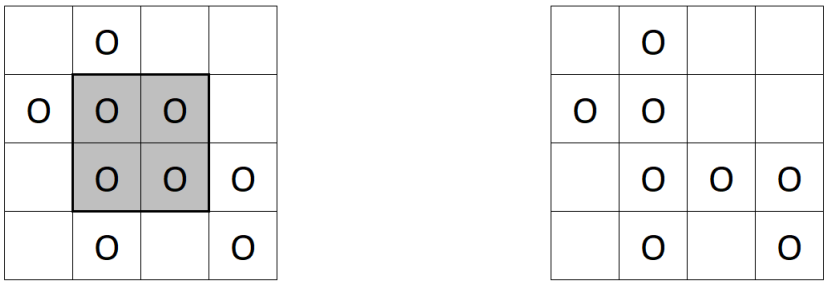
A. Agua Infinita

Autor: Lucio Rivera

En el reconocido videojuego *MinaCrack* —donde los cracks minan metales preciosos en peligrosas cuevas— millones de jugadores alrededor del mundo construyen increíbles refugios y ambiciosos generadores de recursos. Desafortunadamente, muchas personas se han convencido de que MinaCrack es un videojuego diseñado únicamente para niños. Este problema pretende desafiar esa idea y eliminar este prejuicio.

En *MinaCrack* todo está construido utilizando *bloques*, que son la unidad mínima manipulable del videojuego. Dos bloques de agua se consideran conectados entre sí si y solo si uno de ellos es vertical u horizontalmente adyacente al otro. Cuando múltiples bloques de agua conectados se unen, adquieren la capacidad de formar patrones complejos. Estas estructuras se denominan *suministros de agua*.

Una de las muchas mecánicas de MinaCrack que los jugadores pueden aprovechar es la generación de suministros de agua infinitos. Un suministro de agua es infinito si y solo si dentro de su estructura contiene al menos un patrón cuadrado de dos bloques de agua de alto y ancho.



Dos suministros de agua, donde el carácter "O" indica la presencia de un bloque de agua. En la primera imagen, se visualiza un suministro de agua infinita y se destaca el patrón descrito. En la segunda imagen, se visualiza un suministro de agua que no es infinito.

En este problema, se le solicita que escriba un programa para determinar la cantidad de suministros de agua infinita en una disposición de bloques de *MinaCrack*.

Entrada

La entrada consta de múltiples lineas.

La primera línea contiene dos enteros n y m ($1 \leq n, m \leq 1000$), el alto y ancho de la disposición de bloques.

Las siguientes n líneas contienen m caracteres en mayúscula c_{ij} ($c_{ij} \in \{$"O", "X"$\}$) , representando el tipo de bloque ubicado en la intersección entre la fila i y la columna j, donde el carácter "O" indica que el bloque es de agua y el carácter "X" indica que se trata de otro tipo de bloque.

Salida

La salida consta de una línea, la cantidad de suministros de agua infinita presentes en la disposición dada.

Ejemplos

entrada estándar	salida estándar
4 4 X O X X O O O X X O O O X O X O	1
4 4 X O X X O O X X X O O O X O X O	0
5 5 O O X X X O O X X X X O O O O X X O O O O X O X O	1

B. Bananas

Autor: Fernando Sebastián Lomazzi

El espectacular Coco -el mono araña- ha logrado infiltrarse en la celebración secreta del Gran Banquete de la Selva horas antes de que comience. Frente a él se encuentra una enorme variedad de platos exóticos de todos los tipos y colores. Sin embargo, lo que más le interesa a Coco son los r racimos de bananas. Como es un mono de refinados modales, Coco se dedicará a devorar cada racimo de bananas de principio a fin antes de pasar al siguiente. Además, Coco ha decidido que, para no sobrecargarse, cada minuto siempre comerá la misma cantidad de bananas, una cantidad que él mismo elige con mucho cuidado. Si en algún momento un racimo tiene menos bananas de las que ha decidido consumir, se comerá todas las que haya y no comerá más durante ese minuto.

A Coco le encantan las bananas, pero también es un mono prudente y no quiere sentirse mal después de comer. Por eso, siempre prioriza comer lo más lento posible sin exceder el tiempo disponible, es decir, busca la cantidad mínima de bananas por minuto que le permita terminar de comer todas las bananas antes de que lleguen los demás invitados.

Tu tarea es ayudar a Coco a encontrar esa cantidad mínima para que pueda disfrutar de todas las bananas y no ser sorprendido por el resto.

