

Torneo Argentino de Programación

17 de septiembre de 2016

- 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, Ingeniería y Agrimensura
Universidad Nacional del Rosario
- Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
- Facultad de Ciencias Exactas y Tecnología
Universidad Nacional de Tucumán
- Facultad de Informática
Universidad Nacional de La Plata
- Facultad de Informática
Universidad Nacional del Comahue
- Facultad de Matemática, Astronomía, Física y Computación
Universidad Nacional de Córdoba
- Facultad Regional Resistencia
Universidad Tecnológica Nacional

Sesión de Competencia

Este conjunto contiene 12 problemas; las páginas están numeradas de 1 a 24.

Información General

Salvo indicación en contrario, lo siguiente vale para todos los problemas.

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, Ñ, é, È, 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, Ñ, é, È, 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.

Problema A – Ayudando al abuelo Laino

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

No sabemos concretamente por qué, pero al abuelo Laino no le agrada una de las vocales de la lengua castellana. Tal vez sea para él engorroso declamarla, con certeza el tartamudeo estaba mal contemplado en épocas pasadas. En todo caso, afortunadamente esa vocal no está presente en la palabra “abuelo”, de manera que la prole de sus vástagos no se encuentra en apuros cuando le llama afectuosamente de ese modo.

El abuelo Laino trabajó con nulo descanso durante décadas, por lo que se prepara para tomar un justo receso de sus arduas tareas. En este lapso, desea emprender una aventura atravesando parajes lejanos, para lo cual está ahora empacando su maleta. El abuelo Laino no desea llevar en ella objetos cuyos nombres contengan la vocal que tanto lo consterna, no vaya a ser que al verlos se vea forzado a pensar en la tan censurable letra durante su reposo. Su tarea es ayudarlo en esta labor, para lo cual deben aconsejarlo sobre cuáles de los objetos que posee puede empacar.

Entrada

Una línea conteniendo una cadena no vacía de hasta 20 caracteres de la ‘a’ a la ‘z’, indicando el nombre de un objeto que posee el abuelo Laino.

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si el abuelo Laino puede empacar el objeto cuyo nombre aparece en la entrada. El carácter debe ser una ‘S’ si el abuelo Laino puede empacarlo, y una ‘N’ caso contrario.

Entrada de ejemplo	Salida para la entrada de ejemplo
remera	S

Entrada de ejemplo	Salida para la entrada de ejemplo
camisa	N

Entrada de ejemplo	Salida para la entrada de ejemplo
buey	S

Entrada de ejemplo	Salida para la entrada de ejemplo
i	N

Entrada de ejemplo	Salida para la entrada de ejemplo
abuelo	S

Entrada de ejemplo	Salida para la entrada de ejemplo
estenoporquetienelai	N

Problema B — Buscando el camino

AUTOR: ARIEL ZYLBER - UNIVERSIDAD DE BUENOS AIRES

El festival de pastelería ha llegado a la ciudad. Esta es una gran oportunidad para que los pequeños emprendimientos del rubro recauden dinero y puedan darse a conocer. Cada emprendimiento que desee participar del festival lo puede hacer instalando un pequeño puesto en el salón que funciona como sede del mismo. Por cuestiones de seguridad, cada puesto debe ser instalado contra una de las paredes del salón. En su puesto, el emprendimiento exhibe los productos que tiene a la venta y dispone de lo necesario para poder venderlos al público que asista al festival. Una característica importante de estos puestos es que ofrecen una muestra gratis, es decir, una pequeña porción de alguno de sus productos es ofrecida de manera gratuita al que la desee. El objetivo es mostrar la gran calidad de sus recetas y así tentar al que degusta para que compre en el puesto.

Gracias a las muestras gratis, el festival atrae mucho público que aprovecha para comer deliciosos postres sin costo alguno, recogiendo distintas muestras a medida que recorre el salón. La mayoría de los asistentes compra a cambio algunos productos para ayudar a los emprendedores que realmente se destacan con sus platos. Uno de los concurrentes más famosos del festival es el Señor Barriga, que siempre recorre todos los puestos probando las muestras, y hasta otorga un premio a la mejor de ellas.

El Señor Barriga no quiere perder demasiado tiempo en el festival, por lo que le gustaría poder probar la comida de todos los puestos recorriendo la menor distancia posible. Para ello, posee el mapa que se encuentra en el folleto que publicita el festival, repartido con antelación por los organizadores. El mapa tiene dibujada la forma del salón, que este año es un polígono convexo. Además, tiene marcados N sitios importantes, dos de los cuales corresponden a la entrada y la salida del salón, siendo los $N - 2$ sitios restantes los puestos del festival. Cada sitio importante está representado como un punto sobre el borde del polígono que representa las paredes del salón.

El Señor Barriga les pide ahora ayuda para completar su misión. Les va a proporcionar las coordenadas en el plano cartesiano (X, Y) de los N sitios importantes del salón, ordenados en sentido antihorario (es decir en el orden en el que los visitaría si recorriera el salón manteniendo su mano derecha sobre la pared interior). Quiere saber cuál es la mínima distancia que debe recorrer para visitar todos los puestos, si empieza en la entrada del salón y termina en la salida, y elige óptimamente el orden en el que recorre los puestos.

Entrada

La primera línea contiene tres enteros N , E y S . El entero N representa la cantidad de sitios importantes en el mapa, que están numerados del 1 al N ($2 \leq N \leq 4000$). Los enteros E y S representan los sitios que corresponden a la entrada y la salida del salón, respectivamente ($1 \leq E, S \leq N$ con $E \neq S$). Cada una de las siguientes N líneas contiene dos enteros X e Y , representando los enteros de la i -ésima línea las coordenadas (X, Y) del i -ésimo sitio importante ($-10^4 \leq X, Y \leq 10^4$). Todos los sitios importantes están en puntos distintos, y se asegura que existe un polígono convexo que los contiene a todos en su borde.

Salida

Imprimir en la salida una línea conteniendo un racional que representa la mínima distancia que debe recorrer el Señor Barriga para recorrer todos los puestos del festival, comenzando en la entrada del salón y terminando en la salida. Imprimir el resultado con exactamente 6 dígitos luego del punto decimal, redondeando de ser necesario.

Entrada de ejemplo	Salida para la entrada de ejemplo
6 1 6 1 0 2 0 3 1 2 2 1 2 0 1	6.242641

Entrada de ejemplo	Salida para la entrada de ejemplo
6 1 4 0 0 10 0 20 0 20 1 10 1 0 1	23.000000

Problema C — Correlatividades

AUTOR: MARIANO CROSETTI - UNIVERSIDAD NACIONAL DE ROSARIO

Terminar una carrera universitaria no sólo es cuestión de estudiar y aprender. Para hacerse con el preciado título universitario, cada alumno tiene que demostrar lo que ha aprendido, y para ello debe aprobar N materias. A menudo es necesario además respetar innumerables y deliberadamente caprichosos laberintos reglamentarios.

