

Torneo Argentino de Programación

26 de septiembre de 2015

- Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
- Escuela de Tecnologías de la Información y las Comunicaciones
Universidad Nacional de Chilecito
- Facultad de Ciencias Exactas, Físico-Químicas y Naturales
Universidad Nacional de Río Cuarto
- Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario
- Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
- Facultad de Informática
Universidad Nacional de La Plata
- Facultad de Informática
Universidad Nacional del Comahue
- Facultad de Matemática, Astronomía y Física
Universidad Nacional de Córdoba
- Facultad Regional Resistencia
Universidad Tecnológica Nacional

Sesión de Competencia

Este conjunto contiene 11 problemas; las páginas están numeradas de 1 a 22.

Información General

Entrada

1. La entrada se debe leer de 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 otros datos en la entrada.
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. No hay líneas en blanco.
4. No hay letras con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ã, é, Ì, ô, Ü, ç, etcétera).
5. Todas las líneas, incluyendo la última, tienen la marca usual de fin de línea.

Salida

1. La salida se debe escribir 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 debe haber otros datos en la salida.
3. Cuando una línea de resultados contiene varios valores, éstos se deben separar utilizando exactamente *un* espacio entre ellos. Ningún otro espacio debe aparecer en la salida. No debe haber líneas en blanco.
4. No debe haber letras con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ã, é, Ì, ô, Ü, ç, etcétera).
5. Todas las líneas, incluyendo la última, deben tener la marca usual de fin de línea.
6. Para escribir números reales, redondearlos al racional más cercano con la cantidad de dígitos luego del punto decimal que se especifica en el enunciado. El caso de prueba es tal que no va a haber empates en el redondeo.

Tiempo límite

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

Colaboradores

Los problemas de este conjunto fueron preparados con la colaboración de Pablo Blanc, Brian Curcio, Pablo A. Heiber, Ramiro Lafuente, Juan Cruz Piñero, Federico Pousa, Fidel I. Schaposnik, Melanie Sclar y Ariel Zylber.

Problema A — AM/FM

AUTOR: PABLO ARIEL HEIBER - UNIVERSIDAD DE BUENOS AIRES

Amelia ha decidido retirarse de las competencias de programación y mudarse a un lugar más tranquilo, alejado del bullicio de la ciudad. Su sueño es poder sentarse frente a su casa a ver el atardecer sobre el campo, mientras escucha alguna de las radionovelas que tanto le gustan. Sin embargo, antes de poder cumplir su sueño debe resolver un último problema, que es elegir a dónde debe mudarse.

El campo al que quiere mudarse Amelia es muy grande y llano, tanto que puede representarse mediante un plano infinito sobre el que imaginamos un sistema cartesiano de coordenadas (X, Y) . En este campo hay N estaciones de radio numeradas desde el 1 hasta el N . La i -ésima estación emite su señal desde una antena ubicada en el punto (X_i, Y_i) , teniendo la señal un rango de alcance R_i . Dicha estación puede ser sintonizada desde cualquier punto (X, Y) cuya distancia a la antena emisora sea menor o igual al rango de alcance correspondiente, es decir que satisfaga

$$\sqrt{(X - X_i)^2 + (Y - Y_i)^2} \leq R_i.$$

Las señales de las distintas estaciones de radio pueden superponerse, pero no interfieren entre sí. Para poder escuchar la mayor cantidad de radionovelas distintas, Amelia desea ubicar su casa en un punto que esté dentro del rango de alcance de la mayor cantidad posible de estaciones de radio. Ahora Amelia quiere saber, dada la descripción del campo, cuál es el máximo número de estaciones de radio que va a poder sintonizar mientras ve el atardecer sobre el campo sentada al frente de su casa.

Entrada

La primera línea contiene un entero N , que representa la cantidad de estaciones de radio que hay en el campo ($1 \leq N \leq 100$). Cada una de las siguientes N líneas contiene tres enteros X_i , Y_i y R_i , que representan respectivamente las coordenadas de la antena emisora y el rango de alcance de la i -ésima estación ($-1000 \leq X_i, Y_i \leq 1000$ y $1 \leq R_i \leq 1000$ para $i = 1, 2, \dots, N$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la máxima cantidad de estaciones de radio que Amelia puede sintonizar si elige adecuadamente el punto a donde mudarse.

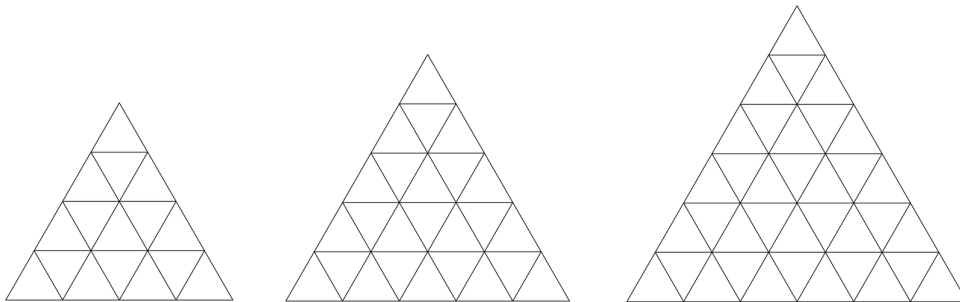
Entrada de ejemplo	Salida para la entrada de ejemplo
5 -1 0 2 1 0 2 0 -2 1 0 0 1 0 2 1	4

Página intencionalmente dejada (casi) en blanco.