Entrada

La entrada consta de dos líneas.

La primer línea contiene dos enteros r y m ($1 \leq r \leq 10^4$, $r \leq m \leq 10^9$), que representan la cantidad de racimos y la cantidad de minutos que deben pasar antes de que lleguen el resto de los invitados.

La segunda línea contiene r enteros b_i ($1 \leq b_i \leq 10^9$), representando la cantidad de bananas del racimo i-ésimo.

Salida

La salida consta de una línea, la cantidad de bananas mínima que tiene que comer Coco por minuto para acabar todos los racimos antes de que lleguen los demás invitados.

Ejemplos

entrada estándar	salida estándar
4 8 7 5 10 3	4
5 5 21 29 10 17 5	29
5 6 21 29 10 17 5	21

C. Ciberataque

Autor: Fernando Sebastián Lomazzi

Ignacius Chantus es un reconocido ciberdelincuente de juegos en línea especialista en ataques de inyección de código y líder de la banda *Nachain Security Company*. En su última jugarreta, Ignacius, logró obtener acceso al sistema del juego *Conexiones*. En este juego, los participantes se turnan para colocar fichas en una línea de espacios contiguos. El objetivo es alinear una cantidad específica de fichas en fila para ganar. En cada turno, los jugadores pueden agregar una ficha en cualquier posición, no pudiendo colocarla en un espacio ocupado por otra ficha.

Ignacius ha conseguido la habilidad de colocar hasta p fichas en un solo turno, y ese turno ha llegado, enfrentándolo a una decisión estratégica: si no logra alinear k fichas en este movimiento, será descubierto por sus oponentes.

Tu tarea es ayudar a Ignacius a determinar si hay alguna forma de asegurar su victoria en este turno, dados el estado actual del tablero, la cantidad de fichas necesarias para ganar, y la cantidad de fichas que puede agregar.

Entrada

La entrada consta de dos líneas.

La primer línea contiene tres enteros n, k y p ($1 \leq n \leq 10^5$, $1 \leq k \leq n$, $0 \leq p \leq n$), que representan la cantidad de espacios para colocar fichas, la cantidad de fichas consecutivas necesarias para ganar y la cantidad de fichas que puede agregar Ignacius este turno.

La segunda línea contiene n valores f_i ($0 \leq f_i \leq 1$), representando si existe o no una ficha en la posición i-ésima.

Salida

La salida consta de una línea, "SI" si Ignacius puede ganar en este turno o "NO" en caso contrario.

Ejemplos

entrada estándar	salida estándar
10 5 3 1000010010	SI
4 4 2 0100	NO

D. Degradé

Autor: Lucio Rivera

Píxel Perfect es un talentoso diseñador gráfico que trabaja para una innombrable firma de ciberseguridad. En un día convencional de su trabajo, Píxel utiliza *FotoTienda*, una herramienta que le permite editar imágenes preexistentes, así como crear sus propios diseños. La empresa detrás del programa de edición es *Adobo*, quien es reconocida mundialmente por desarrollar programas que llevan al límite los recursos computacionales de sus usuarios. En FotoTienda, una imagen es procesada como una matriz de enteros, donde el elemento en fila i y la columna j corresponde al color del píxel en esa posición.

Una de las funcionales preferidas de Píxel es la generación de encantadores degradados verticales y horizontales. Para generar un degradé vertical, la herramienta comienza pintando todos los píxeles de la primera fila de un color al azar. Seguidamente, pinta todos los píxeles de la segunda fila de un color distinto al anterior, generando así que todas las filas estén pintadas del mismo color y que el color de una fila i sea necesariamente distinto al color de la fila $i + 1$. La herramienta repite este proceso hasta pintar todas las filas. El proceso de generación de un degradé horizontal es análogo al anterior, pero recorre columnas en lugar de filas. Nótese que este último proceso genera que todas las columnas estén pintadas del mismo color y que el color de una columna j sea necesariamente distinto al color de la columna $j + 1$.

Píxel sospecha que su versión de FotoTienda no está funcionando correctamente, así que, ni lento ni perezoso, le ha propuesto a los organizadores de la *Tecnomate* una brillante idea: aprovechar las brillantes mentes allí reunidas para que escriban un programa que le permita determinar si una imagen es el resultado del proceso de generación de un degradé vertical, uno horizontal, o no es un degradé correctamente generado. De esta forma Píxel podrá confirmar o descartar su hipótesis.