En el *Instituto Con Pocas Correlatividades* (ICPC) rigen antiguas normas que impiden a los alumnos aprobar ciertas materias no habiendo aprobado antes algunas otras. Estas últimas son llamadas “correlativas” de las primeras. Cada materia puede tener cero o más materias correlativas, pero no existen correlatividades cíclicas, de modo que siempre es posible terminar una carrera.

Gabina es una alumna de Ciencias de la Contentura, y afortunadamente sus profesores son personas extremadamente comprensivas. Por ello, le permiten a Gabina aprobar materias sin tener sus correlativas aprobadas. El inconveniente es que el sistema informático del ICPC sólo puede registrar las materias como aprobadas respetando el régimen de correlatividades. De este modo, una materia estará registrada si y sólo si está aprobada y tiene todas sus materias correlativas registradas.

Ver su progreso mantiene a Gabina motivada, y le ayuda a continuar con sus estudios. Es por esto que cada vez que aprueba una materia, chequea después en el sistema informático la cantidad de materias que figuran registradas. A veces encuentra que esta cantidad no ha variado, puesto que no poseía registradas todas las correlativas de la materia recientemente aprobada. Otras veces, recibe la grata sorpresa de que la cantidad de materias registradas ha aumentado. En ocasiones, el aumento puede incluso ser en más de uno, lo cual ocurre cuando la materia que aprobó estaba en condiciones de ser registrada, y al serlo destrabó el registro de una serie de materias aprobadas con anterioridad, que ahora están en condiciones de ser a su vez registradas por el sistema.

Gabina ya tiene planeado el orden en que aprobará todas las materias de su carrera. Desea ahora determinar la cantidad de materias que figurarán registradas en el sistema luego de aprobar cada una de ellas. Su tarea es escribir un programa que ayude a Gabina a predecir esto, para que ella pueda terminar felizmente la carrera de Ciencias de la Contentura.

Entrada

La primera línea contiene dos enteros N y M , que representan la cantidad de materias de la carrera y la cantidad de relaciones de correlatividad entre pares de materias, respectivamente ($1 \leq N, M \leq 5 \times 10^4$). Las materias están identificadas por los números del 1 al N . Cada una de las siguientes M líneas contiene dos enteros A y B ($1 \leq A, B \leq N$ con $A \neq B$) indicando que la materia A es correlativa de B . Esto significa que la materia A debe estar registrada como aprobada antes de poder registrar la materia B como aprobada. No hay en la entrada relaciones de correlatividad repetidas ni correlatividades cíclicas. La última línea contiene N números P_1, P_2, \dots, P_N que representan las materias en el orden en el que Gabina las aprobará ($1 \leq P_i \leq N$ para $i = 1, \dots, N$, con $P_i \neq P_j$ para $i \neq j$).

Salida

Imprimir N líneas con un número cada una. El número en la i -ésima línea representa la cantidad de materias registradas en el sistema inmediatamente después de que Gabina apruebe cada una de las materias de la carrera en el orden dado en la entrada.

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

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

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

Problema D — Dibujando triángulos

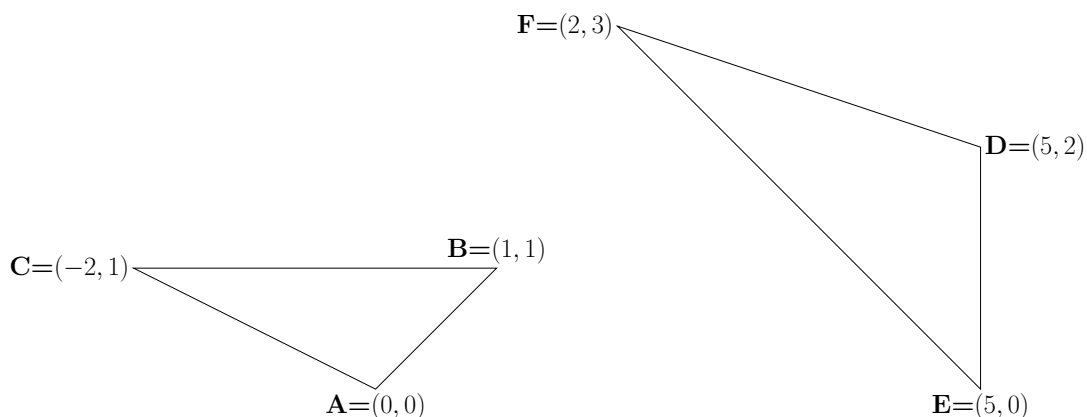
AUTOR: PABLO BLANC - UNIVERSIDAD DE BUENOS AIRES

A Daniela le regalaron un libro para dibujar de *Game of Thrones*. En las hojas del libro hay marcados N puntos que están numerados de 1 a N , y el desafío es unirlos de manera tal que quede dibujado un dragón. Este problema sería muy divertido si se titulara “Dibujando Dragones” y su protagonista fuera Daenerys Targaryen, sin embargo no es así. La protagonista no es

Daenerys de la Tormenta
La Primera de su Nombre
Reina de Meereen
Reina de los Ándalos, los Rhoynar y los Primeros Hombres
Señora de los Siete Reinos
Khaleesi del Gran Mar de Hierba
La que no Arde
Protectora del Reino
Rompedora de Cadenas
Madre de Dragones

sino que es Daniela, y a ella le gusta dibujar triángulos y estudiar sus propiedades. ¡Esto es definitivamente mucho más divertido que dibujar dragones!

Daniela está interesada en los triángulos semejantes. Un triángulo es la figura formada al unir con segmentos tres puntos no alineados. Dos triángulos son semejantes si las razones entre sus lados correspondientes son iguales. En la figura los triángulos ABC y DEF son semejantes pues $\frac{AB}{DE} = \frac{BC}{EF} = \frac{CA}{FD}$.



Daniela lleva un rato mirando la hoja y pensando en el triángulo formado por los primeros tres puntos. Se pregunta cuántos triángulos semejantes al formado por los primeros tres puntos se pueden formar con los puntos marcados en la hoja. Hay muchos puntos y le va a llevar un rato largo encontrar la respuesta. Tiene mucho sueño, pero sabe que no se va a poder dormir sin saberla. Ayúdenla a contar cuántos triángulos semejantes al formado por los primeros tres puntos hay (contando al triángulo formado por los tres primeros puntos), para que pueda irse a dormir tranquila durante la larga noche.

