

Problema A. Amistad

Autor: *Lucio Rivera*
Tiempo límite: 1 segundo
Memoria límite: 256 megabytes

En algún tranquilo y pacífico prado de Lisse en los Países Bajos, *Bram* se encuentra inmerso en un cálido cortejo con todos los amigos que ha abrazado a lo largo de su vida. El calor de la fogata sobre la que reposan dota de ligereza al sereno, pero a la vez amigable ambiente.

Después de tantas horas de diversión y jocosidad, Bram decide que es el momento idóneo para poner a prueba la afinidad del grupo proponiéndole a sus invitados intentar formar un *Círculo de la Amistad*.

La creación de un *Círculo de la Amistad* implica colocar un tronco por cada invitado alrededor de la fogata tal que estos formen un círculo y en donde cada invitado elijirá donde sentarse. Luego Bram, quien se encuentra con los ojos vendados, dirá el nombre de cada uno de los invitados quienes, a medida que van siendo nombrados, abandonan el círculo.

Se dice que un grupo sortea la prueba del *Círculo de la Amistad* si cada vez que un invitado abandona el círculo, el siguiente en hacerlo es aquel que se encontraba inmediatamente a la derecha de este.

Su tarea consiste en proporcionar a *Bram* un programa que le permita determinar si sus amigos superaron la prueba dado un vector circular ordenado de la misma forma en la que estaban dispuestos con el momento en el que cada uno de ellos abandonó el círculo.

Entrada

La entrada consta de múltiples líneas. La primera línea contiene un entero $1 \leq n \leq 10^5$, que representa la cantidad de invitados. La siguiente línea contiene n números enteros $1 \leq m_i \leq n$, representando el orden en el que el invitado ubicado en la posición i abandona el círculo.

Salida

La salida consta de una línea, la palabra "SI" en caso de que la prueba sea superada y "NO" en caso contrario.

Ejemplos

entrada estándar	salida estándar
10 7 8 9 10 1 2 3 4 5 6	SI
7 1 2 3 4 6 5 7	NO

Problema B. Bug en la Matrix

Autor: *Fernando Sebastián Lomazzi*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

Julián inventó un artefacto revolucionario, el *SIMP (Sistema Ideal de Multivisión Profunda)*. Este artefacto consiste en unas gafas de realidad aumentada las cuales permiten ver el código genotípico subyacente de una persona, el cual determina características intrínsecas de la misma, nombrado ahora como la *Matrix*. Julián notó que cuando una persona posea algún tipo de enfermedad, este código posea un patrón particular, se encontraba infectado de *bugs*.

El código genotípico de una persona está determinado por una matriz de tamaño $n \times m$ (n filas y m columnas), compuesta por letras mayúsculas y minúsculas del alfabeto inglés que representan características de la persona. El nivel de enfermedad de una persona se mide de acuerdo a la cantidad de veces que aparece la palabra *bug*, en forma horizontal o vertical sin diferenciar entre mayúsculas y minúsculas en su código genotípico.

Esta característica permitiría determinar enfermedades no diagnosticadas de las personas, por lo que es importante poder, en base al código genotípico de una persona, encontrar su nivel de enfermedad. Ayuda a Julián a poder determinar el nivel de enfermedad de las personas.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene dos enteros $1 \leq n, m \leq 10^3$, la cantidad de filas y columnas de la matriz genotípica. Las siguientes n líneas contienen m letras del alfabeto inglés, en mayúsculas o minúsculas, representando el código genotípico de la persona estudiada.

Salida

La salida consta de una línea, el nivel de enfermedad de la persona en base a su código genotípico.

Ejemplos

entrada estándar	salida estándar
4 5 bgbzb oujau gbgsg cvbbd	2
4 3 bug ugb gbu bug	4

Problema C. Contraseña Segura

Autor: *Lucio Rivera*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

La organización de *Tecnomate* le ha notificado a los autores de esta edición que para próximos años va a ser necesario que cada alumno posea una contraseña segura para darse de alta en el sistema de la competición.

Es bien sabido que los autores de esta edición son expertos en optimización de procesos siempre que esto implique menos horas de trabajo y más horas de ocio, ¡Cómo olvidar aquella vez que le pidieron a una Inteligencia Artificial que generase una competencia de programación!

Dado que aquí han identificado una oportunidad única de hacer uso de sus habilidad de delegación de trabajo, han decidido que los competidores de esta edición serán los encargados de desarrollar la pieza de software que verificará la seguridad de las contraseñas de los participantes en futuras ediciones.