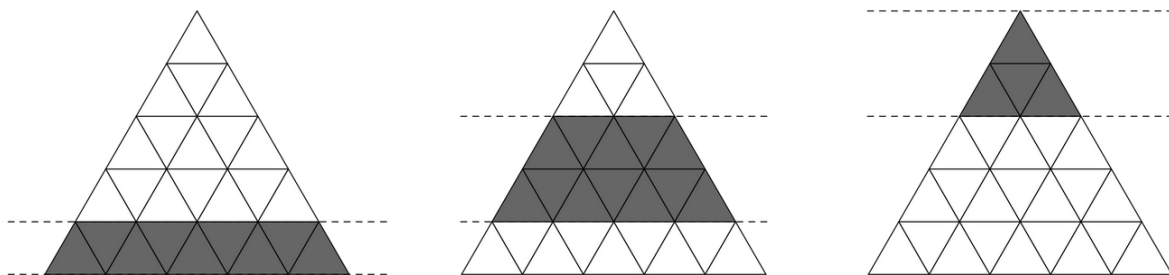
Problema B — Buen kilo de pan Flauta

AUTOR: MELANIE SCLAR - UNIVERSIDAD DE BUENOS AIRES

La empresa de pan *Flauta* quiere sacar a la venta un nuevo envase que contiene un kilo de pan (curiosamente, no se trata de pan flauta sino de pan lactal). Como parte del equipo de diseño de envases, les encargaron diseñar una pirámide nutricional acorde a los requerimientos de la empresa. La pirámide nutricional será representada por una grilla triangular de lado L , que es un triángulo equilátero dividido en triángulos unitarios (o sea, triángulos equiláteros de lado uno) trazando paralelas a los lados del triángulo mayor. Por ejemplo, en la siguiente figura se ven grillas para $L = 4, 5$ y 6 .

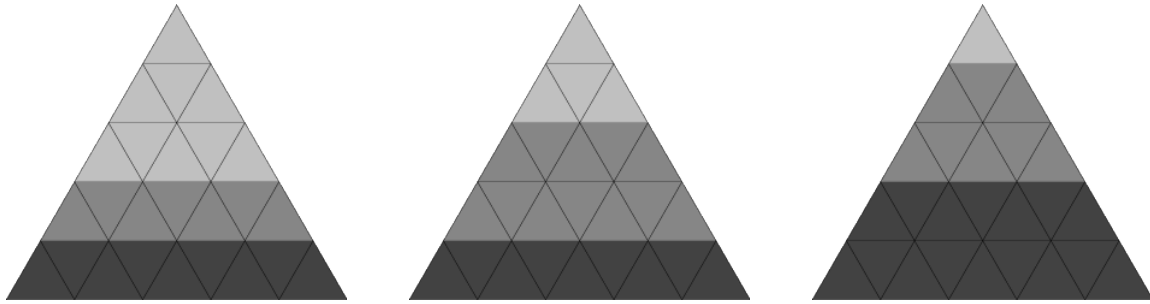


Se sabe que hay K tipos de alimentos distintos para mostrar en la pirámide nutricional, cada uno de los cuales será representado por una sección horizontal de la misma. Una sección horizontal es la porción de la pirámide que queda entre dos rectas paralelas a la base de la grilla, trazadas de manera tal que sólo toquen triángulos unitarios en sus bordes, sin atravesarlos. En el gráfico a continuación se pueden ver sombreadas tres secciones horizontales válidas de una grilla con $L = 5$.



Es importante que las K secciones en que quede dividida la pirámide se puedan ver bien, y para ello es necesario que cada una esté conformada por muchos triángulos unitarios. Sin embargo, como el tamaño de la grilla está fijo, hay una cantidad limitada de triángulos unitarios para repartir entre las K secciones. Para asegurarse de que ninguna sección quede desproporcionadamente pequeña, les han pedido que averigüen cuál es la máxima cantidad de triángulos unitarios que pueden conformar la sección más pequeña, si se eligen las secciones óptimamente.

Por ejemplo, a continuación se muestran tres formas distintas de dividir una grilla de lado $L = 5$ en $K = 3$ secciones.



En la grilla de la izquierda hay dos secciones con 9 triángulos unitarios cada una, teniendo la otra sección 7 triángulos unitarios. En la grilla central, las secciones tienen 12, 9 y 4 triángulos unitarios, mientras que en la grilla de la derecha tienen 16, 8 y 1. De estas tres formas de elegir las secciones horizontales, la mejor es entonces la de la grilla de la izquierda, dado que en ella la sección más pequeña tiene 7 triángulos unitarios, mientras que en la grilla central y de la derecha las secciones más pequeñas tienen 4 y 1 triángulos unitarios, respectivamente. De hecho, la elección de las secciones representada en la grilla a la izquierda de la figura es óptima, ya que no hay ninguna otra elección posible de $K = 3$ secciones horizontales en la que la más pequeña de ellas tenga más de 7 triángulos unitarios.

Entrada

Una línea conteniendo dos enteros L y K , representando respectivamente la longitud del lado de la grilla triangular y la cantidad de secciones en las que se desea dividir dicha grilla ($1 \leq K \leq L \leq 10^6$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la máxima cantidad de triángulos unitarios que pueden conformar la sección horizontal más pequeña.

Entrada de ejemplo	Salida para la entrada de ejemplo
5 3	7

Entrada de ejemplo	Salida para la entrada de ejemplo
5 2	9

Entrada de ejemplo	Salida para la entrada de ejemplo
1000000 1	1000000000000

Entrada de ejemplo	Salida para la entrada de ejemplo
1000000 30000	32679639

Problema C – CompuTenis recargado

AUTOR: JUAN CRUZ PIÑERO - UNIVERSIDAD NACIONAL DEL COMAHUE

Posiblemente recuerden de un TAP anterior al CompuTenis, aquel deporte especialmente adaptado a un público sin estado físico mensurable y cuyas reglas involucran programar con el codo pegado a la oreja. En esta ocasión volvemos a tener un problema sobre esta apasionante disciplina, por si la solución de aquel problema les resultó demasiado fácil.