Entrada

La primera línea contiene un entero N , que representa la cantidad de puntos marcados en la hoja ($3 \leq N \leq 1000$). Cada una de las siguientes N líneas contiene dos enteros que corresponden a un punto marcado en la hoja. Los enteros en la i -ésima de estas líneas son X_i e Y_i , y representan las coordenadas del i -ésimo punto en el plano cartesiano ($-100 \leq X_i, Y_i \leq 100$ para $i = 1, 2, \dots, N$). Todos los puntos de la entrada son distintos, y los tres primeros puntos siempre forman un triángulo.

Salida

Imprimir en la salida una línea conteniendo un entero que representa el número de triángulos semejantes al formado por los primeros tres puntos que se pueden formar con vértices en los puntos marcados en la hoja (contando al triángulo formado por los tres primeros puntos).

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

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

Entrada de ejemplo	Salida para la entrada de ejemplo
4 0 0 12 12 3 21 28 -4	3

Entrada de ejemplo	Salida para la entrada de ejemplo
4 -100 -100 -100 100 100 -100 100 100	4

Problema E — El tío quejoso

AUTOR: LEOPOLDO TARAVILSE - UNIVERSIDAD DE BUENOS AIRES

Los inviernos en Nlogonia son muy duros, por lo que el tío Ernie se decidió a comprar un caloventor para no pasar frío este año. Fue terriblemente difícil conseguirlo, pero siguiendo el consejo de sus amigos compró uno inteligente, que puede ser manejado desde el teléfono celular. Sin embargo, el tío no entiende muy bien su teléfono celular, por lo que le cuesta trabajo encontrar la aplicación adecuada para ajustar la temperatura del caloventor.

El tío Ernie tiene instaladas en su celular N aplicaciones que están numeradas del 1 al N , correspondiendo el número 1 a la aplicación que controla el caloventor. El celular tiene M botones numerados del 1 al M , que sirven para pasar de una aplicación a otra. Más específicamente, si el celular tiene abierta la aplicación i y el tío aprieta el botón j , entonces la aplicación i se cierra, y se abre luego la aplicación $T_{i,j}$. El problema es que el tío no puede distinguir entre las distintas aplicaciones, de modo que nunca puede saber si tiene abierta la aplicación correcta.

Al tío Ernie le gusta quejarse por cualquier cosa, de modo que han decidido ayudarlo para evitar escuchar sus quejas cada vez que la temperatura del caloventor no sea la adecuada. Su tarea es proporcionarle una lista de botones a modo de instrucciones, tal que al apretar los botones de la lista en el orden en el que aparecen en la misma, el teléfono del tío tenga abierta la aplicación que controla el caloventor. Como no quieren darle más de una lista, deben confeccionar una que funcione correctamente independientemente de cuál sea la aplicación que está abierta al momento de comenzar a ejecutar las instrucciones.

Consideremos por ejemplo el caso en el que el teléfono tiene $N = 3$ aplicaciones y $M = 2$ botones, siendo $T_{1,1} = T_{2,1} = 3$, $T_{3,1} = T_{1,2} = 2$ y $T_{2,2} = T_{3,2} = 1$. En este caso, una secuencia de botones que podrían darle al tío sería $\{1, 2\}$, ya que al apretarlos ocurrirá alguna de las siguientes situaciones:

- Si el tío empieza con la aplicación 1 abierta, al apretar el botón 1 la aplicación abierta pasa a ser la 3; luego al apretar el botón 2 vuelve a tener abierta nuevamente la aplicación 1.
- En cambio, si el tío empieza en la aplicación 2 al apretar el botón 1 pasa a tener abierta la aplicación 3; entonces al apretar el botón 2 la aplicación abierta será finalmente la 1.
- Por último, si empieza con la aplicación 3 abierta, al apretar el botón 1 pasa a tener abierta la aplicación 2; luego al apretar el botón 2 pasará a tener abierta la aplicación 1.

Por lo tanto, independientemente de la aplicación que esté abierta al comenzar a apretar la secuencia de botones, el tío siempre llegará a la aplicación 1 al concluirla.

Ahora bien, en ocasiones es imposible encontrar una secuencia de botones para darle al tío tal que la aplicación que quede abierta en el celular al terminar de seguir las instrucciones sea siempre la 1. Por ejemplo, en el caso con $N = 3$ y $M = 2$ si tenemos $T_{1,1} = T_{1,2} = 2$, $T_{2,1} = T_{2,2} = 3$ y $T_{3,1} = T_{3,2} = 1$ la aplicación abierta al terminar de apretar una secuencia de botones depende de qué aplicación se encontraba abierta al momento de comenzar, y esto independientemente de qué secuencia elijamos. Por lo tanto, en este

caso es imposible lograr que el tío termine una secuencia de instrucciones siempre con la aplicación 1 abierta.

Para no perder tiempo buscando secuencias de botones que no existen, quieren primero determinar si es posible dejar conforme al tío Ernie encontrando una secuencia de botones como se describe arriba. De ser así, el tío podrá colocar el caloventor en el nivel dos, su preferido, y les estará eternamente agradecido.

Entrada

La primera línea contiene dos enteros N y M , que representan la cantidad de aplicaciones y de botones que tiene el celular del tío Ernie, respectivamente ($1 \leq N, M \leq 1000$ con $1 \leq N \times M \leq 10^4$). Cada una de las siguientes N líneas contiene M enteros, siendo el j -ésimo entero en la i -ésima línea $T_{i,j}$, que representa la aplicación que se abre cuando apretamos el botón j teniendo abierta la aplicación i ($1 \leq T_{i,j} \leq N$ para $i = 1, 2, \dots, N$ y $j = 1, 2, \dots, M$).

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si es posible hallar una secuencia de botones como se pide en el enunciado. El carácter debe ser una 'S' si la secuencia puede ser hallada, y una 'N' en caso contrario.

Entrada de ejemplo	Salida para la entrada de ejemplo
3 2 3 2 3 1 2 1	S

Entrada de ejemplo	Salida para la entrada de ejemplo
3 2 2 2 3 3 1 1	N

Problema F — ¡Felicitaciones, Fidel!

AUTOR: LEOPOLDO TARAVILSE - UNIVERSIDAD DE BUENOS AIRES

Fidel ha finalizado una fructífera etapa de su formación en la facultad, obteniendo su doctorado en física. Esto es fundamentalmente fruto de la firmeza a la hora de formalizar los fabulosos resultados de su fantástica investigación.