La organización ha establecido que una contraseña s de tamaño n se considerará segura cuando la totalidad de las condiciones siguientes sean satisfechas:

- $5 \leq n \leq 32$.
- n debe ser un número impar.
- El caracter central debe ser un número primo.
- El primer caracter debe ser una letra minúscula.
- El último caracter debe ser una letra mayúscula.

¡Si no resuelves este problema no podrá haber una nueva edición de tecnomate!

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq t \leq 100$, que representa la cantidad de casos de prueba. Cada caso consiste en dos líneas, la primer línea contiene un entero $1 \leq n \leq 10^6$, la longitud de la contraseña s ; la segunda línea contiene la contraseña, compuesta de dígitos del 0 al 9, así como también letras mayúsculas y minúsculas del alfabeto inglés.

Se garantiza que la suma de los tamaños de las contraseñas entre todos los casos de prueba no supera 10^6 .

Salida

Para cada caso de prueba, deberá imprimir la frase *CONTRASENA SEGURA* si la contraseña es segura y *CONTRASENA NO SEGURA* en caso contrario.

Ejemplo

entrada estándar	salida estándar
3	CONTRASENA NO SEGURA
11	CONTRASENA SEGURA
sonMu1VagoS	CONTRASENA NO SEGURA
7	
xms7abC	
14	
demasiadoVagos	

Problema D. DVD

Autor: *Fernando Sebastián Lomazzi*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

Lucas se encuentra viendo la nueva película *Chrek*, acerca de un orco verde y su asno en su reproductor de DVD. En un momento, *Chrek* y *Asno* son emboscados en una trampa planeada por el malvado *Lord Closequaad*, lo que enfureció tanto a *Lucas* que apagó la película y fue a por refrigerios.

A su vuelta, quedó fascinado ante el logo de DVD que se desplazaba en su pantalla. El mismo rebotaba constantemente, hasta que en un momento, chocó con la esquina superior izquierda. Luego de este increíble suceso, *Lucas* se preguntó: ¿Cuándo volverá el logo a golpear otra esquina de la pantalla?

Lucas tomó su regla y midió: su pantalla tiene 4 pulgadas de alto y 7 de ancho, mientras que el logo de DVD tiene 2 pulgadas de alto y 3 de ancho. Notó además que el logo se desplaza una pulgada vertical y horizontalmente cada segundo, y en caso de choque con un borde de la pantalla, cambia su sentido en la dirección de choque.



Lucas le solicita que realice un programa que calcule el tiempo que debe esperar considerando como condición inicial al logo en la esquina superior izquierda, con movimiento en sentido abajo-derecha. Deberá realizarlo para distintas mediciones, ya que desea ayudar a todo amante del cine con su misma pregunta.

Entrada

La entrada consta de múltiples líneas. La primera línea contiene un entero $1 \leq t \leq 10^4$, que representa la cantidad de casos de prueba. Cada caso consiste en dos líneas. La primera línea contiene dos enteros $2 \leq N, M \leq 10^9$, el alto y ancho de la pantalla en pulgadas. La segunda línea contiene dos enteros $1 \leq n < N$ y $1 \leq m < M$, el alto y ancho del logo de DVD en pulgadas.

Salida

Por cada caso de prueba, deberá imprimir la cantidad de segundos que deben transcurrir para que el logo de DVD choque, por primera vez, cualquier esquina de la pantalla partiendo desde la esquina superior izquierda.

Ejemplo

entrada estándar	salida estándar
2	4
4 7	36
2 3	
6 13	
2 4	

Problema E. El computador de la diosa Pitón

Autor: *Lucio Rivera*
Tiempo límite: 1 segundo
Memoria límite: 256 megabytes

Un grupo de investigadores aficionados, bajo la dirección del renombrado experto *Roberto Martilloabierto* (conocido también como *Robert Oppenheimer* en inglés), han puesto en jaque el título que el *EINAC* (*Computador e Integrador Numérico Electrónico*) ostenta como el primer computador de propósito general de la historia de la humanidad. La evidencia sobre la que los investigadores se apoyan se basa en antiguos registros de una civilización poco conocida que floreció en el año 300 a.C., autodenominada *Los Desarrolladores de la diosa Pitón* (conocidos en inglés como *Python Devs*).

Entre las ruinas de la civilización, se encuentran plasmados enigmáticos jeroglíficos con forma de pirámide que Roberto, fervientemente cree, son en realidad registros de memorias utilizadas por los computadores primitivos, los primeros de la historia de la humanidad. El obstáculo inminente que Roberto debe sortear en el corto plazo es calcular muy rápidamente, la suma de los elementos que conforman estas pirámides.

