y obtener mayor puntaje.

Una mezcla corresponde a una secuencia de números enteros no negativos que representa la cantidad de ítems de cada tipo de alimentos que se comprarán. Para que una mezcla sea válida, su costo debe ser menor o igual al presupuesto con que cuenta la alianza.

Puesto que la cantidad de tipos de alimento distintos es muy grande, una de las alianzas decidió escribir un programa para encontrar la configuración que le saque mayor provecho a su dinero.

Entrada

La entrada consiste en 4 líneas. La primera línea contiene un número entero positivo P ($1 \leq P \leq 10000$) que corresponde al presupuesto con que cuenta la alianza. La segunda línea contiene un número entero positivo N ($1 \leq N \leq 25$), que determina la cantidad de tipos de alimentos considerados. La tercera línea contiene N valores no negativos, separados por un espacio, representando el costo asociado a un ítem de cada uno de los N tipos de alimento. Finalmente, la cuarta línea contiene N valores no negativos, separados por un espacio, representando el puntaje asociado a un ítem de cada uno de los N tipos de alimento.

Salida

La salida corresponde a una línea con dos valores enteros no negativos, separados por un espacio, correspondientes al puntaje total y al costo total de la mezcla óptima para la alianza.

Entrada de ejemplo 1 17 3 2 3 5 50 80 120	Salida para la entrada de ejemplo 1 450 17
Entrada de ejemplo 2 19 3 5 3 3 10 1 5	Salida para la entrada de ejemplo 2 35 18

Problem B – Hermosas alpacas

Autor: Sebastián Barbieri

¡Científicos del centro para el Tratamiento y Control de la Población, TCP, han realizado un impresionante descubrimiento! Nuestras queridas alpacas, en lugar de portar un código genético compuesto de bases adenina (A), citosina (C), guanina (G) y timina (T), poseen bases completamente distintas: su ADN está compuesto de bases (A), (C), (L) y (P) donde (L) es lanina y (P) es preciosina.

Más aún, los investigadores del TCP descubrieron que el código genético de las alpacas es extremadamente estructurado. Este se puede codificar como una secuencia sobre el alfabeto $\{A, C, L, P\}$ aplicando algunas reglas. Partiendo con la letra A la secuencia que describe el ADN puede generarse aplicando N veces el siguiente conjunto de reglas de forma simultánea:

- Reemplazar cada ocurrencia de (A) por (AL)
- Reemplazar cada ocurrencia de (L) por ($PACA$)
- Reemplazar cada ocurrencia de (P) por (CP)
- Reemplazar cada ocurrencia de (C) por (PC)

Por ejemplo, si $N = 3$ la secuencia obtenida será $ALPACACPALPCAL$:

$$A \longrightarrow AL \longrightarrow ALPACA \longrightarrow ALPACACPALPCAL.$$

Los científicos están estudiando la hermosura de las alpacas. Hasta el momento han descubierto que existen M tipos de hermosura distintas. Y más aún, también han logrado relacionar el tipo de hermosura de una alpaca con la cantidad de veces que la subcadena ($ALPACA$) aparece en su secuencia de ADN. En particular, si ($ALPACA$) aparece D veces en la secuencia de ADN de una alpaca, entonces su tipo de hermosura está dado por el resto de la división de D por M . ¿Podrías ayudar a nuestros científicos a determinar qué tan bella es una alpaca en particular?



Entrada

La entrada consiste en una única línea que contiene dos números N y M separados por un espacio, donde N indica el número de iteraciones que describen el ADN de la alpaca ($1 \leq N \leq 10^{15}$), y M es la cantidad de tipos de hermosura ($2 \leq M \leq 10^9$).

Salida

Un único entero conteniendo el tipo de hermosura de la alpaca ($D \bmod M$).

Entrada de ejemplo 1 1 187	Salida para la entrada de ejemplo 1 0
Entrada de ejemplo 2 2 187	Salida para la entrada de ejemplo 2 1
Entrada de ejemplo 3 3 187	Salida para la entrada de ejemplo 3 1
Entrada de ejemplo 4 10 5000	Salida para la entrada de ejemplo 4 171
Entrada de ejemplo 5 987654321 123456789	Salida para la entrada de ejemplo 5 41191906

Problem C – Cachipún

Autor: Federico Meza

Martina y Benjamín son amigos desde el jardín de niños, y nunca logran ponerse de acuerdo. Hasta la decisión más sencilla, como cuál película ver, o dónde ir a tomar un café, termina en una discusión. Su amistad ha perdurado a través de los años pues desde niños encontraron un sistema para resolver sus disputas. Cada vez que tienen que decidir algo, y no logran ponerse de acuerdo, juegan al TCP y quien gana decide. El TCP, o *Tijera–Cartón–Piedra*, es un juego ancestral que se conoce también en algunos países como *Cachipún*, o como *Piedra–Papel–Tijera*.