Entrada

La entrada consta de múltiples líneas.

La primera línea contiene dos enteros n, m ($2 \leq n, m \leq 1000$), el alto y el ancho en píxeles de la imagen.

Las siguientes n líneas contienen m enteros p_{ij} ($0 \leq p_{ij} \leq 1000$), representando el color del píxel ubicado en la intersección entre la fila i y la columna j.

Salida

La salida consta de una única línea, la frase "VERTICAL", si la imagen es el resultado de un degradé vertical; la frase "HORIZONTAL", si es el resultado de un degradé horizontal y "NADA" en caso contrario. Se puede demostrar que una imagen no puede ser un degrade vertical y horizontal a la vez.

Ejemplos

entrada estándar	salida estándar
4 4 1 2 1 3 1 2 1 3 1 2 1 3 1 2 1 3	HORIZONTAL
4 4 8 8 8 8 1 1 1 1 0 0 0 0 0 0 0 0	NADA

E. Abuela tecnológica

Autor: Fernando Sebastián Lomazzi

Marta es una abuela a la que le encanta tejer y salir con sus amigas al bingo. El mes pasado, su nieto Sebastián le regaló un nuevo y revolucionario dispositivo: un celular. Como ella no está acostumbrada a la tecnología, envía mensajes a sus amigas sosteniendo el celular con una mano, y pulsando letra a letra con su dedo índice. Su nieto, al observar esto, decidió explicarle la ineficiencia de no utilizar ambos pulgares. Como buen ingeniero en sistemas, se propuso obtener una métrica representativa del nivel de ineficiencia de su abuelita.

Sebastián notó que Marta escribe moviendo su dedo índice de una tecla a la siguiente de manera contigua, desplazándose solo en dirección horizontal o vertical hasta llegar a la siguiente letra. Por ejemplo, al escribir la palabra "hola", siempre comienza en la tecla en la esquina superior izquierda del teclado. Desde allí, mueve su dedo hasta la letra "h", la presiona, y luego repite el proceso desde la "h" para llegar a la siguiente letra, continuando de esta forma hasta completar toda la palabra.

Sebastián quiere ayudar a todos los nietos del mundo, por lo que deberás ayudarlo a él primero. Dada una disposición de teclado y una palabra que escribirá Marta comenzando desde la tecla en la esquina superior izquierda, deberás imprimir la cantidad total de teclas que Marta pasará por encima con su dedo índice para completar el mensaje, es decir, la cantidad total de movimientos entre teclas adyacentes que serán necesarios para escribir la palabra completa.

Entrada

La entrada consta de múltiples líneas.

La primer línea contiene dos enteros n y m ($1 \leq n,\ m \leq 26$), la cantidad de filas y columnas del teclado.

Las siguientes n líneas contienen m letras en minúsculas del abecedario **distintas**, representando las teclas del teclado.

La siguiente línea contiene un entero q ($1 \leq q \leq 100$), representando la cantidad de palabras que escribió su abuela.

Las siguientes $2 \cdot q$ líneas contienen un entero z ($1 \leq z \leq 100$) y una palabra s, representando el tamaño de la palabra y la palabra en minúsculas que escribió su abuela.

Se garantiza que $1 \leq n \cdot m \leq 26$ para todos los casos de prueba.

Salida

Por cada palabra, deberá imprimir la cantidad total de teclas que Marta pasará por encima con su dedo índice para completar el mensaje, si comienza desde la tecla en la esquina superior izquierda del teclado.

Ejemplos

entrada estándar	salida estándar
3 8 q w e r t y u i a s d f g h j k o l c v b n m p 1 4 gato	20
2 3 t e m a n s 4 5 menta 4 tema 5 antes 2 te	7 5 7 1

F. Fuga de memoria

Autor: Lucio Rivera

Fernando es un programador muy famoso en su ciudad. La velocidad a la que este prodigioso joven programa es motivo de envidia para todos sus colegas, quienes tiemblan de miedo cada vez que lo ven acercarse —con teclado RGB y mouse óptico en mano— a las competencias de programación que se organizan en su ciudad. Si bien Fernando es un programador extremadamente veloz, la legibilidad de su código y la prevención de bugs no son aspectos que se le den muy bien.