Las pirámides de distintas alturas, aunque pudo observarse que han sido elaboradas siguiendo un mismo patrón entre sí.

```
      1
     2 1 2
    3 2 1 2 3
   4 3 2 1 2 3 4
  5 4 3 2 1 2 3 4 5
```

Pirámide de altura 5.

Nótese que la columna central, de altura m , está compuesta por m elementos con valor 1, mientras que las dos columnas adyacentes constan de $m - 1$ elementos con valor 2, las siguientes constan de $m - 2$ elementos de valor 3 y así sucesivamente hasta llegar a columnas de los extremos, cada una de ellas compuestas por un solo elemento con valor m .

Usted, quien casualmente se encontraba en la zona sin mayores preocupaciones, ha sido reclutado a la fuerza por Roberto para colaborar en su investigación. Él percibe en usted una afinidad especial por las matemáticas y la computación, exactamente las habilidades necesarias para demostrar su teoría.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq n \leq 1000$, que representa la cantidad de casos de prueba. Cada caso consiste en un entero $1 \leq m \leq 10^6$, indicando la altura de la pirámide descrita.

Salida

Por cada caso de prueba, deberá imprimir la suma de los elementos de la pirámide.

Ejemplo

entrada estándar	salida estándar
3	65
5	36
4	343300
100	

Nota

Roberto cree que quizás, la siguiente formula le pueda ser útil.

$$1^2 + 2^2 + \dots + k^2 = \frac{k(k+1)(2k+1)}{6}$$

Problema F. Semejanza

Autor: *Lucio Rivera*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

Ezequiel, flamante estudiante de Ingeniería en Sistemas de Información en la UTN, ha enfrentado a lo largo de su vida incontables burlas por parte de sus compañeros debido a la gran cantidad de dibujos animados a los que Ezequiel se asemeja. Entre todas las comparaciones que ha experimentado, una en particular parece atormentarlo por encima de las demás: Ezequiel y el pez *Óscar* de *Pecezuelos* son prácticamente idénticos. Las pocas diferencias que puedan existir entre ellos resultan prácticamente imperceptibles para el ojo humano.

Agotado por el tormento que enfrenta a diario, Ezequiel ha invertido incansables esfuerzos en investigar diferentes algoritmos de comparación de imágenes. Su objetivo es demostrar que no existe una similitud real entre él y Óscar, basándose en una métrica específica al comparar una fotografía suya con una imagen de Óscar.

El algoritmo que Ezequiel ha seleccionado para llevar a cabo su defensa se llama *FACIL* (*Face Analysis and Color Interpolation Layers*). La métrica FACIL se basa en contar la cantidad de columnas compuestas exclusivamente por ceros, luego de sustituir cada elemento por el resto del cociente entre dicho elemento y el valor mínimo de su fila.

Con la esperanza de que Ezequiel deje de sufrir, se le solicita a usted que escriba un programa capaz de calcular la métrica FACIL de una imagen cuadrada.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq n \leq 1000$, el tamaño en píxeles de la imagen cuadrada sometida al algoritmo. Las siguientes n líneas contienen n enteros $1 \leq p_{ij} \leq 1000$, representando el valor del pixel ubicado en la intersección entre la fila i y la columna j .

Salida

La salida consta de una línea, la métrica FACIL de la imagen dada.

Ejemplos

entrada estándar	salida estándar
4 2 7 6 256 27 3 9 11 5 10 5 125 16 15 4 5	2
3 2 7 5 3 1 9 4 3 4	0

Problema G. Medias

Autor: *Fernando Sebastián Lomazzi*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

Elidan Tromba es un millennial amante del frío, cada tarde toma un café negro junto al fuego de su chimenea a leña mientras admira la belleza de la nieve en su ventana. Su amor es tan, tan profundo que hasta se autoproclama un *Team Invierno*. Además, Elidan es un amante de la ropa de hilo, por lo que tiene suéters, medias, gorros y guantes de todo tipo y color.

El señor Tromba se despertó una mañana fresca como cualquier otra en *Siberia*, su ciudad, con una sensación térmica de -20°C . Calentito en su cama y con la luz apagada, Tromba, buscó en su cajón un par de medias del mismo tipo para vestirse. Al no poder ver debido a la falta de luz y, como levantarse de su cama no era una opción, se preguntó ¿Cuántas medias debería sacar del cajón para asegurarme de tener al menos un par de medias del mismo tipo?

