

Torneo Argentino de Programación

27 de septiembre de 2014

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

Sesión de Competencia

Este conjunto contiene 11 problemas; las páginas están numeradas de 1 a 20.

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 — Al azar

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

Los juegos de cartas son muchos y muy variados, y su origen se remonta a tiempos ancestrales. A veces puede resultar sorprendente que sigan siendo capaces de proveernos entretenimiento después de siglos de ser jugados con las mismas reglas, pero entonces debemos comprender que cada partida es esencialmente distinta de todas las demás que se han jugado en la historia de la humanidad, dada la gran cantidad de posibles formas de ordenar las cartas antes de iniciarla. En efecto, pocos juegos resultan entretenidos si utilizamos las cartas siempre en el mismo orden, o si existe una correlación entre cartas sucesivas que nos permita predecir el orden en el que se encuentran. Esta es la razón por la cual se suele mezclar las cartas antes de empezar cada juego, y por esto mismo les pedimos ahora que hagan un programa para controlar que una secuencia de cartas ha sido bien mezclada.

Para simplificar el problema, vamos a concentrarnos solamente en las barajas de *cartas españolas*, que consisten en 48 cartas distintas. Cada carta está identificada por un *valor*, que es un número del 1 al 12, y por un *palo*, que puede ser “bastos”, “copas”, “espadas” u “oros”. Ahora bien, como no queremos simplificar excesivamente su tarea, vamos a tener en cuenta que no todos los juegos utilizan las 48 cartas de la baraja. Dada una secuencia de N cartas, decimos que está bien mezclada si no hay en ella dos cartas sucesivas que comparten el mismo valor o el mismo palo. Caso contrario, decimos que ha sido mal mezclada. ¿Pueden ayudarnos a decidir si una secuencia está bien mezclada?

Entrada

La primera línea contiene un entero N , que representa la cantidad de cartas que se utilizan en el juego que vamos a considerar ($2 \leq N \leq 48$). Cada una de las siguientes N líneas contiene la descripción de una carta de la secuencia que queremos analizar, dada por un entero V que representa su valor ($1 \leq V \leq 12$) y un carácter P que representa su palo: “b” para bastos; “c” para copas; “e” para espadas y “o” para oros. Todas las cartas dadas son distintas, y se dan en la entrada en el mismo orden en el que aparecen en la secuencia.

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si la secuencia de cartas dada en la entrada ha sido bien mezclada o no. El carácter debe ser “B” en caso de que esté bien mezclada y “M” en caso de que esté mal mezclada.

Entrada de ejemplo	Salida para la entrada de ejemplo
4	B
1 b	
2 c	
3 e	
4 o	

Entrada de ejemplo	Salida para la entrada de ejemplo
3 1 b 2 b 3 c	M

Entrada de ejemplo	Salida para la entrada de ejemplo
3 1 b 1 c 2 e	M

Entrada de ejemplo	Salida para la entrada de ejemplo
32 5 c 2 b 4 e 3 o 12 b 1 c 7 e 6 c 12 e 4 o 1 b 6 o 3 e 12 o 11 e 12 c 5 o 10 b 9 o 3 c 4 b 11 c 8 e 9 c 1 e 4 c 8 b 2 o 6 b 9 e 7 b 5 e	B

Problema B — Base-3 balanceada

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

A lo largo de la historia se han desarrollado sistemas de numeración muy diversos. Algunos, como el de números romanos, se han abandonado casi por completo en la actualidad por ser poco convenientes. Otros sistemas aún más exóticos son utilizados sólo para ciertas aplicaciones, como por ejemplo el sistema factorádico en el contexto de la numeración de permutaciones. En este problema vamos a considerar un sistema de numeración conocido como *base-3 balanceada*, que surge naturalmente en el análisis de diversos problemas matemáticos relacionados con las balanzas de platillos.