En su tiempo libre, Fernando suele inspeccionar con especial orgullo el código que escribe en su trabajo. En una de estas inspecciones, se percató de que uno de sus programas tenía un peligroso bug: Fernando quería acceder k veces a la posición i-ésima de un vector indexado en cero; pero, incorrectamente, termina evaluando la expresión $v_1[\, v_2[\cdots v_{k-1}[\, v_k[i]]]]$. Este no es un error que pueda tomarse a la ligera, pues enviar el código a producción ocasionaría que todo el área de informática de su empresa tenga que capacitarse, asistiendo a cursos de ciberseguridad todos los lunes por la mañana.

Mientras Fernando estaba haciendo la edición anterior de *Tecnomate*, descubrió un teorema que revolucionó el mundo del álgebra lineal y lo llevo a convertirse en multimillonario. Por este motivo, ahora se encuentra vacacionando en *Miami*, lo que le impide puede encomendar su error. Fernando le solicita a usted, a cambio de varios millones de dólares, que escriba un programa idéntico al defectuoso que codificó; pero que, dado un vector de entrada, advierta a sus colegas cuando se produzca una fuga de memoria.

Entrada

La entrada consta de múltiples líneas.

La primera línea contiene un entero n ($1 \leq n \leq 10^6$), que representa el tamaño del vector.

La siguiente línea contiene n números enteros v_i ($0 \leq v_i \leq 10^9$), representando el elemento en la posición i del vector.

La tercera línea contiene un entero q ($1 \leq q \leq 100$), representando la cantidad de consultas a responder. Cada consulta consiste en dos enteros k e i ($1 \leq k \leq 10^4$, $0 \leq i \leq 10^9$), la cantidad de veces y el índice inicial en el que el vector es evaluado, respectivamente.

Salida

Para cada consulta, deberá imprimir el resultado de la expresión descrita, si para su evaluación no se requiere acceder a una posición inválida del vector. Caso contrario deberá impirmir la frase *"Mi primera fuga de memoria"*.

Ejemplo

entrada estándar	salida estándar
6 2 0 4 3 5 7 4 1 5 3 1 2 3 5 0	7 4 3 Mi primera fuga de memoria

G. Cuenta Impagable

Autor: Lucio Rivera

El infame *BurpChein*, CEO de una compañía de ciberseguridad, ha cometido un grave error. En un acto de total imprudencia, BurpChein adquirió una licencia de software para su empresa en nombre de la universidad, que ahora debe pagar el costo de este imperdonable y, no precisamente barato, descuido.

Desafortunadamente para BurpChein, *el departamento de sistemas*, hará valer los 400 dólares que debió pagar por la de licencia y ha decidido someter al vil estudiante a un juego denominado *Secuencia 1 a 9*. El departamento determinó que, si BurpChein logra vencer a todos los profesores de sistemas en este juego, se le perdonará su error y podrá continuar estudiando en la universidad.

En una partida del juego, saldrá victorioso el jugador que, a partir de un vector de números enteros positivos indexados en cero, logre encontrar la secuencia de operaciones de menor costo total que transforme el vector dado en el vector [1, 2 , ... , 8, 9]. Las operaciones permitidas son las siguientes:

1. *Reemplazar*, el número v_i en la posición i por $v_i \ mod \ 10^{\phi(v_i)-k}$. Donde la función $\phi(x)$ es la cantidad de dígitos de un número entero positivo x y $0 \leq k < \phi(v_i)$. Costo: k.
2. *Eliminar*, el último dígito del número v_i en la posición i siempre y cuando $\phi(v_i) > 1$. Costo: 1.
3. *Eliminar*, el número v_i en la posición i. Costo i.
4. *Intercambiar*, el número v_i en la posición i con el número v_j en la posición j. Costo: 0.
5. *Copiar*, el número v_i en la posición i siempre y cuando v_i no haya sido alterado anteriormente por las operaciones 1 o 2. La copia es insertada al final del vector. Costo: i.

BurpChein sabe que, por algún motivo que aún desconoce, muchos profesores son imbatibles en este juego. Aterrorizado por la posibilidad de ser expulsado de la universidad, le solicita a usted que escriba un programa que le permita encontrar el costo total mínimo posible para asegurarse de siempre salir victorioso en el desafío. BurpChein le promete un cargo en su compañía si logra ayudarlo, aunque le advertimos que quizás este sea un motivo más para no ayudarlo.

Entrada

La entrada consta de múltiples líneas.

La primera línea contiene un entero n ($2 \leq n \leq 10^5$), que representa el tamaño del vector inicial.

La segunda línea contiene n números enteros v_i ($1 \leq v_i \leq 10^5$), representando el número en la posición i del vector.