Como Elidan es muy perezoso quisiera sacar la cantidad mínima de medias (ya que volver a ponerlas al cajón requeriría un esfuerzo enoorme), y, al poseer una memoria extraordinaria, se acuerda a la perfección la cantidad de medias que posee de diferentes tipos en el cajón.

Ayuda a Elidan a poder ponerse las medias con el mínimo esfuerzo y sin pasar frío.

Entrada

La entrada consta de dos líneas. La primer línea contiene un entero $1 \leq m \leq 10^5$ que representa la cantidad de tipos de medias diferentes en el cajón. La segunda línea contiene m números enteros $1 \leq m_i \leq 10^9$, representando la cantidad de medias del tipo i en el cajón.

Salida

La salida consta de una línea, la cantidad mínima de medias que se deben sacar del cajón para asegurarse tener al menos un par del mismo tipo o “-1” (sin las comillas) en caso que no sea posible realizar lo solicitado.

Ejemplo

entrada estándar	salida estándar
3 2 4 2	4

Problema H. Ta te ti

Autor: *Fernando Sebastián Lomazzi*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

Alicia es una alegre niña que se encuentra en una extravagante fiesta del té junto a sus curiosos amigos, *el Sombrero Loco* y *el Conejo Blanco*. Entre risas y charlas sin sentido, Alicia propone jugar a un juego de su tierra, el *Ta te ti*. Al escuchar un nombre tan extravagante y, al no tener idea de qué se trataba, sus amigos le pidieron que les enseñe a jugar.

Para dar comienzo al juego, es necesario colocar tantas sillas como participantes alrededor de la mesa redonda del té y cada jugador deberá elegir una para sentarse. Luego, se escoge al azar a un jugador para que diga una frase, en este caso, Alicia dice la frase típica del juego: "Ta te ti suerte para mí, si no es para mi será para ti, ta te ti". Finalmente, por cada sílaba de la frase y, partiendo siempre de la misma silla inicial, se va señalando a un jugador en sentido horario. El jugador que esté siendo apuntado cuando se diga la última sílaba de la frase será el ganador.

Una sílaba maravillosa es toda sílaba compuesta de dos letras del alfabeto inglés, mientras que una palabra maravillosa es una concatenación de sílabas maravillosas. Así *ga-to* es una palabra maravillosa, pero *pe-rro* no lo es. Como Alicia es aún una niña, existen palabras que le son difíciles de pronunciar, aún así, como ella es una niña maravillosa, todas las palabras que dice son maravillosas.

Alicia, ni tonta ni perezosa, le solicita su ayuda para que genere un programa que, dada la cantidad de participantes y la frase dicha por ella, le diga qué silla debe escoger para ganar el juego. Se considera la primer silla la seleccionada para comenzar el juego.

Entrada

La entrada consta de dos líneas. La primer línea contiene dos enteros $1 \leq p \leq n$ y $1 \leq n \leq 10^5$, la cantidad de participantes del juego y el tamaño de la frase dicha por Alicia. La segunda línea contiene la frase dicha por Alicia compuesta de letras del alfabeto inglés, espacios, comas y puntos.

Salida

La salida consta de una línea, la posición en sentido horario de la silla que debe escoger Alicia para ganar el juego.

Ejemplos

entrada estándar	salida estándar
3 63 Ta te ti sute pada mi, si no es pada mi	1 seda pada ti, ta te ti.
5 43 En la casa de Espinete, todo suma hata	2 sete

Problema I. Patos y Burbujas

Autor: *Lucio Rivera*
 Tiempo límite: 1 second
 Memoria límite: 256 megabytes

El eterno enfrentamiento de carácter apocalíptico entre patos y burbujas podría estar llegando a su épico final. La histórica derrota patil en la *Batalla de los Panes de Queso* es quizás un golpe del cual los patos nunca jamás se puedan recuperar.

Para los patos, la situación es crítica. Las burbujas han tomado un control casi total del planeta Tierra, y los pocos patos que aún quedan con vida se encuentran refugiados en el único estanque que todavía no ha sido conquistado por las extremadamente peligrosas y hábiles burbujas. Parece que la aniquilación patil es solo cuestión de tiempo.

Sin embargo, después de más de cien años de contienda y billones de *compatotas* caídos en combate, las honorables aves están decididas a dar batalla hasta el final. Aún, un pequeño rayo de esperanza ilumina el estanque de los patos, ya que entre sus filas se encuentra el experto patógrafo *Pating Turing*.