Fidel ha fomentado siempre la felicidad entre sus amigos y familiares, por lo que hemos decidido firmarle una felicitación a nuestro doctor favorito. Habiendo finalizado la carta de felicitación, solo resta incorporar la fervorosa firma de los F firmantes. Para esto, hemos comprado dos fibrones, uno de color azul Francia, y el otro de color fucsia, con los que cada firmante escribirá su nombre.

Es nuestra intención hacer foco en el título de doctor que acaba de obtener Fidel, y por lo tanto queremos que en nuestras firmas se vea camuflada muchas veces la abreviación de doctor (“DR”). Para esto, firmaremos utilizando ambos colores, y escribiremos algunas letras en azul y otras en fucsia, de modo tal que si Fidel lee sólo las letras de color fucsia pueda leer la sigla “DR”.

Nuestro objetivo es que, al leer únicamente las letras escritas en fucsia, Fidel sólo lea las letras ‘D’ y ‘R’ de manera alternada. Por lo tanto, la primera letra de color fucsia debe ser una ‘D’, y para cada letra ‘D’ que está escrita en fucsia, la próxima letra de ese color deberá ser una ‘R’. Análogamente, para cada letra ‘R’ de color fucsia la siguiente letra de ese color deberá ser una ‘D’, siendo entonces una ‘R’ la última letra de color fucsia.

Queremos escribir las letras en fucsia de modo tal que Fidel lea las letras ‘D’ y ‘R’ en ese color la mayor cantidad de veces posible. Para cumplir nuestro objetivo, podemos elegir en qué orden escribir nuestros nombres, y qué letras escribir de cada color. Como hay muchas formas de hacer esto, les pedimos ayuda para que nos digan cuál es la mayor cantidad de veces que podemos escribir las siglas “DR” de color fucsia si respetamos las reglas dadas en el párrafo anterior.

Entrada

La primera línea contiene un entero F , representando el número de firmantes ($1 \leq F \leq 1000$). Cada una de las siguientes F líneas contiene el nombre de uno de los amigos de Fidel. El nombre de cada amigo está compuesto por no más de 100 caracteres de la ‘A’ a la ‘Z’.

Salida

Imprimir en la salida una línea conteniendo un entero que representa la máxima cantidad de veces que puede aparecer la sigla “DR” en fucsia al firmar nuestra carta de felicitación, si escribimos las letras ‘D’ y ‘R’ alternadamente como se describe en el enunciado.

Entrada de ejemplo	Salida para la entrada de ejemplo
10 RAMIRO AUGUSTO JOAQUIN JACINTO NICOLAS ALEJANDRO DIJKSTRA KAJITA MCDONALD SCHRODINGER	4

Entrada de ejemplo	Salida para la entrada de ejemplo
4 DDD RRR DRDR RDRD	5

Entrada de ejemplo	Salida para la entrada de ejemplo
12 MELANIE DAMIAN RAMIRO AUGUSTO JOAQUIN JACINTO NICOLAS ALEJANDRO DIJKSTRA KAJITA MCDONALD SCHRODINGER	5

Entrada de ejemplo	Salida para la entrada de ejemplo
4 ABCEFG HIJKLM NOPQST UVWXYZ	0

Problema G — Gestión eficiente

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

La red de trenes de Nlogonia consta de N estaciones, cada una estratégicamente ubicada en una ciudad distinta del reino. Ciertos pares de estaciones están conectados por vías, y sobre cada vía circula un servicio de trenes en ambos sentidos. Desde hace siglos el *Instituto para la Conexión Perfecta de Ciudades* (ICPC) se encarga de optimizar el transporte público de Nlogonia, y hoy en día el sistema ferroviario es tan eficiente que existe exactamente una forma de viajar en tren entre cualquier par de ciudades. Para ello, es posible que el pasajero deba tomar varios trenes sucesivamente, en caso de que no exista una conexión directa entre las estaciones de las ciudades entre las que va a viajar. En otros parajes podría considerarse que esto es un inconveniente, pero los habitantes de Nlogonia son felices sabiendo que nunca deben perder tiempo pensando en qué camino tomar para ir de una ciudad a otra.

El pasaje para cada servicio de trenes tiene un costo determinado, de modo que cuando un pasajero viaja entre dos ciudades tomando uno o más trenes debe comprar el boleto correspondiente antes de subir a cada uno de ellos. La moneda de Nlogonia también es extremadamente eficiente, porque existen billetes con valores equivalentes a todas las potencias no negativas de dos. Esto es, la denominación de los billetes de Nlogonia es de $2^0 = 1$ unidad, $2^1 = 2$ unidades, $2^2 = 4$ unidades, y así siguiendo. Como consecuencia de esta eficiencia monetaria, los habitantes de Nlogonia siempre pagan sus boletos entregando la mínima cantidad de billetes con los que es posible alcanzar el valor exacto del pasaje que van a comprar.

Para agilizar la compra de los pasajes, la *Agencia de Cobro Minucioso* (ACM) desea introducir la siguiente promoción. Cuando un pasajero va a realizar un viaje, puede pagar todos los pasajes que va a necesitar con antelación. Al hacerlo, debe presentar todos los billetes que utilizaría a lo largo de su recorrido, y la ACM tomará solamente uno de cada denominación para la cual la cantidad entregada de dicha denominación sea impar. De esta forma, si un pasajero por ejemplo desea comprar tres boletos con valores de 3, 7 y 10 unidades, entregará dos billetes para el primero (con denominaciones 1 y 2), tres billetes para el segundo (con denominaciones 1, 2 y 4), y dos billetes para el tercero (con denominaciones 2 y 8). La ACM tomará entonces solamente un billete de denominación 2, junto con los de denominaciones 4 y 8, devolviendo al pasajero dos billetes de denominación 1 y otros dos de denominación 2.

Ahora bien, el comité directivo de la ACM está preocupado porque considera que esta promoción puede llegar a ser demasiado costosa para las arcas del reino. La preocupación está justificada, porque nótese que incluso es posible viajar gratuitamente (por ejemplo, cualquier viaje de ida y vuelta será gratuito, ya que se requerirá un número par de pasajes de cada valor). Su tarea es averiguar hasta qué punto este puede ser un problema, para lo cual la ACM les ha encomendado determinar cuál es el precio máximo que puede tener que pagar un pasajero que viaje partiendo de cada una de las N estaciones de Nlogonia.

Entrada

La primera línea contiene un entero N , indicando la cantidad de estaciones de tren en Nlogonia ($2 \leq N \leq 10^5$). Las estaciones de tren de Nlogonia están identificadas por los

números del 1 al N . Cada una de las siguientes $N - 1$ líneas contiene tres enteros A , B y C , indicando que existe una vía que conecta directamente las estaciones A y B , siendo C el precio del pasaje del servicio de trenes que circula por dicha vía ($1 \leq A, B \leq N$ y $1 \leq C \leq 10^9$, con $A \neq B$). La descripción del sistema ferroviario siempre es tal que para todo par de estaciones distintas existe exactamente una secuencia de servicios de tren que las conecta.