Nuevamente, para los efectos de este problema lo único que hace falta saber sobre el CompuTenis es que en un partido compiten dos jugadores, a los que denominaremos A y B. Gana el partido aquel jugador que gane primero S sets, cada uno de los cuales está compuesto por uno o mas juegos. En cada set se disputan tantos juegos como hagan falta para que alguno de los dos jugadores venza en al menos J juegos, con una diferencia de al menos D juegos ganados más que su oponente. Aquel jugador que cumpla con ambas condiciones es entonces el ganador del set correspondiente.

La Asociación de Clubes Modernos (ACM) encontró recientemente un registro de partidos de CompuTenis prehistóricos. Cada registro consiste en una cadena compuesta por N caracteres 'A' o 'B', indicando el jugador que ganó cada uno de los N juegos que tuvo el partido, en el orden en el que fueron sucediendo. Ahora la ACM quiere saber, para cada registro, cuál fue el resultado del partido.

Entrada

La primera línea contiene cuatro enteros N , S , J y D . El valor N representa la cantidad de juegos presentes en el registro a analizar ($1 \leq N \leq 10^5$). El valor S indica la cantidad de sets que es necesario ganar para ganar un partido ($1 \leq S \leq 10$). El valor J es la cantidad mínima de juegos que es necesario ganar para ganar un set, mientras que el valor D indica que un jugador debe ganar al menos esa cantidad de juegos más que su oponente para ganar el set ($1 \leq D \leq J \leq 100$). La segunda línea contiene una cadena compuesta por N caracteres 'A' o 'B'. El i -ésimo caracter de la cadena indica qué jugador ganó el i -ésimo juego disputado en el partido. La cadena de la entrada representa un registro válido de un partido completo.

Salida

Imprimir en la salida una línea conteniendo dos enteros que representan la cantidad de sets ganados por el jugador A y por el jugador B, respectivamente.

Entrada de ejemplo	Salida para la entrada de ejemplo
10 5 2 1 AAAAAAAAAA	5 0

Entrada de ejemplo	Salida para la entrada de ejemplo
21 3 3 2 AABABBBABBBABABABBABB	1 3

Página intencionalmente dejada (casi) en blanco.

Problema D — Directo a la felicidad

AUTOR: FIDEL I. SCHAPOSNIK - UNIVERSIDAD NACIONAL DE LA PLATA

El estado separatista de Nlogonia tiene N ciudades numeradas del 1 al N . Algunos pares de ciudades están conectados por rutas, de modo tal que si hay una ruta entre la ciudad A y la ciudad B es posible trasladarse desde A a B , o viceversa desde B a A . Decimos que un habitante de la ciudad i puede visitar a otro de la ciudad j si existe una secuencia de ciudades distintas c_1, c_2, \dots, c_m con $m \geq 2$ tal que $c_1 = i$, $c_m = j$ y hay una ruta entre las ciudades c_k y c_{k+1} para $k = 1, 2, \dots, m - 1$.

Los habitantes de Nlogonia son muy sociables, por lo que tienen amigos en todas las ciudades que pueden visitar. Sin embargo, también son algo perezosos, de modo que no son felices salvo que puedan visitar a cada uno de sus amigos tomando exactamente una ruta directamente desde su ciudad hasta la del amigo en cuestión.

Para evitar la escisión de Nlogonia, la reina ha decidido llevar a cabo un conjunto de obras para lograr que todos sus habitantes sean felices. Por un lado, puede ordenar la construcción de nuevas rutas entre las ciudades de Nlogonia, incurriendo en un costo R por cada nueva ruta que se construya. Por otro lado, como construir muchas rutas puede llegar a ser muy caro, también pueden construirse estadios en algunas de las ciudades de Nlogonia. La construcción de un estadio tiene un costo E , y hace inmediatamente felices a todos los habitantes de la ciudad en la que se lleva a cabo la construcción. Sin embargo, hay que tener en cuenta que los habitantes de Nlogonia son además algo celosos, de modo que no serán felices si no hay un estadio en su ciudad pero sí en la de alguno de sus amigos. ¿Pueden ayudar a la reina a calcular el mínimo costo de las obras necesarias para lograr que todos los habitantes de Nlogonia sean felices?

Entrada

La primera línea contiene cuatro enteros N , M , R y E . El entero N representa la cantidad de ciudades en Nlogonia ($2 \leq N \leq 1000$), M representa la cantidad de rutas que hay originalmente ($1 \leq M \leq 10^5$), mientras que R y E representan el costo de construcción de una ruta y un estadio, respectivamente ($1 \leq R, E \leq 1000$). Cada una de las siguientes M líneas describe una ruta distinta mediante dos enteros A y B , que representan las ciudades que son conectadas por dicha ruta ($1 \leq A, B \leq N$ con $A \neq B$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa el mínimo costo de las obras necesarias para lograr que todos los habitantes de Nlogonia sean felices.

Entrada de ejemplo	Salida para la entrada de ejemplo
9 6 11 12	71
1 2	
3 2	
4 5	
5 6	
6 7	
9 7	

Página intencionalmente dejada (casi) en blanco.

Problema E — Empaquetamiento perfecto

AUTOR: FIDEL I. SCHAPOSNIK - UNIVERSIDAD NACIONAL DE LA PLATA

Una de las etapas más difíciles de automatizar en una línea de producción industrial de galletitas es la del empaquetado. El objetivo es diseñar un aparato capaz de contar un número preciso G de galletitas para colocar en un paquete, y la dificultad reside en que estas pueden tener formas a veces muy distintas, por ejemplo de diversos animales.