Resulta que para comunicarse entre sí, las burbujas envían mensajes cifrados utilizando un lenguaje de invención propia llamado *Código Misterio*, que solo puede ser descifrado con un artefacto especial llamado *Máquina Misteriosa*. Si los patos fueran capaces de romper el Código Misterio, podrían adelantarse a los planes de los altos mandos burbujiles y así tal vez recuperar lo que, hace ya muchos años, alguna vez les perteneció.

Los increíbles esfuerzos de Pating Turing han dado sus frutos y los patos se encuentran muy cerca de lograr romper el Código Misterio. El reconocido experto ha llegado a la conclusión de que las burbujas utilizan únicamente tres números para cifrar sus mensajes, pero estos números no son sencillos de obtener. La robustez del Código Misterio radica en que son los primeros tres dígitos de un número a^k extremadamente grande. Pating Turing subestima a las burbujas y no cree que sean capaces de calcular dicho número por sí mismos. Sospecha que las burbujas poseen programas informáticos que les permiten obtener la llave de cifrado.

Usted, desde su estrecha celda, es capaz de observar una aterradora silueta. Pating Turing ha decidido acudir a usted, el único humano y programador que aún vive, para que escriba un programa que les permita romper el Código Misterio a cambio de su libertad.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq t \leq 100$, que representa la cantidad de casos de prueba. Cada caso consiste en dos enteros $2 \leq a \leq 10^6$ y $1 \leq k \leq 10^6$ correspondientes a un número a^k .

Se garantiza que a^k posee como mínimo tres dígitos.

Salida

Para cada caso de prueba, deberá imprimir los dígitos necesarios para vencer a las burbujas.

Ejemplos

standard input	standard output
3	128
2 7	100
100 1	240
7 4	
2	305
500 500	100
1000 1000	

Problema J. Salvación de destreza

Autor: *Fernando Sebastián Lomazzi*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

¡Rayos y centellas! Después de meses de jugar una campaña del famoso juego de rol de mesa *Trasgos y Mazmorras*, Rombo activó una trampa y fue carbonizado por una bola de fuego. Si tan sólo hubiera pasado esa tirada de salvación de destreza para esquivar la llamarada que se le aproximaba...

Siempre con la mente en el pasado, Rombo empieza a cuestionarse cuáles eran realmente sus oportunidades para lograr esa tirada de salvación, dándose cuenta que no es nada trivial. Él tenía desventaja al estar sobre un terreno inestable pero, por otra parte, tenía un modificador de salvación de destreza realmente alto al ser un pícaro...

Una tirada básica de salvación de destreza con dificultad d se realiza de la siguiente forma: Se tira un d20 (un dado de veinte caras). Si el resultado es un 1, la tirada falla automáticamente; si es 20, es un éxito inmediato. En cualquier otro caso, se suma el modificador de salvación de destreza m al resultado del dado. Si la suma es al menos la dificultad d el resultado es exitoso, en caso contrario, es fallido.

Existe un caso particular para esto: en caso que la tirada tenga ventaja o desventaja. En estos casos, dos d20 son tirados en vez de uno, pero solo el mayor (para ventaja) o el menor (para desventaja) es considerado para el cálculo final.

Rombo, recordando haber activado múltiples trampas durante la campaña, le solicita que escriba un programa que, dado los datos básicos para una tirada de salvación de destreza (la dificultad, el modificador de destreza y si poseía ventaja o desventaja), calcule las probabilidades de que esa tirada sea exitosa.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq n \leq 10^5$, que representa la cantidad de casos de prueba. Cada caso consiste en dos líneas, La primer línea contiene dos enteros $0 \leq d \leq 30$ y $-10 \leq m \leq 10$, la dificultad de la tirada y el valor del modificador de la característica. La segunda línea contiene una palabra indicando si la tirada tiene ventaja o desventaja. Esta palabra es "normal" (en caso que no haya ventaja ni desventaja), "ventaja" o "desventaja".

Salida

Por cada caso de prueba, deberá imprimir la probabilidad de que la tirada de salvación de destreza sea exitosa.

Ejemplos

entrada estándar	salida estándar
1 12 4 normal	0.6500000000
2 10 3 ventaja 5 -5 desventaja	0.9100000000 0.3025000000

Problema K. Linaje

Autor: *Lucio Rivera*
 Tiempo límite: 1 segundo
 Memoria límite: 256 megabytes

De todas las dinastías influyentes en la historia argentina, es probable que la familia *Maldomado* sea la más destacada y reconocida de todas ellas. Los fundadores de esta familia, distinguidos ciudadanos ilustres de origen europeo, se caracterizaron por su profundo apego a las tradiciones familiares y preservación de su reconocido linaje. Estos valores han perdurado a lo largo de las generaciones futuras de la dinastía.