Salida

Imprimir N líneas conteniendo un entero cada una. El entero impreso en la i -ésima línea debe corresponder al máximo valor de los pasajes que puede llegar a pagar un pasajero que inicie su viaje en la estación identificada por el número i , cuando se aplica la promoción descrita en el enunciado.

Entrada de ejemplo	Salida para la entrada de ejemplo
4	14
1 2 3	13
2 3 7	10
3 4 10	14

Entrada de ejemplo	Salida para la entrada de ejemplo
6	7
1 2 1	7
3 2 2	5
2 4 3	5
4 5 4	7
4 6 5	7

Entrada de ejemplo	Salida para la entrada de ejemplo
7	6
1 2 1	7
1 3 2	7
1 4 3	7
1 5 4	7
1 6 5	7
1 7 6	7

Problema H – Habemus nuevo TAP

AUTOR: PABLO BLANC - UNIVERSIDAD DE BUENOS AIRES

Estamos considerando cambiar las reglas del Torneo Argentino de Programación a partir del año que viene. Pero antes, los organizadores necesitamos evaluar cuán justo es el nuevo sistema, y para eso necesitamos que nos ayuden.

El nuevo torneo se va a desarrollar con N equipos en $N - 1$ rondas. En cada ronda dos equipos se van a enfrentar compitiendo por ser los primeros en resolver un problema, y el equipo perdedor quedará eliminado. En la primera ronda se elegirán dos equipos al azar y el perdedor será ubicado en el último lugar en la tabla de posiciones, mientras que el ganador seguirá en competencia. En cada una de las siguientes rondas se elegirán dos equipos al azar entre los que sigan en la competencia, y el perdedor será ubicado en el último lugar entre los restantes, quedando fuera del torneo.

Por ejemplo, si en el torneo participan cuatro equipos, “aWArush”, “Buen Kilo de Pan Flauta”, “Melarita” y “Type Mismatch”, el torneo se desarrollará en tres rondas. Supongamos que en la primera se enfrentan “Buen Kilo de Pan Flauta” y “Melarita”, resultando el primero ganador; en la segunda ronda “aWArush” vence a “Buen Kilo de Pan Flauta”; y finalmente en la última ronda “aWArush” vence a “Type Mismatch”. Entonces los equipos quedarán en la tabla de posiciones en el siguiente orden: 1^{ro} “aWArush”, 2^{do} “Type Mismatch”, 3^{ro} “Buen Kilo de Pan Flauta” y 4^{to} “Melarita”.

Para analizar cuán justo es el nuevo formato del torneo, vamos a considerar a los equipos numerados de 1 a N de forma tal que a menor número sea mejor el equipo. Vamos a suponer entonces que si en una ronda se enfrentan dos equipos, el de número más pequeño ganará indefectiblemente. Queremos que nos ayuden a responder la siguiente pregunta: ¿Cuál es la probabilidad de que el equipo X quede en la posición Y -ésima?

Entrada

Una línea conteniendo tres enteros N , X e Y . El entero N representa la cantidad de equipos participantes en el torneo ($2 \leq N \leq 1000$), X representa el número del equipo e Y representa la posición final ($1 \leq X, Y \leq N$).

Salida

Imprimir en la salida una línea conteniendo un racional que representa la probabilidad de que el equipo con número X termine el torneo en la posición Y -ésima. Imprimir el resultado con exactamente 4 dígitos luego del punto decimal, redondeando de ser necesario.

Entrada de ejemplo	Salida para la entrada de ejemplo
3 2 2	0.6667

Entrada de ejemplo	Salida para la entrada de ejemplo
10 3 6	0.0946

Entrada de ejemplo	Salida para la entrada de ejemplo
10 1 5	0.0000

Entrada de ejemplo	Salida para la entrada de ejemplo
1000 1 1	1.0000

Entrada de ejemplo	Salida para la entrada de ejemplo
1000 1000 1000	0.0020

Problema I — Invasión de insectos

AUTOR: PABLO BLANC - UNIVERSIDAD DE BUENOS AIRES

Ignacio solía divertirse participando en competencias de programación como el TAP durante sus años de estudio en la universidad. Era muy feliz, y cuando se recibió, consiguió un buen trabajo. Sin embargo, con el paso del tiempo la rutina y la vida en la gran ciudad lo fueron agobiando. Un día, harto de todo eso, decidió mudarse al campo y empezar una nueva vida como granjero. No tenía mucho dinero ahorrado, pero logró comprarse un campo con forma circular.

Su vida como granjero no tuvo un buen comienzo. Antes de poder disfrutar de su primera cosecha las desgracias comenzaron. En el centro de su campo un espantapájaros se encargaba de mantener las aves a raya, pero por alguna extraña razón estaba conectado a un escape de gas radiactivo proveniente de una planta nuclear cercana, y una mañana el gas se liberó destruyendo casi todo su campo. Ignacio no pudo hacer nada al respecto, solo una pequeña franja en el borde de su campo quedó ilesa y utilizable. Eso no fue todo, ya que las pocas plantas que le quedaban fueron luego atacadas por una invasión de insectos mutantes. Esta vez Ignacio no podía quedarse de brazos cruzados: decidió combatir a los insectos con ranas entrenadas.

En el borde de su campo circular creó N charcos para las ranas, y los numeró en sentido horario de 1 a N . En una casa especializada en ranas de circo compró R ranas numeradas de 1 a R . Durante la noche colocó las ranas en los charcos, ubicando a la rana i -ésima en el charco B_i . Las ranas están muy bien entrenadas, y con la primera luz del sol comenzarán a saltar realizando cada una un salto por minuto. Cada rana repite un patrón de saltos cada K minutos. La rana i -ésima en el primer minuto saltará avanzando $A_{i,1}$ charcos en sentido horario, luego saltará avanzando $A_{i,2}$ charcos en el mismo sentido, y así siguiendo hasta el K -ésimo minuto cuando saltará avanzando $A_{i,K}$ charcos. Luego repetirá el patrón saltando en el minuto $K + 1$ para avanzar $A_{i,1}$ charcos, en el minuto $K + 2$ para avanzar $A_{i,2}$ charcos, etc. Por ejemplo, consideremos el caso en el que hay $N = 5$ charcos y $K = 3$. En este caso, si la rana número 1 comienza en el charco $B_1 = 2$, siendo su patrón de saltos $A_{1,1} = 1$, $A_{1,2} = 2$ y $A_{1,3} = 1$, en sus primeros saltos recorrerá los charcos en el siguiente orden: 2, 3, 5, 1, 2, 4, 5, 1, 3, 4, 5, ...