Se garantiza que todos los dígitos de 1 a 9 aparecen como mínimo una vez en algún número. Se puede demostrar que siempre existe una secuencia de operaciones ganadora.

Salida

La salida consta de una única línea, el costo mínimo posible para ganar la partida.

Ejemplos

entrada estándar	salida estándar
9 1 2 3 4 5 6 7 8 9	0
4 12 334 675 8889	18

H. Derrape

Autor: Fernando Sebastián Lomazzi

Te diriges hacia Mar del Plata para disfrutar de unas vacaciones en familia. Todo va bien, comienzan su viaje con entusiasmo cantando una de sus canciones favoritas:

Vamos de paseo, pi pi pi,

En un auto feo, pi pi pi,

Pero no me importa, pi pi pi,

Porque llevo torta, pi pi ¡ruuum!

De repente, un pequeño carpincho cruza el camino y, al realizar una maniobra para evitarlo, el auto derrapa perdiendo el control. Te despiertas sin saber dónde se encuentran pero, afortunadamente, tu celular guardó las posiciones GPS durante el derrape.

Tienes disponibles tres ubicaciones: dos que indican tu posición antes del derrape y una que muestra tu posición actual, después de perder el control del vehículo. Sabes que las dos primeras ubicaciones corresponden a una trayectoria recta en dirección a Mar del Plata. Con esta información, tu objetivo es determinar si el derrape te desvió hacia la izquierda, continuó en línea recta o te llevó hacia la derecha en relación con la ruta que seguías.

Entrada

La entrada consta de múltiples líneas.

La primer línea contiene un entero t ($1 \leq t \leq 10^5$), la cantidad de casos de prueba.

Cada caso contiene seis enteros x_1, y_1, x_2, y_2, x_3 y y_3 ($-10^3 \leq x_1, y_1, x_2, y_2, x_3, y_3 \leq 10^3$), las ubicaciones ordenadas en forma temporal guardadas en tu celular.

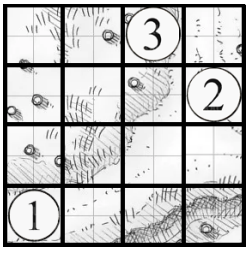
Salida

Por cada caso de prueba, deberá imprimir “IZQUIERDA”, “RECTO” o “DERECHA”, la dirección hacia la que ocurrió el derrape.

Ejemplos

entrada estándar	salida estándar
2 0 0 3 2 6 1 2 1 4 9 3 5	DERECHA RECTO
3 0 0 3 2 2 3 2 2 4 0 3 0 4 0 2 2 5 4	IZQUIERDA DERECHA DERECHA

Nota



Ejemplo del primer derrape del segundo caso de prueba

I. Batalla Continuada

Autor: Lucio Rivera

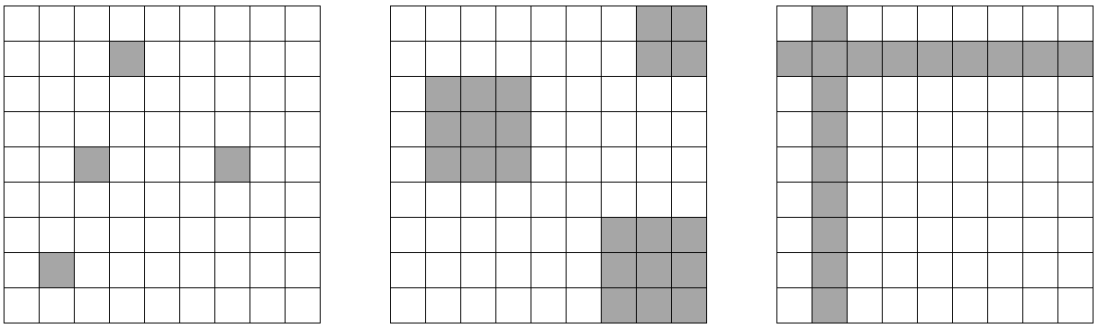
La épica batalla apocalíptica entre patos y burbujas continúa en esta edición de la *TecnoMate*. Desafortunadamente, en la edición anterior de la competencia, ningún humano pudo ayudar al patógrafo *Pating Turing* a desencriptar el *Código Misterio*. Esta situación obligó al reconocido experto a pedirle ayuda al novedoso chatbot *DuckGPT*, desarrollado por sus *compatotas*, para resolver el problema.