En la fábrica de galletitas en la que trabajan el equipo de diseño no ha podido sobreponerse a estas dificultades. Lo mejor que ha conseguido lograr es una máquina de conteo que al ser activada puede seleccionar C_i galletitas con probabilidad P_i , para $i = 1, 2, \dots, K$. Habiendo gastado ya todo el presupuesto destinado a esta tarea, tendrán que buscar la manera de hacer algo útil con esto.

Afortunadamente, se les ocurrió la siguiente idea para llevar a buen puerto el proyecto. Para construir el empaquetador automático utilizarán N máquinas de conteo como la arriba descrita, junto con un programa regulador especial que llevará a cabo el siguiente proceso de manera iterada. Antes de comenzar la primera iteración se activan las N máquinas de conteo, seleccionando cada una cierta cantidad de galletitas. En cada iteración, el programa elige de las N máquinas el subconjunto no vacío S para el que la suma de las galletitas seleccionadas sea lo más cercana posible al valor deseado G (en el sentido de que el valor absoluto de la diferencia entre dicha cantidad y G sea mínimo). De haber más de un subconjunto que cumpla con esta condición, el programa elige como S a cualquiera de ellos con probabilidad uniforme. A continuación, las galletitas seleccionadas por las máquinas de S se retiran de las mismas para ser empaquetadas en un solo paquete. Finalmente, cada una de las máquinas de S se activa nuevamente para seleccionar cierta cantidad de galletitas, mientras que las máquinas que no pertenecen a S permanecen inalteradas (es decir, con la cantidad de galletitas seleccionadas en la iteración anterior). Este proceso se repite hasta obtener la cantidad deseada M de paquetes.

Por ejemplo, supongamos que hay $N = 3$ máquinas a las que llamamos m_1 , m_2 y m_3 , pudiendo cada una de ellas seleccionar $C_1 = 1$ ó $C_2 = 2$ galletitas con probabilidad $P_1 = P_2 = \frac{1}{2}$. Si se desea producir $M = 2$ paquetes con $G = 5$ galletitas por paquete, un posible desarrollo del proceso sería el siguiente. Antes de comenzar, las tres máquinas seleccionan $C_2 = 2$ galletitas cada una (esto ocurrirá con probabilidad $\frac{1}{2} \times \frac{1}{2} \times \frac{1}{2} = \frac{1}{8}$). En la primera iteración, el programa puede entonces elegir los subconjuntos $\{m_1, m_2, m_3\}$, $\{m_1, m_2\}$, $\{m_1, m_3\}$ y $\{m_2, m_3\}$, cada uno con probabilidad $\frac{1}{4}$ ya que todos ellos tienen una diferencia de una galletita con la cantidad deseada. Si suponemos que se elige el subconjunto $S = \{m_2, m_3\}$, a continuación se empaquetan las cuatro galletitas seleccionadas por las máquinas m_2 y m_3 . Estas máquinas volverán a activarse, por ejemplo obteniendo que m_2 selecciona $C_1 = 1$ galletita y m_3 selecciona $C_2 = 2$ galletitas (lo cual ocurrirá con probabilidad $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$). En este punto termina la primera iteración, con las galletitas seleccionadas por las tres máquinas m_1 , m_2 y m_3 siendo 2, 1 y 2, respectivamente. En la segunda iteración el programa debe seleccionar necesariamente el subconjunto $S = \{m_1, m_2, m_3\}$, pues éste contiene exactamente cinco galletitas, las cuales pasarán a formar parte de un nuevo paquete. Finalmente, se activan una vez más las tres máquinas y termina el proceso habiéndose producido los $M = 2$ paquetes deseados. Nótese que la probabilidad neta de que ocurra el proceso aquí descrito es $\frac{1}{8} \times \frac{1}{4} \times \frac{1}{4} = \frac{1}{128}$, y que la cantidad media de galletitas por paquete resultó ser en este caso de 4.5 galletitas (porque se produjeron dos paquetes, uno de cuatro y otro de cinco galletitas).

Su jefe no está completamente convencido de que este sistema vaya a funcionar, de modo que les exige una prueba concreta de la idea. Para convencerlo, bastará que calculen la cantidad esperada de galletitas que hay por paquete luego de producir M paquetes consecutivamente, si suponen que las N máquinas del aparato seleccionan siempre cantidades de galletitas según las probabilidades dadas.

Entrada

La primera línea contiene cuatro enteros N , K , G y M . El valor N representa la cantidad de máquinas de conteo a utilizar ($1 \leq N \leq 4$), K representa el número de posibles cantidades de galletitas que dichas máquinas pueden seleccionar ($1 \leq K \leq 6$), G representa la cantidad deseada de galletitas por paquete ($1 \leq G \leq 100$), y M representa la cantidad total de paquetes a producir ($1 \leq M \leq 10^7$). Cada una de las siguientes K líneas contiene un entero C_i y un racional P_i , indicando que la máquina de conteo seleccionará C_i galletitas con probabilidad P_i ($1 \leq C_i \leq 100$ y $0 < P_i \leq 1$ para $i = 1, 2, \dots, K$, siendo dado P_i con exactamente dos dígitos después del punto decimal). Nótese que $C_i \neq C_j$ para $i \neq j$, y que además $\sum_{i=1}^K P_i = 1$.

Salida

Imprimir en la salida una línea conteniendo un racional que representa la cantidad esperada de galletitas por paquete habiendo producido M paquetes como se describe en el enunciado. Imprimir el resultado con exactamente 6 dígitos luego del punto decimal, redondeando de ser necesario.