Nachain Maldomado XVII, un descendiente lejano del fundador *Dieguinho Maldomado IV* y hermano del legendario *Totein Maldomado XVII*, ha advertido que no todas las familias de la dinastía poseen la misma pureza de sangre. Por lo tanto, ha indagado y posteriormente descubierto un método para evaluar la pureza de una familia Maldomado en función de los nombres de sus miembros vivos.

El método propuesto por *Nachain Maldomado XVII* consiste en obtener la longitud del sufijo común más largo entre los nombres de los miembros de la dinastía en consideración, siendo este su nivel de pureza. Cuanto más largo sea ese sufijo, mayor será la pureza de la familia que lleva con orgullo el apellido Maldomado.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene un entero $1 \leq n \leq 1000$, que representa la cantidad de miembros en la dinastía. Las siguientes n líneas contienen un entero $1 \leq |s_i| \leq 1000$, representando el tamaño del nombre del miembro i -ésimo, y una cadena de caracteres s_i , representando el nombre del miembro i -ésimo, compuesta de letras mayúsculas y minúsculas del alfabeto inglés.

Salida

La salida consta de una línea, el nivel de pureza de sangre de la familia Maldomado dada.

Ejemplos

entrada estándar	salida estándar
2 7 Nachain 6 Totein	2
4 5 messi 4 enzo 8 riquelme 7 lomazzi	0

Problema L. La huida de Biciladrón

Autor: *Lucio Rivera*
 Tiempo límite: 1 second
 Memoria límite: 256 megabytes

La caótica ciudad de Buenos Aires se ha convertido en el escenario de una emocionante persecución. *Biciladrón*, un apasionado defensor de los ciclistas, se encuentra huyendo de las fuerzas policiales de la ciudad, quienes lo persiguen por dañar los espejos laterales de los vehículos circundantes. La pasión de Biciladrón por el ciclismo alcanza niveles extremos, llegando al punto de dañar los automóviles de cualquier conductor que se atreva a invadir su gloriosa y sagrada bicisenda.

Al grito de “¡Asesino!”, “¡Largarto!” y “¡Es ciclovía!”, Biciladrón escapa de la policía, pero no sin antes comenzar a subir el metraje del incidente que capturó con su cámara corporal a las redes. Esto servirá como evidencia necesaria para exponer el comportamiento de los automovilistas ante la sociedad.

Al momento que Biciladrón comienza a subir el metraje a internet, él comienza a escapar de la policía y esta comienza a perseguirlo. Inicialmente, se encuentran en barrios distintos y para transitar a un barrio adyacente deben recorrer el actual, lo que les lleva un tiempo determinado. Existen barrios sin salida en donde solo existe un único barrio adyacente a estos. La transición de un barrio a otro es instantánea, es decir, no consume tiempo adicional.

Diremos que la policía atraparé a Biciladrón cuando esté en un barrio adyacente al suyo y, dado que la policía quiere a atrapar a Biciladrón a toda costa, tomará siempre las mejores decisiones para lograrlo.

Los medios de comunicación están expectantes por saber si Biciladrón tendrá tiempo suficiente para publicar su metraje a las redes en el mismo instante o antes que la policía lo capture por lo que se le solicitó que escriba un programe que permita aliviar la expectativa de la gente.

Entrada

La entrada consta de múltiples líneas. La primer línea contiene cuatro enteros $2 \leq n \leq 10^4$, $0 \leq b \leq n - 1$, $0 \leq p \leq n - 1$ y $0 \leq m \leq 10^9$, representando la cantidad de barrios, la posición inicial de Biciladrón, la posición inicial de la policía y el tiempo necesario para publicar el video. La segunda línea contiene n números enteros $0 \leq t_i \leq 10^5$, representando el tiempo necesario para transitar por completo el barrio i .

Salida

La salida consta de una línea, la palabra “SI” en caso de que Biciladrón tenga una forma de huír tal que pueda subir su video a las redes en el mismo instante o antes de ser atrapado por la policía o “NO” en caso contrario.

Ejemplos

standard input	standard output
4 2 0 2 1 1 1 1	SI
5 1 4 6 1 6 1 2 2	NO