En una maniobra de increíble creatividad, las audaces burbujas lograron acceder al código fuente de DuckGPT y lo alteraron para que los patos no puedan usar la herramienta a su favor. Afortunadamente, el chatbot no fue eliminado en su totalidad y aún conserva algunos recuerdos de su pasado.

Como resultado de una alucinación, DuckGPT accede a ayudar a los patos, pero solo si las aves logran vencerlo en su juego favorito. Este juego comienza con el chatbot creando una matriz cuadrada, donde el elemento en la fila i y la columna j puede ser una mina explosiva mortal o, en su defecto, agua completamente inofensiva. La idea del juego, es que los oponentes no sepan dónde se encuentran las minas.

El objetivo de los patos es destruir, sin conocer su ubicación exacta, todas las minas de la matriz. DuckGPT permite a sus rivales realizar tres tipos de ataques:

- *Ataque 1*: Destruye la mina en la fila i y la columna j.
- *Ataque 2*: Destruye todas las minas dentro de un cuadrado de 3x3 cuyo centro está ubicado en la fila i y la columna j.
- *Ataque 3*: Destruye todas las minas en la fila i y la columna j.



Los tipos de ataques descritos. En la primera imagen, se visualizan múltiples ataques de tipo 1; en la segunda imagen, se visualizan múltiples ataques de tipo 2; y en tercera imagen se visualiza un único ataque de tipo 3.

Los patos le han concedido una última oportunidad de ser liberado; para lograrlo, usted deberá escribir un programa que, dada una matriz inicial y los ataques realizados, determine la cantidad de minas que quedan sin ser destruidas. Con su invención, los patos pretenden practicar para su enfrentamiento con el chatbot.

Entrada

La entrada consta de múltiples líneas.

La primera línea contiene un entero n ($1 \leq n \leq 1000$), que representa el tamaño de la matriz creada por el chatbot.

Las siguientes n líneas contienen n caracteres mayúsculos c_{ij} ($c_{ij} \in$ {"M","A"}), representando el tipo de elemento ubicado en la intersección entre la fila i y la columna j, donde el carácter "M" indica que el elemento es una mina y el carácter "A" indica que es agua. La siguiente línea contiene un entero q ($1 \leq q \leq 100$), que representa la cantidad de ataques realizados.

Cada ataque consiste en tres enteros a, i y j ($1 \leq a \leq 3$, $1 \leq i, j \leq n$), indicando el tipo de ataque y su ubicación.

Salida

La salida consta de una línea, la cantidad de minas que permanecen sin ser destruidas después de los ataques.

Ejemplos

entrada estándar	salida estándar
6 M M M M A M M M M A A M M M M A A M A A A M A M A A A A A M M M M M M M 3 1 1 4 2 2 2 3 6 6	1
9 A M A A A A A M M M M M M M M M M M A M M M A A A A A A M M M A A A A A A M M M A A M A A A M A A A A A A A A M A A A A M M M A M A A A A M M M A M A A A A M M M 7 1 8 2 1 5 3 1 2 4 2 4 3 2 1 9 2 8 8 3 2 2	1

J. Té extraño

Autor: Lucio Rivera

La prestigiosa asociación *Hot Tea and Machine Learning* (mejor conocida como HTML), es la entidad encargada de regular y controlar, con especial dedicación y tenacidad, todo té preparado por programadores alrededor del mundo. Recientemente, esta asociación ha decidido estandarizar las preparaciones de la infusión a nivel global.

Según el estricto ente, para que una bebida sea considerada té, las hojas de la planta *Camellia Sinensis* deben ser tostadas a temperaturas específicas. Dependiendo de este factor, una infusión de hojas con agua caliente puede clasificarse como té blanco, verde, amarillo, oolong, negro o, directamente, no ser considerada té.

	Blanco	Verde	Amarillo	Oolong	Negro
Temperatura (C°)	65-75	76-80	81-85	86-95	96-110

Tipos de té según la temperatura a la que son tostadas las hojas.

Adicionalmente, todo té puede ser enriquecido con aromatizantes. Los únicos ingredientes permitidos por HTML para los aromatizantes son bergamota, lavanda, canela, anís y jengibre. Algunas combinaciones de estos ingredientes han sido oficialmente nombradas por la asociación, que exige su correcta identificación. Según las combinaciones de ingredientes, un aromatizante puede ser Gingergrey, Karma, Police, Golang o no ser una combinación estandarizada.