Entrada de ejemplo	Salida para la entrada de ejemplo
3 2 5 1 1 0.50 2 0.50	4.312500

Entrada de ejemplo	Salida para la entrada de ejemplo
3 2 5 2 1 0.50 2 0.50	4.327148

Entrada de ejemplo	Salida para la entrada de ejemplo
1 2 1 1 1 0.40 2 0.60	1.600000

Problema F – Favoritismo inducido

AUTOR: MELANIE SCLAR - UNIVERSIDAD DE BUENOS AIRES

En un reino muy muy lejano hay N ciudades numeradas del 1 al N , estando algunos pares de ciudades conectados mediante rutas. Cuando dos ciudades están conectadas directamente por una ruta, se dice que dichas ciudades son vecinas. Como resultado del cuidadoso planeamiento de sus monarcas, el sistema vial del reino tiene características muy especiales. Se sabe que no hay dos rutas que conecten al mismo par de ciudades, y que todas las rutas conectan dos ciudades distintas. Además, una característica muy particular de las rutas es que hay exactamente un camino entre dos ciudades cualesquiera, constituido por una secuencia de rutas que conectan ciudades vecinas desde la ciudad inicial hasta la ciudad final.

En el reino muy muy lejano en cuestión, el rey quiere imponer el *curling* como deporte nacional. Para ello, quiere que cada ciudad tenga un equipo favorito de curling, pues si toda la ciudad apoya a un mismo equipo se incrementa el sentimiento de pertenencia de sus ciudadanos. Hay algunas ciudades que ya tienen un equipo favorito, y como el curling es pasión de multitudes no se puede cambiar dicha elección. Otras ciudades aún no tienen uno predilecto, de modo que deberá asignárseles un equipo para que pase a ser su favorito.

El presidente del Instituto de Curling Público Competitivo (ICPC) los llamó para pedirles ayuda, porque el rey le encargó que asigne los equipos favoritos para las ciudades que aún no tienen uno. El problema es que hay demasiadas ciudades en el reino, de modo que el ICPC no sabe cómo completar la tarea. Sabemos que en todo el reino hay E equipos que han sido numerados del 1 al E , y no hay presupuesto para crear más. El ICPC les facilitó un *índice de disturbios* entre equipos para todo par de equipos, es decir un entero D_{ij} que representa el grado de enemistad entre los seguidores de los equipos i y j , para $i, j = 1, 2, \dots, E$. Nótese que incluso existe el índice de disturbios para un equipo consigo mismo, pues podría ocurrir que los habitantes de dos ciudades vecinas con el mismo equipo favorito puedan comenzar a pelearse para ver cuál es la mejor hinchada.

El ICPC les pidió que asignen equipos favoritos para todas las ciudades del reino que aún no tienen uno, de manera tal que se minimicen los disturbios entre ciudades vecinas. Para ello, deben minimizar el *índice nacional de disturbios*, que se calcula sumando todos los índices de disturbios de equipos asignados a ciudades vecinas. ¿Pueden ayudar al ICPC a determinar cuál es el mínimo valor posible del índice nacional de disturbios?

Entrada

La primera línea contiene dos enteros N y E , que representan respectivamente la cantidad de ciudades y la cantidad de equipos de curling que hay en el reino ($2 \leq N \leq 5 \times 10^4$ y $1 \leq E \leq 50$). Las siguientes E líneas describen los índices de rivalidad entre los equipos de curling. Cada una de estas líneas contiene E enteros, siendo el j -ésimo entero de la i -ésima de estas líneas D_{ij} , el índice de disturbios entre los equipos i y j ($0 \leq D_{ij} \leq 1000$ con $D_{ij} = D_{ji}$ para $i, j = 1, 2, \dots, E$).

A continuación siguen E líneas que describen los equipos favoritos de las ciudades que ya tienen uno elegido. La i -ésima de estas líneas comienza con un entero no negativo K_i seguido por una lista de K_i ciudades cuyo equipo favorito es el número i ($0 \leq K_i \leq N$

para $i = 1, 2, \dots, E$). Ninguna ciudad tiene más de un equipo favorito, y no hay elementos repetidos en las listas.

Las últimas $N - 1$ líneas describen las rutas entre las ciudades del reino. Cada una de ellas contiene dos enteros A y B , indicando que existe una ruta entre la ciudad A y la ciudad B ($1 \leq A, B \leq N$ con $A \neq B$). Las rutas son bidireccionales y no hay rutas repetidas en la entrada. Se garantiza además que existe un único camino entre cada par de ciudades, posiblemente pasando por otras ciudades intermedias.

Salida

Imprimir en la salida una línea conteniendo un entero que representa el mínimo valor del índice nacional de disturbios que es posible alcanzar si se asignan los equipos favoritos de manera óptima.

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

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

Problema G — Genética alienígena II

AUTOR: FIDEL I. SCHAPOSNIK - UNIVERSIDAD NACIONAL DE LA PLATA

Hay una característica fundamental e inalterable que comparte toda la vida en la Tierra, desde el más minúsculo microbio hasta las ballenas, los dinosaurios y los seres humanos: el ADN. Este es el único mecanismo conocido para la transmisión y replicación de información genética, lo cual plantea una de las preguntas más importantes que la biología moderna no ha podido responder hasta ahora: ¿es el ADN la única forma de codificar esta información, o hay otros mecanismos posibles?

El profesor Gould es un exobiólogo teórico, y se dedica a estudiar la posibilidad de que exista vida extraterrestre cuya información genética no se encuentre codificada en la forma de ADN. Actualmente, está desarrollando un modelo basado en la codificación de información genética en la forma de una cadena de pseudo-nucleótidos, que vamos a representar mediante los caracteres ‘b’, ‘d’, ‘o’, ‘p’, ‘q’, ‘v’, ‘w’ y ‘x’. Cada pseudo-nucleótido tiene un conjugado: ‘o’, ‘v’, ‘w’ y ‘x’ son cada uno su propio conjugado, mientras que ‘b’ es el conjugado de ‘d’, ‘p’ es el conjugado de ‘q’, y viceversa, ‘d’ es el conjugado de ‘b’ y ‘q’ es el conjugado de ‘p’.