Recientemente Martina le confesó a Benjamín que ha guardado un registro diario de las veces que han jugado y de quién ha ganado cada vez. Si bien esto le pareció algo extraño a Benjamín, decidió que era una oportunidad para ver en cuántos días la suerte había favorecido a su amiga y en cuántos a él. Dado lo extenso del registro, debió escribir un programa para saberlo.

Entrada

La entrada consiste en una línea que contiene un entero positivo N ($1 \leq N \leq 100000$), seguido de N líneas donde cada una contiene una serie de números. Cada serie comienza con un número entero positivo M ($1 \leq M \leq 100$), seguido de M números cuyo valor puede ser 0 ó 1. Estos valores corresponden a los M juegos que tuvieron lugar ese día. Cuando aparece un 0 debe interpretarse como que Martina fue la vencedora de ese juego. Por el contrario, cuando aparece un 1 significa que Benjamín ganó esa partida.

Salida

La salida contiene una única línea con 3 números enteros no negativos. El primero corresponde a la cantidad de días en que Martina resultó ganadora, es decir, en que ganó más partidas que su amigo. El segundo número corresponde a la cantidad de días en que Benjamín resultó ganador. Finalmente, el último número indica la cantidad de días en que se produjo un empate.

Entrada de ejemplo 1	Salida para la entrada de ejemplo 1
4 7 1 0 1 1 1 0 1 3 0 0 0 4 1 0 1 0 5 1 1 1 1 0	1 2 1

Problem D – Cocina de programadores

Autor: Rodrigo Basoalto

El Taller de Cocina de Programadores (TCP) celebra anualmente una cena de camaradería. Este año asistirán N comensales. La cena tiene un menú predefinido con varias opciones de platos y cada comensal ha escogido con anticipación su plato preferido. Cada plato está descrito por una receta con sus ingredientes y el tiempo que toma prepararla. Algunos ingredientes son *ingredientes básicos* que se encuentran disponibles en cantidades ilimitadas en la mesa de los cocineros. Los otros ingredientes se denominan *ingredientes compuestos*, y deben ser preparados siguiendo una receta a partir de otros ingredientes. Un ingrediente compuesto puede ser utilizado como un plato para un comensal o como ingrediente para otros ingredientes compuestos.

El equipo de cocina ha sido conformado por un selecto grupo de programadores que pueden preparar cualquier cantidad de ingredientes compuestos en paralelo (pues son programadores), pero ninguna receta puede iniciarse si no están todos sus ingredientes disponibles previamente. Por ejemplo, para preparar una *pizza* los ingredientes son *masa*, *queso*, *salsa de tomates* y *pepperoni*, y su tiempo de preparación es 20 minutos; de ellos, la *masa* y la *salsa de tomates* son ingredientes compuestos. Para preparar la *salsa de tomates* se requiere *tomates*, *sal*, *albahaca* y *pimienta*, y su tiempo de preparación es de 30 minutos. La receta para la *masa* requiere *harina*, *levadura*, *sal* y *aceite*, y su tiempo de preparación es de 90 minutos. Los ingredientes *masa* y *salsa de tomates* deben ser preparados antes de poder preparar la *pizza*, sin embargo ambos pueden prepararse simultáneamente de manera que el tiempo mínimo para preparar la *pizza* es $\max(\{90, 30\}) + 20 = 110$ minutos.

Entrada

La entrada contiene varias líneas. La primera línea contiene un entero I ($0 < I \leq 10^5$) correspondiente a la cantidad de ingredientes básicos y compuestos. A continuación siguen I líneas cada una describiendo un ingrediente. Cada línea comienza con 2 enteros no negativos, T y D que representan respectivamente el tiempo en minutos de preparación del ingrediente ($0 \leq T \leq 1000$), y la cantidad de ingredientes requeridos para prepararlo ($0 \leq D \leq 10$). Le sigue en la misma línea el nombre del ingrediente descrito y, a continuación una secuencia de D ingredientes separados por espacios donde cada ingrediente está descrito como una cadena de largo máximo 20 formada por caracteres alfabéticos y el símbolo '_'. Los ingredientes básicos son aquellos en que $T = D = 0$. Luego de la lista, hay un entero positivo N ($0 < N \leq 10^5$) que indica la cantidad de comensales. Finalmente hay N líneas donde cada una contiene el nombre de un plato escogido por un comensal. Los platos siempre corresponderán a un ingrediente compuesto. Se garantiza que siempre será posible preparar los platos en tiempo finito.