	Bergamota	Lavanda	Canela	Anís	Jengibre
Gingergrey	✓	✓	✓		✓
Karma	✓	✓	✓	✓	✓
Police		✓		✓	
Golang	✓		✓	✓	

Tipos de aromatizantes estandarizados según los ingredientes que lo componen.

Se le solicita que escriba un programa que permita a HTML determinar la nomenclatura correcta de una infusión.

Entrada

La entrada consta de múltiples líneas.

La primera línea contiene un entero T ($40 \leq T \leq 130$), que representa la temperatura a la que las hojas son tostadas.

Las siguientes cinco líneas contienen una cadena de caracteres mayúsculos s_i ($s_i \in \{$"SI", "NO"$\}$), indicando si el aromatizante utilizado contiene o no al ingrediente i-ésimo y donde $i = 1$ corresponde a bergamota, $i = 2$ a lavanda, $i = 3$ a canela, $i = 4$ a anís y $i = 5$ a jengibre.

Salida

La salida consta de una línea. Si la infusión es un té, la línea contiene la frase "Te", seguida del tipo de tostado y, en caso de ser necesario, el tipo de aromatizante estándar utilizado, separadas por espacios. Si la bebida no es considerada té, la asociación le exige que el programa imprima únicamente la frase "Te extrano".

Ejemplos

entrada estándar	salida estándar
40 SI SI SI SI SI	Te extrano
105 SI SI SI NO SI	Te Negro Gingergrey
75 NO NO SI NO NO	Te Blanco

K. Contraseña Olvidada

Autor: Lucio Rivera

Fideo, un flamante estudiante de primer año, dedica gran parte de sus días a resolver problemas de programación competitiva. A pesar de la notable mejoría que ha experimentado, su salud se ha deteriorado considerablemente, afectando directamente a su memoria. Horas antes del inicio de la *Tecnomate*, Fideo se percató de que olvidó las credenciales necesarias para acceder al juez en línea. Afortunadamente, Fideo es una persona extremadamente sistemática por lo que sus contraseñas siempre son la concatenación de tres cosas cualesquiera que le gusten mucho y le resulten fáciles de recordar.

Para recuperar su contraseña, Fideo decidió probar todas sus posibles contraseñas. Para esto, se concentró en pensar algunas de sus palabras favoritas. Luego, dividió y anotó estas palabras arbitrariamente en tres hojas, donde las palabras de la primera hoja eran los prefijos, las de la segunda eran las bases y las de la tercera eran los sufijos. Seguidamente, Fideo comenzó a listar en otra hoja todas las posibles contraseñas siguiendo un proceso bien definido: primero, toma el primer prefijo de la hoja de prefijos y lo combina con cada base, siguiendo el orden en que están escritas en la hoja. Para cada combinación de prefijo y base, agrega cada sufijo, también respetando su orden de escritura. Fideo repite este proceso con el segundo prefijo y así sucesivamente, hasta que todos los prefijos, bases y sufijos hayan sido listados en el orden en que fueron escritos.

Justo cuando parecía que Fideo estaba a punto de salvar su participación en la competencia, se percató de que en una de sus potenciales contraseñas contenía la palabra *AEED*. Esta revelación causó en Fideo un inmenso terror, quien recordó aquellos días pasados y provocó que ya no sea capaz de dirigir su mirada hacia las palabras que listó. La situación es tan grave que, independientemente de si la palabra aparece en mayúsculas, minúsculas o en una combinación de ambas, Fideo simplemente no soporta verla.

Se le solicita que escriba un programa que, dados los sufijos, las bases y los prefijos que Fideo anotó en cada hoja; liste, siguiendo el mismo proceso, todas las posibles contraseñas que no aterroricen a Fideo.

Entrada

La entrada consta de múltiples líneas.

Inicialmente, se ingresará la hoja de sufijos, después la de bases y finalmente la de prefijos.

La primera línea del ingreso de una hoja contiene un entero n ($1 \leq n \leq 100$), que representa la cantidad de palabras anotadas en ella.

Las siguientes n líneas contienen un entero $|s_i|$ ($1 \leq |s_i| \leq 100$), representando el tamaño de la i-ésima palabra anotada, y una cadena de caracteres s_i, representando la i-ésima palabra anotada, compuesta de letras mayúsculas y minúsculas del alfabeto inglés.

Salida

Si no hay ninguna contraseña que no aterrorice a Fideo, deberá imprimir la frase "SIN SOLUCION". Caso contrario, deberá imprimir cada posible contraseña en el orden que resultaría del proceso descrito.