En el modelo del profesor Gould, un organismo puede sufrir una mutación a partir de cierta posición de su cadena de pseudo-nucleótidos, que resulta en la inversión y conjugación de la segunda parte de dicha cadena. Más específicamente, si la cadena de pseudo-nucleótidos original es $a_1 a_2 \cdots a_N$, y la mutación ocurre a partir de la posición i , la cadena de pseudo-nucleótidos resultante es $a_1 a_2 \cdots a_{i-1} \bar{a}_N \bar{a}_{N-1} \cdots \bar{a}_{i+1} \bar{a}_i$, donde \bar{a}_k representa el conjugado del pseudo-nucleótido originalmente en la posición k .

A lo largo de su evolución, un organismo dado puede sufrir varias mutaciones de este tipo, siendo la única restricción que las sucesivas mutaciones deben ocurrir a partir de posiciones cada vez más cercanas al final de la cadena de pseudo-nucleótidos. Por ejemplo, la cadena “bdopqvw” puede sufrir una mutación a partir de la posición 3 resultando en la cadena “bdxwvpqo”, y luego otra mutación a partir de la posición 7 para dar la cadena “bdxwvpop”, pero estas dos mutaciones no podrían haber ocurrido en el orden inverso.

En este punto de su investigación, el profesor Gould tiene dos cadenas de pseudo-nucleótidos que son particularmente interesantes, y querría saber la mínima cantidad de mutaciones que debe sufrir la primera de ellas para transformarse en la otra. ¿Pueden ayudarlo?

Entrada

La primera línea contiene un entero N , que representa la longitud de las dos cadenas de pseudo-nucleótidos a ser analizadas ($1 \leq N \leq 1000$). Cada una de las siguientes dos líneas contiene una cadena de N caracteres ‘b’, ‘d’, ‘o’, ‘p’, ‘q’, ‘v’, ‘w’ y ‘x’, representando una cadena de pseudo-nucleótidos.

Salida

Imprimir en la salida una línea conteniendo un entero que representa la mínima cantidad de mutaciones que debe sufrir la primera cadena de la entrada para transformarse en la segunda. De no ser posible esto, imprimir el número -1 .

Entrada de ejemplo	Salida para la entrada de ejemplo
8 bdopqvw bdxwvpop	2

Entrada de ejemplo	Salida para la entrada de ejemplo
5 bdopq bdopq	0

Entrada de ejemplo	Salida para la entrada de ejemplo
10 d b b d d b b d d b b d d b b d d b b	1

Entrada de ejemplo	Salida para la entrada de ejemplo
13 o p o q p v b d x w w b p v w p o p v x x b d p q q	-1

Problema H — Haciendo la tarea

AUTOR: FIDEL I. SCHAPOSNIK - UNIVERSIDAD NACIONAL DE LA PLATA

Hugo va a la escuela primaria, pero está convencido de que allí no le dan suficiente tarea para el hogar. En particular, recientemente le enseñaron a restar, pero cada día le dan solamente una resta para hacer en su casa. Hugo sabe que para aprender una técnica tan compleja debe practicar mucho más, de modo que ha decidido tomar cartas en el asunto y crear su propia tarea.

Como no es fácil para Hugo inventar ejercicios sobre un tema que no domina, se le ocurrió el siguiente método para poder hacer muchas restas. Empieza pidiéndole a su madre que le diga un número N , luego arma el número M que tiene los mismos dígitos que N pero ordenados de menor a mayor de izquierda a derecha, y finalmente realiza la resta $N - M$. Así, si por ejemplo su madre elige el número $N = 321$, entonces $M = 123$ y la resta que debe hacer Hugo es $N - M = 321 - 123 = 198$.

Como Hugo no quiere molestar a su madre muy seguido, va a repetir este procedimiento usando el resultado de la resta $N - M$ de un paso como el número N para comenzar el paso siguiente. Esto terminará solamente cuando Hugo llegue en algún momento al valor $N = 0$, dado que este caso no sirve para practicar restas porque tendría $M = 0$ y él ya sabe que si no tiene ningún caramelo no puede comer ningún caramelo, por lo que seguirá no teniendo ni comiendo caramelos para siempre.

Ahora la madre de Hugo quiere saber, dado un número N , cuántas restas va a hacer Hugo si ella le da ese número para comenzar su tarea. En el ejemplo anterior, en el segundo paso Hugo tendrá $N = 198$, de modo que $M = 189$ y $N - M = 198 - 189 = 9$. Entonces, en el tercer paso $N = 9$, $M = 9$ y $N - M = 9 - 9 = 0$, por lo que aquí se termina la diversión, siendo que en el cuarto paso tendría $N = 0$. Por lo tanto, empezando con el número $N = 321$ Hugo debe hacer 3 restas.

Entrada

Una línea conteniendo un entero N , que representa el número que su madre le da a Hugo para comenzar su tarea ($1 \leq N \leq 10^9$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la cantidad de restas que Hugo debe hacer si empieza su tarea con el número N .

Entrada de ejemplo	Salida para la entrada de ejemplo
321	3

Entrada de ejemplo	Salida para la entrada de ejemplo
20	2

Entrada de ejemplo	Salida para la entrada de ejemplo
960687301	91

Página intencionalmente dejada (casi) en blanco.