Salida

La salida consiste en una línea con un único entero que indica cuántos minutos le tomará a los cocineros preparar la cena.

Entrada de ejemplo 1	Salida para la entrada de ejemplo 1
12 0 0 Sal 0 0 Tomates 0 0 Harina 0 0 Levadura 0 0 Aceite 0 0 Albahaca 0 0 Pimienta 0 0 Queso 0 0 Pepperoni 30 4 Salsa Tomates Sal Albahaca Pimienta 90 4 Masa Harina Levadura Sal Aceite 20 4 Pizza Masa Salsa Queso Pepperoni 1 Pizza	110

Problem E – El jardín de Gardenio

Autor: Juan Pablo Canepa

Gardenio el jardinero es especialista en sistemas de riego y ha creado un nuevo tipo de aspersor de riego capaz de regar uniformemente una superficie rectangular, arbitrariamente configurable en ancho y largo a través de un delicado sistema mecánico. Gardenio es extremadamente popular por sus habilidades y es contratado por diversos empleadores para instalar su sistema de riego en sus prados. Cada prado tiene todo tipo de obstáculos y diferentes formas, lo que obliga a Gardenio a ubicar sus aspersores en forma estratégica de manera de cubrir toda la superficie en cuestión. Por sobre todo, Gardenio es extremadamente cuidadoso de no malgastar agua y por lo tanto configura siempre sus aspersores para que cubran toda la superficie en cuestión, pero tratando de minimizar las áreas que son regadas por más de un aspersor. Para verificar su trabajo, Gardenio lo ha contratado a usted para desarrollar un programa, que dado un conjunto de coordenadas que describen las zonas cubiertas por los distintos aspersores (en metros), le reporte el área de la zona más grande que es regada simultáneamente por cualquier *par* de aspersores al mismo tiempo.

Entrada

Cada archivo de entrada contiene un único caso de prueba. La primera línea del archivo contiene un entero positivo N_r ($0 \leq N_r \leq 1000000$) que representa la cantidad total de áreas cubiertas por aspersores. A continuación hay N_r líneas, donde cada línea contiene cuatro enteros positivos que representan las coordenadas de un área, siguiendo el orden $x_{izquierda}$, y_{abajo} , $x_{derecha}$, y_{arriba} . Todos los valores son menores o iguales a 2^{31} .

Salida

Para cada archivo de entrada debe escribir un valor entero que representa el área de la zona más grande existente regada simultáneamente por cualquier *par* de aspersores.

Entrada de ejemplo 1 4 0 0 10 10 0 10 0 20 10 10 20 20 10 0 20 10	Salida para la entrada de ejemplo 1 0
Entrada de ejemplo 2 3 10 5 25 20 25 5 35 20 15 15 30 25	Salida para la entrada de ejemplo 2 50

Problem F – Interferencia en la WiFi

Autor: Nico Lehmann

Calle Larga es una comuna situada en el sector precordillerano de la región de Valparaíso, constituida esencialmente por una calle de gran longitud poblada de casas a lo largo de toda su extensión. Los vecinos de Calle Larga están preocupados porque la calidad de su conexión Wi-Fi ha empeorado considerablemente en los últimos años y debido a esto no pueden ver Game of Thrones tranquilamente.

Cada una de las casas en la calle posee un enrutador cuyas características determinan el alcance de su señal. La señal de cada enrutador corresponde a un círculo determinado por un radio r . Cuando dos señales se intersectan se produce interferencia lo que degrada considerablemente la conexión.

En la última reunión de la junta de vecinos, la comunidad determinó que la única solución para su problema era compartir la conexión a internet entre algunos vecinos para así poder prescindir de algunos enrutadores. Lamentablemente, la comisión encargada de poner el plan en marcha está teniendo problemas para evaluar qué enrutadores son mejores candidatos para ser mantenidos. Específicamente, dado un enrutador les gustaría determinar la cantidad de señales que este contiene completamente. Si la señal de un enrutador contiene completamente la señal de muchos enrutadores entonces este es un buen candidato para ser mantenido.