Ejemplo

entrada estándar	salida estándar
2 5 Messi 8 CocaCola 2 3 Cae 4 Boca 2 3 Lol 6 Educar	MessiCaeLol MessiBocaLol MessiBocaEducar CocaColaCaeLol CocaColaBocaLol CocaColaBocaEducar

L. LOR

Autor: Fernando Sebastián Lomazzi

Lucio es una joven promesa de las Matemáticas. Entre sus logros ha resueltos problemas triviales como la hipótesis de Rayman, la conjetura del Pancake y NP versus P. Tantos logros no son suficientes para satisfacer al ambisioso joven, ya que su máxima aspiración es trascender en el mundo de las Matemáticas siendo titular del nombre de una operación Matemática.

Para lograr esto, Lucio definió la operación LOR (Lucio Operation Result) entre dos números enteros no negativos como:

Sean a y b dos enteros no negativos y sean a_{2i} y b_{2i} los bits i-ésimos de a y b en base 2, respectivamente. Entonces se define a la operación LOR(a, b) como el resultado de operar bit a bit a y b de la siguiente manera:

$$LOR(a,b)_{2i} = \begin{cases} 1 & \text{si } a_{2i} + b_{2i} = 1 \\ 0 & \text{si } a_{2i} + b_{2i} \neq 1 \end{cases}$$

Como Lucio es un gran matemático, pero no tan hábil en informática, necesita tu ayuda para probar las propiedades de su nueva operación. Para ello, dado un conjunto de números, deberás calcular el valor máximo de LOR obtenible de todos los subconjuntos posibles que contengan más de un elemento.

Entrada

La entrada consta de dos líneas.

La primer línea contiene un entero n ($2 \leq n \leq 20$), que representa la cantidad de elementos en el conjunto.

La segunda línea contiene n enteros m_i ($0 \leq m_i \leq 100$), representando los elementos del conjunto.

Salida

La salida consta de una línea, el valor máximo de LOR obtenible de todos los subconjuntos posibles que contengan más de un elemento del conjunto inicial.

Ejemplos

entrada estándar	salida estándar
4 5 3 7 6	7
3 2 2 4	6
7 57 39 18 11 79 6 40	127

Nota

Para el primer caso de prueba, el subconjunto con LOR máximo es $\{3, 5, 6, 7\}$. Para el segundo caso de prueba el subconjunto con LOR máximo es $\{2, 4\}$. Para el tercer caso de prueba el subconjunto con LOR máximo es $\{6, 39, 40, 57, 79\}$

M. Tirada de la Muerte

Autor: Fernando Sebastián Lomazzi

La Ciudadela de la Corona de Hielo se encuentra más gélida que nunca, ante ti y tu hermandad, se encuentra el Rey Exánime *Arthas Menethil*. Tras semanas, incluso meses de arduo trabajo finalmente tienen la oportunidad de cruzar miradas con el ex-paladín de la Mano de Plata. Ansiedad y nervios invaden al grupo, preparándolos mentalmente para la inminente batalla, no obstante, antes de enfrentar el desafío final, deciden tomarse un descanso para recobrar energías y fortalecer su determinación.

Mientras descansas, decides apostar un poco de tu oro con los miembros de tu hermandad, ya que esa montura mamut no se pagará sola. Decides jugar a la *Tirada de la Muerte*, un juego de dados que aprendiste con ellos.

La Tirada de la Muerte es un juego basado en dos jugadores, los cuales se turnan para arrojar un dado. El dado inicialmente tiene D caras y, a medida que se arroja el mismo, es reemplazado por uno con la misma cantidad de caras que el resultado obtenido. En el momento que uno de los jugadores obtenga un 1 como resultado de su tirada, pierde automáticamente el juego y debe entregar una cantidad de oro D a su rival.

Como necesitas el oro, pero tampoco eres ningún necio, decides obtener la probabilidad de ganar siendo el jugador inicial. ¿Podrás lograrlo antes de que el descanso termine?

Entrada

La entrada consta de una línea "/roll D" siendo $1 \leq D \leq 20$ la cantidad de caras iniciales del dado que se lanzará en la Tirada de la Muerte.

Salida

La salida consta de una línea, la probabilidad de ganar el juego siendo el jugador inicial. La probabilidad deberá imprimirse con exactamente seis cifras decimales.

Ejemplos

entrada estándar	salida estándar
/roll 3	0.416667
/roll 17	0.496732