Ignacio tiene realmente mucha mala suerte, pues la primera rana sufre de una enfermedad contagiosa que le impide comer insectos. Cuando salga el sol y las ranas comiencen a saltar, si una rana enferma se encuentra en un charco con otra sana, le contagiará su enfermedad. En nuestro ejemplo con $N = 5$ y $K = 3$, si hay $R = 2$ ranas y la segunda rana comienza en el charco $B_2 = 4$, siendo su patrón de saltos $A_{2,1} = 1$, $A_{2,2} = 1$ y $A_{2,3} = 1$, ésta recorrerá los charcos 4, 5, 1, 2, 3, 4, ... Por lo tanto, la primera rana le contagiará la enfermedad a la segunda al cabo de 5 minutos, cuando ambas se encuentren en el charco 4. En general, las ranas se irán infectando hasta que estén todas infectadas o hasta que las que queden sanas ya no se encuentren nunca con las enfermas, alcanzándose en ese momento el número máximo de ranas infectadas.

Escribiendo esta historia se hizo de día, y si bien Ignacio notó que la primera rana está enferma, está tan bien entrenada que no logró atraparla. Va a tener que recurrir directamente a la casa especializada en ranas de circo para presentar una queja. Como quiere pedir un reembolso, debe esperar a que la enfermedad se propague hasta alcanzar el número máximo de ranas infectadas. Ignacio no quiere esperar más tiempo innecesariamente, de modo que para ayudarlo a presentar su queja deben responder dos preguntas:

¿Cuál será el número máximo de ranas infectadas?, y ¿En qué minuto tendrá lugar el último contagio?

Entrada

La primera línea contiene tres enteros N , R y K . El entero N representa la cantidad de charcos en el campo ($2 \leq N \leq 10^9$), R representa la cantidad de ranas ($2 \leq R \leq 200$) y K representa la cantidad de minutos tras los cuales las ranas repiten su patrón de salto ($2 \leq K \leq 200$). La segunda línea contiene R enteros B_1, B_2, \dots, B_R , representando el i -ésimo número la posición inicial de la i -ésima rana ($1 \leq B_i \leq N$ para $i = 1, \dots, R$, con $B_i \neq B_j$ si $i \neq j$). Las siguientes R líneas describen el comportamiento de las ranas. La i -ésima de estas líneas contiene K enteros $A_{i,1}, A_{i,2}, \dots, A_{i,K}$, representando la cantidad de charcos que avanza la i -ésima rana en cada uno de sus K saltos, en el orden en el que los realiza ($1 \leq A_{i,j} < N$ para $i = 1, 2, \dots, R$ y $j = 1, 2, \dots, K$).

Salida

Imprimir en la salida una línea conteniendo dos enteros que representan el número máximo de ranas infectadas y el minuto en el que tendrá lugar el último contagio, respectivamente.

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

Entrada de ejemplo	Salida para la entrada de ejemplo
1234 4 4 23 25 1000 67 20 4 26 222 18 28 1232 222 2 4 6 222 2 2 2 2	3 2

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

Problema J — Juntando líneas

AUTOR: LEOPOLDO TARAVILSE - UNIVERSIDAD DE BUENOS AIRES

Hace ya dos años tuvimos un traspie cuando Joaquín, uno de los jurados, tuvo un accidente que hizo que no pudiéramos incluir el problema “Jugando con listas” en la prueba del TAP. Gracias a los participantes de ese año, que muy amablemente nos ayudaron a solucionar aquel problema, pensábamos incluirlo en la prueba de hoy. Lamentablemente tuvimos un nuevo inconveniente con Jacinto, otro de los jurados.

Ocurre que a Jacinto no le gusta que los casos de prueba que ponemos como ejemplo acompañando los enunciados ocupen más de una página. Para el problema “Jugando con listas”, cada caso de prueba consiste en una única línea con la descripción de una lista. No queremos contarles mucho al respecto porque vamos a usar el problema el año que viene, así que solo les vamos a dar la cantidad total de caracteres que ocupa cada caso de prueba, aclarando que no es posible “partir” un caso de prueba para escribirlo en múltiples líneas.

Queremos escribir la entrada de los N casos de prueba de ejemplo en una única página, en la que caben a lo sumo L líneas de C caracteres cada una. El inconveniente surge si hay más casos de prueba que líneas disponibles, de modo que no podemos escribirlos todos en líneas distintas. Para solucionar este problema, Jacinto sugirió trazar segmentos verticales que recorran toda la extensión de la página, dividiéndola a en dos o más columnas. Los segmentos tienen un ancho despreciable, de modo que no reducen la cantidad de caracteres que podemos escribir en la hoja, y actúan visualmente como separadores que dividen cada una de las líneas atravesadas. De esta manera podremos escribir cada caso de prueba en una línea perteneciente a alguna de las columnas, siempre y cuando no atraviesen los segmentos verticales. El orden en el que se colocan los casos de prueba es irrelevante.

Por ejemplo, consideremos la situación en la que tenemos que escribir $N = 5$ casos de prueba en una hoja en la que caben $L = 3$ líneas de $C = 11$ caracteres cada una. Si los casos de prueba tienen $K_1 = 3$, $K_2 = 4$, $K_3 = 5$, $K_4 = 6$ y $K_5 = 7$ caracteres, entonces podemos dividir la hoja en dos columnas de forma tal que una columna tenga 7 caracteres de ancho y la otra 4. En la columna más grande podemos colocar los casos de $K_3 = 5$, $K_4 = 6$ y $K_5 = 7$ caracteres en algún orden, mientras que en la otra podemos escribir los casos de $K_1 = 3$ y $K_2 = 4$ caracteres, nuevamente en cualquier orden.

Dos de las formas en las que podemos acomodar los casos de prueba en este ejemplo son entonces las siguientes

← 5/7 →	← 3/4 →		← 7/7 →
← 7/7 →	← 4/4 →	← 4/4 →	← 5/7 →
← 6/7 →		← 3/4 →	← 6/7 →

donde por ejemplo 5/7 indica que se usaron 5 de los 7 caracteres disponibles de la columna para un caso de prueba en la línea correspondiente.