Entrada

La primera línea contiene dos enteros N y Q ($0 \leq N \leq 2 \times 10^5, 0 \leq Q \leq 5 \times 10^4$) correspondientes respectivamente a la cantidad total de enrutadores y la cantidad de enrutadores por los cuales se hará una consulta. Cada una de las siguientes N líneas describe un enrutador. La línea i -ésima describe el enrutador i con dos enteros p y r ($0 \leq p \leq 10^9, 0 < r \leq 10^9$) que representan respectivamente la posición del enrutador en la calle y el radio de alcance de su señal. No habrá dos enrutadores en la misma posición. Las siguientes Q líneas contiene la descripción de una consulta. Cada consulta está descrita con un entero i ($1 \leq q \leq N$) indicando que se desea determinar la cantidad de señales que están completamente contenidas en la señal del enrutador i .

Salida

Por cada consulta debe imprimirse un entero en una línea separada. Cada entero debe corresponder a la cantidad de señales que están contenidas completamente en la señal del enrutador de la consulta.

Entrada de ejemplo 1	Salida para la entrada de ejemplo 1
5 2	1
0 2	2
1 3	
4 1	
6 2	
6 3	
2	
5	

Problem G – Reuniones bipartidistas

Autor: Nico Lehmann

La nación de Nlogonia posee un sistema bipartidista donde los dos partidos dominantes son el Tratado Ciudadano Popular (TCP) y la Union Demócrata Patriótica (UDP). Antes de las elecciones de este año las direcciones de ambos partidos han decidido unir fuerzas para así formar un gran partido único nacional. Con el fin de dilucidar los detalles de la fusión, los altos mandatarios de ambos partidos deben realizar una serie de reuniones bilaterales. Debido a la alta cantidad de reuniones, la administración de los partidos está teniendo problemas para organizarlas.

Nlogonia está distribuida en N ciudades conectadas por $N - 1$ carreteras que pueden ser recorridas en ambos sentidos. Las ciudades están distribuidas de forma tal que siempre es posible viajar entre dos ciudades utilizando las carreteras. En cada ciudad de Nlogonia existe una sala de reuniones de la cual pueden disponer los mandatarios. Cuando dos mandatarios ubicados en las ciudades a y b desean tener una reunión, deben determinar un punto intermedio para realizarla, eligiendo la sala de reuniones de alguna de las ciudades en el camino que hay entre a y b . Cada sala tiene un costo asociado de uso y para escoger cual utilizar los mandatarios deben seguir algunas restricciones monetarias. Dependiendo de la importancia, cada reunión tendrá un presupuesto asignado y esta no podrá realizarse en una sala con un costo asociado mayor. Como un precio mayor está asociado a mejores condiciones, los mandatarios siempre decidirán realizar la reunión en la sala más cara posible.

Cada reunión tiene una duración determinada y mientras se realiza una reunión su sala no puede ser utilizada para otras reuniones. Una vez terminada una reunión, la sala queda inmediatamente habilitada para ser usada nuevamente. Es factible que en cierto momento sea imposible realizar una reunión, pues no hay salas disponibles para el presupuesto asignado. En caso de no ser posible realizar una reunión, esta simplemente será cancelada.

Teniendo en cuenta estas restricciones y dada la descripción de las reuniones, tu tarea es asignar una sala para cada una de las reuniones o determinar que no se pueden realizar.

Entrada

La primera línea de la entrada contiene un entero N ($1 \leq N \leq 10^5$) correspondiente a la cantidad de ciudades en Nlogonia. Cada ciudad es identificada con un entero entre 1 y N . La siguiente línea contiene N enteros $c_1 \cdots c_N$, donde c_i ($0 \leq c_i \leq 10^9$) corresponde al costo de realizar la reunión en la sala de reuniones de la ciudad i . No habrá dos ciudades con el mismo costo asociado a utilizar su sala de reuniones. A continuación siguen $N - 1$ líneas describiendo las carreteras. Cada una de estas líneas contiene dos enteros u y v ($1 \leq u, v \leq N$) indicando que existe una carretera entre las ciudades u y v . Posteriormente continúa una línea con un entero R ($1 \leq R \leq 2 \times 10^4$) correspondiente a la cantidad de reuniones. Cada una de las siguientes R líneas contiene la descripción de una reunión. Una reunión es descrita con cinco enteros t, a, b, d y p . El entero t ($0 \leq t \leq 10^9$) corresponde a la hora en que la reunión debe llevarse a cabo, representada como la cantidad de segundos desde el inicio del programa. Los enteros a y b ($1 \leq a, b \leq N$) indican que la reunión es entre mandatarios en las ciudades a y b . El entero d ($0 < d \leq 10^9$) corresponde al tiempo en segundos de duración de la reunión. Finalmente el entero p ($0 < p \leq 10^9$) corresponde al presupuesto asignado para la reunión. Las reuniones serán entregadas en orden creciente según su hora de inicio.