Problema I – Invasión extraterrestre

AUTOR: FIDEL I. SCHAPOSNIK - UNIVERSIDAD NACIONAL DE LA PLATA

En la novela “*A for Andromeda*” de Fred Hoyle y John Elliot, una civilización que habita un planeta en órbita alrededor de una estrella de la constelación de Andrómeda, a 200 años luz del sistema solar, decide colonizar la galaxia. Para evitar los largos y costosos viajes interestelares, esta civilización decide realizar la colonización a distancia, enviando una señal en lugar de naves (dicha señal contiene instrucciones para armar una supercomputadora con una inteligencia artificial capaz de apoderarse del mundo de las desdichadas civilizaciones que la construyan, pero ese no es nuestro problema por el momento).

Uno de los inconvenientes que los humanos deben sortear para construir la supercomputadora es la decodificación de la señal. Ocurre que los alienígenas emiten dos mensajes en dos frecuencias distintas, repitiendo en cada una de ellas un código de N caracteres en un bucle infinito. Por ejemplo, si $N = 3$ y uno de los códigos es “abc”, el mensaje alienígena en la frecuencia correspondiente será “. . .cabcabcab. . .”, donde los puntos suspensivos indican que el código se repite infinitamente tanto hacia atrás como hacia adelante. Por esta razón, las estaciones terrestres que reciben la señal no son capaces de determinar cuál es realmente el código emitido, ya que puede haber más de un código compatible con un dado mensaje. En el ejemplo anterior, sabiendo que $N = 3$ podrían interpretar que el código es cualquiera de las tres posibilidades “abc”, “bca” y “cab”.

Para complicar aún más las cosas, si bien los dos mensajes recibidos están compuestos solamente por caracteres del alfabeto de la ‘a’ a la ‘z’, como son transmitidos en frecuencias distintas existe una ambigüedad en la identificación de los caracteres entre uno y otro. De este modo, si llamamos a los caracteres $c_1 = \text{‘a’}$, $c_2 = \text{‘b’}$, y así siguiendo hasta $c_{26} = \text{‘z’}$, es posible que todas las apariciones del carácter c_i en uno de los mensajes sean reemplazadas por el carácter $c_{\sigma(i)}$ en el otro, siendo $\sigma(i)$ una permutación arbitraria de los números del 1 al 26. Así, si tenemos por ejemplo $\sigma(1) = 24$, $\sigma(2) = 25$ y $\sigma(3) = 26$, el código “abc” en una frecuencia se transformaría en la otra en “xyz”, de manera que el mensaje correspondiente será “. . .zxyzxyzxy. . .”.

Como encargados de la decodificación de la señal alienígena, su tarea es determinar la longitud de la máxima subcadena común que pueden tener dos códigos compatibles con los mensajes recibidos. Esto es, deben determinar el máximo valor de K tal que uno de los mensajes es compatible con el código $a_1a_2 \dots a_N$ y el otro es compatible con el código $b_1b_2 \dots b_N$, existiendo además i y j con $0 \leq i, j \leq N - K$ tales que, a menos de una permutación del alfabeto, $a_{i+k} = b_{j+k}$ para $k = 1, \dots, K$.

Entrada

La primera línea contiene un entero N que representa la longitud de los códigos emitidos por los alienígenas ($1 \leq N \leq 1000$). Cada una de las dos líneas siguientes contiene la descripción del mensaje recibido en una frecuencia distinta, en la forma de una cadena de N caracteres de la ‘a’ a la ‘z’. El mensaje recibido se obtiene repitiendo la cadena correspondiente infinitamente.

Salida

Imprimir en la salida una línea conteniendo un entero que representa la longitud de la máxima subcadena común que pueden tener dos códigos que sean compatibles con los mensajes recibidos.

Entrada de ejemplo	Salida para la entrada de ejemplo
3 abc xyz	3

Entrada de ejemplo	Salida para la entrada de ejemplo
3 aab cdd	3

Entrada de ejemplo	Salida para la entrada de ejemplo
4 abab xyzw	2

Entrada de ejemplo	Salida para la entrada de ejemplo
4 xyzw abab	2

Entrada de ejemplo	Salida para la entrada de ejemplo
18 imzbyqlgjwrvfspthe rubihyvjnomqdznhath	16

Problema J — Jugando con Piedras II

AUTOR: PABLO BLANC - UNIVERSIDAD DE BUENOS AIRES

Jaimito no ha perdido el tiempo desde que le regalaron las piedras en el 2013. Ya no se dedica a jugar con su madre, Jimena, a juegos en los que tiene asegurada la victoria, sino que en estos dos años ha explorado muchísimas variantes de juegos con piedras, como cualquier niño debería. Se ha vuelto muy bueno en algunos de ellos e incluso participó de diversos torneos obteniendo excelentes resultados.

Desde hace unos meses está interesado particularmente por un juego con piedras que se llama *TArros con Piedras* (TAP). En este juego hay N tarros con piedras adentro y participan dos jugadores que juegan alternadamente. En su turno un jugador debe sacar una piedra de un tarro, dos de otro, tres de otro y así siguiendo hasta sacar N piedras de algún tarro. De esta manera, cada jugador saca en su turno piedras de todos los tarros. El juego continúa hasta que uno de los jugadores no puede sacar las piedras de una manera válida en su turno. Dicho jugador pierde la partida, siendo el otro jugador el ganador.

Por ejemplo, una partida con $N = 3$ tarros puede comenzar teniendo uno de ellos $P_1 = 3$ piedras, otro $P_2 = 4$ piedras y el tercero $P_3 = 10$ piedras. En tal partida el jugador que comienza dispone de una estrategia ganadora, ya que puede retirar una piedra del tarro que originalmente tenía diez, dos piedras del que tenía tres y, finalmente, tres piedras del que tenía cuatro. Deja de este modo a los tarros con $P_1 = 1$, $P_2 = 1$ y $P_3 = 9$ piedras, por lo que el segundo jugador no puede sacar piedras de una manera válida.