En una situación análoga en la que cupieran solamente $C = 10$ caracteres por línea, necesitaríamos una columna de al menos 7 caracteres para poder escribir el caso de mayor longitud. Por lo tanto, no podríamos tener otra columna de más de 3 caracteres, y esto implicaría que 4 de los $N = 5$ casos de prueba irían en la columna más grande. Claramente esto es imposible, pues hay sólo $L = 3$ líneas en la hoja. Por otra parte, notemos que si bien los casos de $K_1 = 3$ y $K_5 = 7$ caracteres pueden en principio ser

escritos en una única línea, lo mismo que los casos de $K_2 = 4$ y $K_4 = 6$ caracteres, es imposible colocarlos simultáneamente de esta forma ya que el ancho de cada columna debe ser el mismo para todas las líneas.

Ahora les pedimos que nos ayuden a determinar si es posible dejar conforme a Jacinto, colocando todos los casos de prueba en una misma hoja según se describe más arriba.

Entrada

La primera línea contiene tres enteros N , L y C . El entero N representa la cantidad de casos de prueba, L representa la cantidad de líneas en la hoja, y C representa la cantidad máxima de caracteres por línea ($1 \leq N, L, C \leq 5000$). La segunda línea contiene N enteros K_1, K_2, \dots, K_N , que representan la cantidad de caracteres de los casos de prueba ($1 \leq K_i \leq C$ para $i = 1, 2, \dots, N$).

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si es posible escribir todos los casos de prueba en una misma hoja como se describe en el enunciado. El carácter debe ser una 'S' si esto es posible, y una 'N' caso contrario.

Entrada de ejemplo	Salida para la entrada de ejemplo
5 3 11 3 4 5 6 7	S

Entrada de ejemplo	Salida para la entrada de ejemplo
5 3 10 3 4 5 6 7	N

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

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

Problema K — Koalas

AUTOR: FRANCO MARINO - UNIVERSIDAD NACIONAL DE CÓRDOBA

Mabel Eucaliptos ha pasado toda la noche entrenándose en el arte de comer hojas de eucalipto. Finalmente está preparada para enfrentar a su malvada archinémisis, Pacífica, en un último juego que intentará decidir de una vez por todas quién de las dos es la mejor koala.

El juego se llevará a cabo en un bosque constituido por N árboles de eucalipto numerados del 1 al N . Los árboles están conectados por $N - 1$ cuerdas. Cada cuerda conecta dos árboles diferentes, y permite a las koalas desplazarse de cualquiera de ellos al otro. El bosque de eucaliptos es tal que es posible ir de cualquier árbol a cualquier otro usando sucesivamente estas cuerdas.

Los árboles de eucalipto contienen una cantidad no negativa de hojas. Cuando un árbol contiene cero hojas, se dice que está vacío. Inicialmente ninguno de los N árboles en los que se desarrollará el juego se encuentra vacío.

Antes de empezar el juego, a cada koala se le asigna un árbol diferente. Al principio de la partida cada jugadora sube al árbol que le fue asignado y come todas las hojas que este contiene, dejándolo vacío. A continuación, juegan alternadamente, siendo Mabel la encargada de realizar el primer movimiento. En cada turno la jugadora correspondiente se mueve a un árbol no vacío que esté conectado por una cuerda con el árbol en el que se encuentra ella. Seguidamente come todas las hojas que este nuevo árbol contiene, dejándolo vacío. En caso de no poder realizar un movimiento válido, permanece donde está y pasa a ser el turno de la otra jugadora. El juego termina cuando ninguna de las dos puede hacer un movimiento válido.

Una vez finalizada la partida, se cuentan las hojas que comió cada koala, y se calcula la diferencia entre la cantidad que comió Mabel y la cantidad que comió Pacífica. Mabel jugará tratando de maximizar dicha diferencia, mientras que Pacífica lo hará intentando minimizarla. Su tarea es determinar cuál será el resultado del juego, suponiendo que ambas juegan de manera óptima.

Entrada

La primera línea contiene tres enteros N , M y P , indicando la cantidad de árboles, el árbol desde el que empieza Mabel, y el árbol desde el que empieza Pacífica, respectivamente ($2 \leq N \leq 10^5$ y $1 \leq M, P \leq N$ con $M \neq P$). La segunda línea contiene N enteros C_1, C_2, \dots, C_N , representando C_i la cantidad de hojas que contiene el i -ésimo árbol ($1 \leq C_i \leq 100$ para $i = 1, 2, \dots, N$). Cada una de las siguientes $N - 1$ líneas contiene dos enteros U y V , indicando que hay una cuerda que conecta a los árboles U y V ($1 \leq U, V \leq N$ con $U \neq V$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la diferencia entre la cantidad de hojas que comerá Mabel y la cantidad que comerá Pacífica si ambas juegan de manera óptima.

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

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

Problema L — Leonardo de Pisa

AUTOR: GABRIEL POESIA - UNIVERSIDADE FEDERAL DE MINAS GERAIS

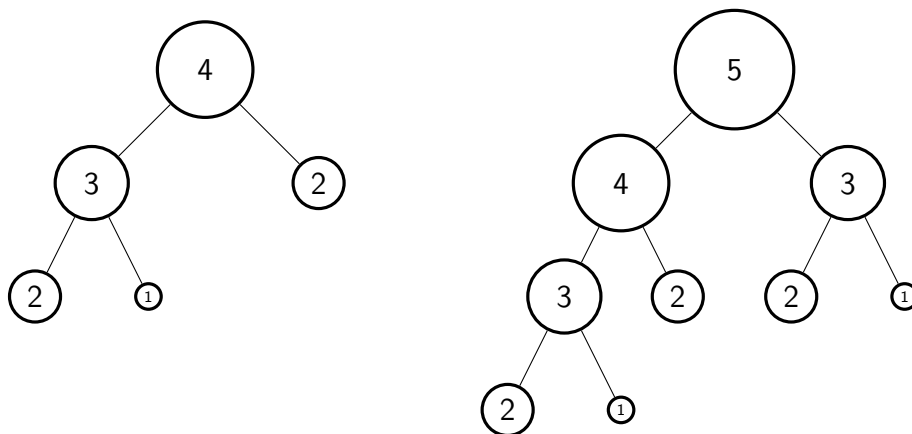
Leonardo de Pisa es un hombre muy precavido, y aún faltando varios meses, él ya compró su árbol de Navidad. Es un árbol muy muy alto, más alto que la torre de Pisa. Leonardo quiere adornar su árbol con pelotitas y luces de colores. Para ello, compró muchas pelotitas de cada diámetro entero entre 1 y N . De hecho, compró tantas pelotitas que no sabe qué hacer con todas ellas.

En Pisa, cada pelotita tiene dos hilos que cuelgan de ella en donde se pueden enganchar otras pelotitas. Así, se aseguran de que las pelotitas no se caigan del árbol y rueden por el piso hasta perderlas de vista abajo de un mueble. Todos los hilos de todas las pelotitas miden 20 centímetros.