Salida

Por cada descripción de reunión en la entrada, la salida debe contener un entero indicando la ciudad en que se realizará la reunión considerando las restricciones del problema. Notar que mientras la reunión aún no haya acabado esta sala no podrá ser usada en las siguientes reuniones. En caso de no ser posible realizar una reunión debe imprimirse el valor -1.

Entrada de ejemplo 1	Salida para la entrada de ejemplo 1
10	9
1 10 9 8 18 7 17 2 12 11	5
2 1	-1
3 2	-1
4 3	10
5 4	
6 5	
7 6	
8 2	
9 6	
10 2	
5	
3 2 9 9 12	
9 5 2 5 20	
12 2 5 2 6	
28 3 3 2 6	
43 10 5 2 15	

Problem H – El misterio de la mano

Autor: Francisco Rojas

Arturo es un escritor aficionado de novelas de misterios. Hace pocos días ha terminado su último libro y, hasta ahora, su mejor obra, titulado "El Misterio de la Mano". No solo eso, Arturo acaba de firmar un contrato con una editorial para que lo ayude a comercializar su libro y lograr que el mundo conozca su trabajo.

Una de las decisiones más importantes que Arturo debe tomar es el tamaño en el que su libro será impreso. Para ello, le ayudaría mucho saber la cantidad de páginas que su libro tendría en cada formato.

Dependiendo del formato del libro, cada página puede contener hasta Z caracteres. Sin embargo, la editorial exige que cada página incluya el título del libro de largo Y , además del número de página actual y la cantidad de páginas del libro separados por el carácter `'/'`. Para escribir el número de página actual se usa la misma cantidad de espacios que el número de páginas, anteponiendo los `'0'` que sean necesarios. Por ejemplo, la página "uno de diez" se escribe `01/10`, ocupando cinco caracteres.

Arturo sabe que su libro tiene X caracteres. Ayuda a Arturo a determinar la mínima cantidad de páginas que tendría su libro, dependiendo del formato que escoja.

Entrada

La primera línea de la entrada contiene tres enteros X, Y y Z , donde X ($1 \leq X \leq 10^5$) es el número de caracteres en el libro, Y ($0 \leq Y \leq 10^2$) es el largo del título, y Z ($0 \leq Z \leq 10^5$) es el número de caracteres que se puede escribir en una página. La entrada está construida de manera que siempre es posible generar un libro.

Salida

Un valor entero que representa el mínimo número de páginas que puede tener el libro.

Entrada de ejemplo 1 1000 12 100	Salida para la entrada de ejemplo 1 13
Entrada de ejemplo 2 100 10 113	Salida para la entrada de ejemplo 2 1
Entrada de ejemplo 3 1000 91 100	Salida para la entrada de ejemplo 3 500
Entrada de ejemplo 4 100000 12 100	Salida para la entrada de ejemplo 4 1266

Problem I – El reino de los superpares

Autor: Federico Meza

En un momento histórico para la reivindicación de los derechos de los números pares (que históricamente han sido reprimidos por los separatistas impares), se ha procedido a firmar el Tratado Continental de Pares, o TCP, en el que se reafirman los derechos y obligaciones de ambas partes para con los otros.

Una sección del documento describe como se han de determinar los miembros de los grupos dirigentes de ambos sectores. En el caso de los números pares, esta responsabilidad ha recaído en un selecto grupo de miembros que son vistos con reverencia por el resto de la comunidad; su solo nombre inspira respeto. Ellos son conocidos como los *superpares*.

Un número entero positivo es un súperpar si es *par* y la suma de todos sus dígitos *también es par*. Por ejemplo, los números 4, 578 y 6000 son súperpares dado que son pares y sus dígitos suman 4, 20 y 6 respectivamente. Por otra parte, los números 12, 679 y 111 no son súperpares; en el primer caso, a pesar de que 12 es par, sus dígitos suman 3; en el segundo caso, a pesar de que los dígitos de 679 suman 22, el número es impar; en el tercer caso, 111 no cumple ninguna de las condiciones dado que es impar y sus dígitos suman 3.

Se debe escribir un programa que identifique números que sean súperpares.