Jaimito está participando de un torneo de TAP y ha llegado a la final, en la que va a enfrentar a Jacinta. Ambos son jugadores expertos y no cometen errores al jugar. Jaimito le contó a su madre cómo va a ser la partida, es decir la cantidad N de tarros que va a haber y cuántas piedras tendrá cada uno. Jimena sabe que Jacinta comenzará la partida y quiere saber si su hijo ganará el torneo, pero no logra determinarlo porque los nervios no le permiten pensar correctamente, *está razonando fuera del recipiente*. ¿Pueden ayudarla?

Entrada

La primera línea contiene un entero N que representa la cantidad de tarros en el juego ($1 \leq N \leq 10^5$). La segunda línea contiene N enteros P_i que representan la cantidad de piedras en cada tarro antes de comenzar la partida ($1 \leq P_i \leq 10^9$ para $i = 1, 2, \dots, N$).

Salida

Imprimir en la salida una línea conteniendo un caracter que indica si Jaimito va a ganar la final o no. El caracter debe ser una 'S' si Jaimito va a ganar, y una 'N' caso contrario.

Entrada de ejemplo	Salida para la entrada de ejemplo
3 3 4 10	N

Entrada de ejemplo	Salida para la entrada de ejemplo
2 10 3	S

Página intencionalmente dejada (casi) en blanco.

Problema K — Kimetto, Kipsang y Kipchoge

AUTOR: FIDEL I. SCHAPOSNIK - UNIVERSIDAD NACIONAL DE LA PLATA

Kenya es la cuna de algunos de los mejores corredores de larga distancia de todos los tiempos. Sin ir más lejos, ocho de las últimas diez mejores marcas en los tradicionales 41.195 km de la maratón corresponden a corredores de dicho país. Dennis Kimetto, Wilson Kipsang y Eliud Kipchoge son tres de estos corredores, y se proponen volver a batir el récord mundial de su disciplina mañana 27 de septiembre cuando compitan en la 42^{da} edición de la Maratón de Berlín.

Kimetto, Kipsang y Kipchoge son buenos amigos, y les gusta entrenar juntos corriendo a la vera del río Tana para poder apreciar los hermosos árboles que allí crecen. Hay N árboles junto al río, que vamos a considerar como numerados desde el 1 hasta el N . El i -ésimo árbol es de la especie S_i , y se encuentra a i metros de la desembocadura del río. A nuestros tres corredores los motiva especialmente ver muchos árboles de diferentes especies. Por esta razón, cada día de entrenamiento eligen un número de árbol K entre 1 y N , y luego corren desde el K -ésimo árbol en el sentido del río, es decir hacia los árboles $K - 1$, $K - 2$ y así siguiendo, sin parar hasta volver a ver un árbol de alguna especie que ya hayan visto ese día, o hasta llegar a la desembocadura del río, lo que ocurra primero.

Por ejemplo, si hay $N = 4$ árboles de las especies $S_1 = 1$, $S_2 = 2$, $S_3 = 1$ y $S_4 = 3$, cuando eligen $K = 4$ el entrenamiento consiste en correr 3 metros, desde el árbol número 4 hasta el árbol número 1 (donde se detienen porque éste es de la misma especie que el árbol número 3). En cambio, cuando eligen $K = 2$ deben correr dos metros hasta la desembocadura del río, donde se detienen a pesar de no haber visto dos árboles de la misma especie en su recorrido.

El entrenamiento de los corredores de larga distancia dura décadas, y en este tiempo es habitual que algunos árboles caigan durante una tormenta. Cuando esto ocurre, el árbol caído es inmediatamente reemplazado por otro, no necesariamente de la misma especie. Kimetto, Kipsang y Kipchoge llevan un diario en el que registran toda la información relevante para su entrenamiento. En particular, saben de qué especie es cada árbol, y qué número de árbol eligieron cada día para comenzar a correr. ¿Pueden ayudarlos a calcular cuánto corrieron durante cada día de entrenamiento?

Entrada

La primera línea contiene dos enteros N y R , que representan la cantidad de árboles a la vera del río y la cantidad de registros en el diario de entrenamiento ($1 \leq N, R \leq 5 \times 10^4$). La segunda línea contiene N enteros S_i especificando la especie del i -ésimo árbol al comenzar el entrenamiento ($1 \leq S_i \leq 10^6$ para $i = 1, 2, \dots, N$). Cada una de las siguientes R líneas contiene la descripción de un registro del diario, en orden cronológico. La descripción de un registro comienza con un caracter que puede ser una 'C' si corresponde a la caída de un árbol, o una 'E' si corresponde a un día de entrenamiento. Los registros de caída de árboles contienen dos enteros K y S luego de la 'C', indicando que cayó el K -ésimo árbol y fue reemplazado por otro de la especie S ($1 \leq K \leq N$ y $1 \leq S \leq 10^6$). Los registros de entrenamiento contienen un entero K luego de la 'E', que indica que los corredores comenzaron un día de entrenamiento en el K -ésimo árbol ($1 \leq K \leq N$). Siempre habrá al menos un registro de entrenamiento en la entrada.

Salida

Imprimir una línea por cada registro correspondiente a un día de entrenamiento, indicando la cantidad de metros que Kimetto, Kipsang y Kipchoge corrieron durante dicho día.

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

Entrada de ejemplo	Salida para la entrada de ejemplo
10 10	1
1 2 3 4 5 6 7 8 9 10	2
E 1	3
E 2	4
E 3	5
E 4	6
E 5	7
E 6	8
E 7	9
E 8	10
E 9	
E 10	

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