Como buen árbol de Navidad, el árbol de Leonardo tiene una punta. En ella, Leonardo colocará una pelotita de diámetro N , por ser la más llamativa. Todas las otras pelotitas estarán colgadas de la punta de forma directa o a través de otras pelotitas. Leonardo estudió cómo necesita colgar las pelotitas para que su árbol sea el más lindo de toda Pisa, y llegó a las siguientes conclusiones:

- Ninguna pelotita de diámetro 1 o 2 deberá tener otra pelotita colgando de ella.
- Toda pelotita de diámetro $k \geq 3$ deberá tener dos pelotitas colgadas: una de diámetro $k - 1$ y otra de diámetro $k - 2$.

A continuación podemos ver dos ejemplos de cómo quedaría el árbol de Leonardo luego de adornarlo con pelotitas. La figura a la izquierda corresponde al caso en el que él compra las pelotitas de diámetro hasta $N = 4$, mientras que la figura a la derecha corresponde al caso con pelotitas de diámetro hasta $N = 5$ (el número en cada pelotita representa su diámetro).

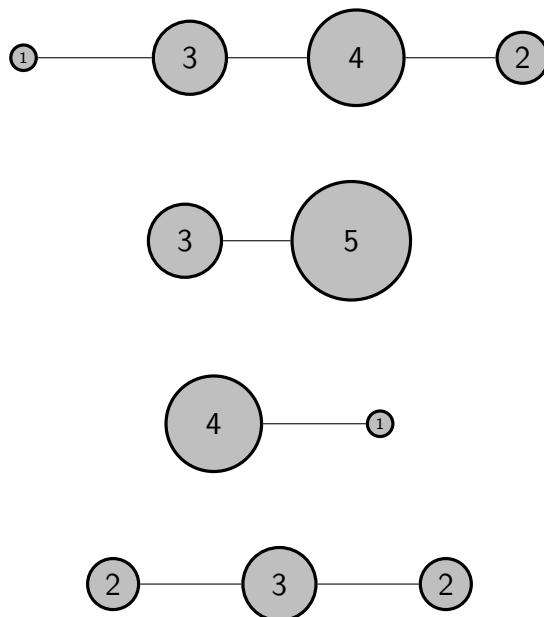


Siempre hay espacio para poner todas las pelotitas que Leonardo quiera, pues su árbol es increíblemente grande. Sin embargo, él siente que su árbol aún no es el más lindo de toda la ciudad: ¡le falta tener luces de colores!

Leonardo compró una tira de luces especial para árboles con pelotitas. La tira tiene K luces unidas por un cable, estando las luces separadas entre sí por 20 centímetros. Cada

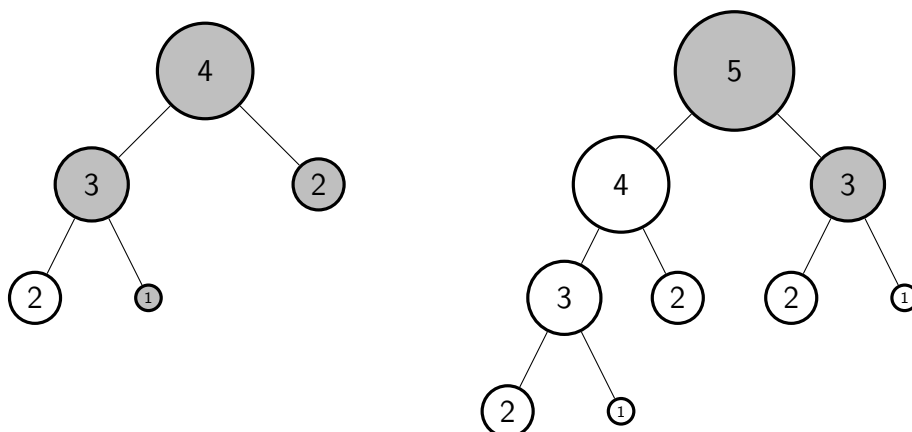
luz se encastra perfectamente a una pelotita dependiendo del diámetro de la misma: una luz de tipo i sólo se puede encastrar en una pelotita de diámetro i , para $i = 1, 2, \dots, N$. Si el diámetro de la pelotita es mayor a i la luz no entrará, y si es menor a i la luz se caerá. No se puede encastrar dos luces en una misma pelotita, y el cable entre las luces debe estar tensado. Esto significa, en particular, que si entre dos pelotitas no hay hilo entonces su distancia no será de exactamente 20 centímetros, y por lo tanto no se podrá colocar dos luces consecutivas en ellas.

A continuación se ven cuatro tiras de luces diferentes, en color gris.



Cuando Leonardo compró la tira de luces ya había adornado su árbol con pelotitas. Le costó tanto esfuerzo hacerlo que decidió que no va a quitar, poner, ni cambiar de lugar ninguna pelotita. Ahora no sabe si va a poder usar la tira que compró, ya que necesita encontrar una secuencia de pelotitas que estén colgadas unas de otras, y tengan los diámetros exactos que entran en la tira.

Por ejemplo, la primera tira mostrada puede colocarse en cualquiera de los dos árboles; la segunda sólo puede colocarse en el segundo árbol; la tercera y la cuarta no pueden colocarse en ningún árbol. A continuación se puede ver la primera tira colocada en el primer árbol y la segunda tira colocada en el segundo árbol.



Ayuden a Leonardo a saber, dada la tira de luces y el diámetro N de la pelotita más grande que compró, si es posible colocar la tira de luces en su árbol.

Entrada

La primera línea contiene dos enteros N y K , representando N el diámetro máximo de las pelotitas y K la cantidad de luces que hay en la tira ($2 \leq N, K \leq 10^5$). La segunda línea contiene K enteros L_1, L_2, \dots, L_K que describen la tira de luces. El i -ésimo entero L_i representa el tipo de la i -ésima luz de la tira ($1 \leq L_i \leq N$ para $i = 1, 2, \dots, K$).

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si Leonardo podrá colocar la tira de luces o no. El carácter debe ser una 'S' si Leonardo puede colocar la tira de luces, y una 'N' en caso contrario.

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

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

Entrada de ejemplo	Salida para la entrada de ejemplo
5 2 3 5	S

Entrada de ejemplo	Salida para la entrada de ejemplo
4 2 4 1	N

Entrada de ejemplo	Salida para la entrada de ejemplo
6 3 2 3 2	N

Entrada de ejemplo	Salida para la entrada de ejemplo
8 4 2 3 3 1	N

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