Entrada

La entrada consiste de un entero positivo de no más de 100 dígitos, que deberá ser evaluado para saber si es un súperpar o no.

Salida

Una línea de salida con la palabra **superpar** si es que el entero es un súperpar, o con la palabra **no** si no lo es.

Entrada de ejemplo 1 4	Salida para la entrada de ejemplo 1 superpar
Entrada de ejemplo 2 679	Salida para la entrada de ejemplo 2 no
Entrada de ejemplo 3 20000000000000002	Salida para la entrada de ejemplo 3 superpar
Entrada de ejemplo 4 11111111111112	Salida para la entrada de ejemplo 4 no

Problem J – No EsCrlbAn Asl

Autor: Francisco Rojas

Rita es una estudiante de intercambio que viene desde México a Chile. Hasta ahora su experiencia ha sido maravillosa ya que ha conocido muchos chilenos, lo que le ha permitido conocer sus modas. Una de las últimas modas que ha descubierto es una divertida forma de escribir.

De acuerdo a esta moda, todos los caracteres alfabéticos de los textos son intercalados entre mayúsculas y minúsculas, partiendo siempre con una mayúscula. Por ejemplo, “Hola amigos” lo escriben como “HoLa AmIgOs”.

Rita quiere encajar con sus nuevos amigos chilenos, pero le cuesta mucho escribir de esta forma. Escribe un programa que ayude a Rita transformando el texto por ella.

Entrada

La primera línea de la entrada contiene un entero positivo N ($N < 10000$). La segunda línea de la entrada contiene un texto de largo N . El texto puede incluir caracteres del alfabeto inglés, dígitos, espacios en blanco y símbolos de puntuación del conjunto $\{!, \$, \%, \&, (,), *, +, -, ., ,, :, ;, =, ?, @, _ \}$.

Salida

Por cada archivo de entrada debe generar escribir un línea de salida que contenga el texto transformado usando las reglas definidas anteriormente.

Entrada de ejemplo 1 15 hola, que hace?	Salida para la entrada de ejemplo 1 HoLa, QuE hAcE?
Entrada de ejemplo 2 29 amigo numero 1 !!! Escribeme!	Salida para la entrada de ejemplo 2 AmIgO nUmErO 1 !!! eScRiBeMe!

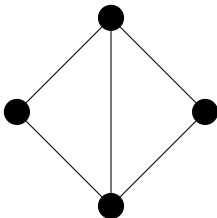
Problem K – Telégrafo

Autor: Andrés Letelier

La compañía Telegráfica Comunal de Pirihueico (TCP) tiene una red sumamente antigua con enlaces bidireccionales que unen pares de estaciones de forma directa. Cada telegrafista es capaz de enviar mensajes desde su propia estación hasta las estaciones con las cuales existe un enlace directo.

A veces uno de los enlaces se corta, y no es posible enviar mensajes directamente de una ciudad a otra. TCP (la compañía) tiene un protocolo de comunicaciones basado en TCP (el protocolo) para reenviar mensajes cuando un enlace entre dos ciudades a y b falla, pero los telegrafistas de TCP son muy flojos. Si no existe un camino de a a b que pasa por exactamente una ciudad adicional c , entonces el telegrafista se da por vencido y se olvida del mensaje.

Tulio Cornelio Policarpo, el supervisor de la red TCP, quiere asegurar que la red sea lo más estable posible. Para esto, define en primer lugar un triángulo como un conjunto de tres estaciones a , b y c , para las cuales existe un enlace directo entre a y b , uno entre b y c y otro entre c y a . Dada esta definición, la fortaleza de la red se define como la cantidad de triángulos que hay en esta. Por ejemplo, la red de la imagen tiene 2 triángulos:



Ayuda a Tulio Cornelio Policarpo a determinar la fortaleza de su red.

Entrada

La entrada comienza con dos enteros N ($1 \leq N \leq 10^5$) y M ($0 \leq M \leq 10^5$) que representan respectivamente el número de estaciones y el número de enlaces de la red. Cada estación es identificada con un entero entre 1 y N . A continuación vienen M líneas cada una de las cuales contiene dos enteros a y b ($1 \leq a < b \leq N$), que indican que existe un enlace bidireccional entre las estaciones a y b . Cada enlace aparece exactamente una vez en la entrada.

Salida

La salida consiste en un único entero que representa la fortaleza de la red.

Entrada de ejemplo 1	Salida para la entrada de ejemplo 1
4 5 1 2 1 3 2 3 2 4 3 4	2