El sistema de base-3 balanceada es similar al sistema decimal o base-10 al que estamos acostumbrados en que es un sistema *posicional*. Los sistemas posicionales tienen *dígitos* cuyo orden relativo determina qué potencia de la *base* los acompaña. Por ejemplo, en base-10 tenemos

$$123 = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0.$$

En los sistemas posicionales estándar, los dígitos permitidos son todos los números del 0 al $B - 1$, donde B es la base del sistema utilizado. Entonces, el número 123 en base-10 queda escrito en base-3 estándar como “11120”, dado que

$$1 \times 3^4 + 1 \times 3^3 + 1 \times 3^2 + 2 \times 3^1 + 0 \times 3^0 = 123.$$

La base-3 balanceada difiere de la base-3 estándar solamente en el hecho de que los dígitos permitidos son el 0, el 1 y el -1 , que vamos a escribir como “0”, “+” y “-” respectivamente. De este modo, el número 123 en base-10 queda escrito como “+----0” en base-3 balanceada, porque

$$1 \times 3^5 + (-1) \times 3^4 + (-1) \times 3^3 + (-1) \times 3^2 + (-1) \times 3^1 + 0 \times 3^0 = 123.$$

Como la conversión de números en base-10 a base-3 balanceada es un proceso mecánico y algo tedioso, requerimos un programa que lo realice por nosotros. ¿Pueden ayudarnos?

Entrada

La entrada consiste en una única línea con un entero positivo N , el número en base-10 que deseamos escribir en base-3 balanceada ($1 \leq N \leq 1000$).

Salida

Imprimir en la salida una línea conteniendo una cadena formada únicamente por los caracteres “0”, “+” y “-” y que no empiece con “0”, que representa los dígitos del número N en base-3 balanceada. La restricción de no comenzar la secuencia con “0” asegura que la representación es única.

Entrada de ejemplo	Salida para la entrada de ejemplo
123	+----0

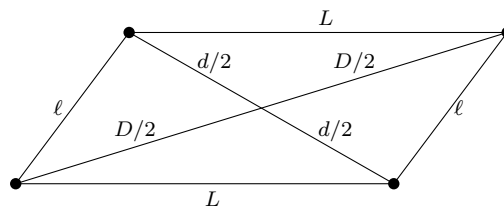
Entrada de ejemplo	Salida para la entrada de ejemplo
729	+000000

Problema C — Constelación del paralelogramo

AUTOR: PABLO A. HEIBER - UNIVERSIDAD DE BUENOS AIRES

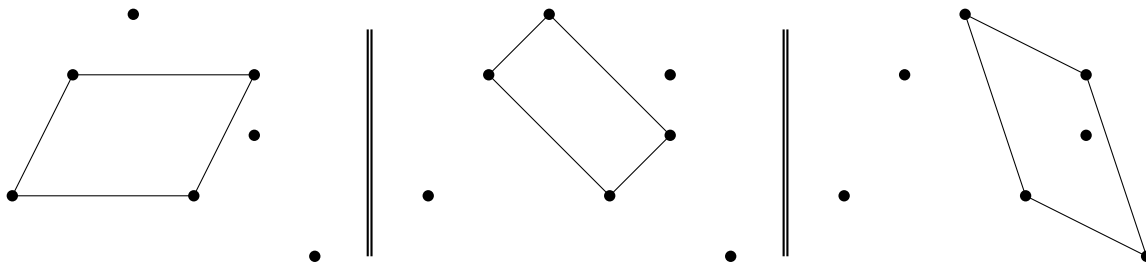
Cuando Ciro Samsa se despertó una mañana después de un sueño intranquilo, se encontró tendido sobre su cama, pero no convertido en un monstruoso insecto. Sin embargo, mirando el cielo aún estrellado por la ventana, sí se encontró con que la configuración visual del cosmos esa noche le estaba mostrando cuatro estrellas posicionadas como los vértices de un perfecto paralelogramo. Como buen amante de la geometría, llamó a esas cuatro estrellas “constelación del paralelogramo”.

Un paralelogramo es un polígono de cuatro lados con los pares de lados opuestos de idéntica longitud. Equivalentemente, un paralelogramo es un cuadrilátero convexo cuyas dos diagonales se cortan en sus puntos medios. La siguiente figura ilustra ambas definiciones, para un paralelogramo de lados de longitudes L y ℓ y diagonales de longitudes D y d .



Lamentablemente, los astrónomos aún no se ponen de acuerdo en cuáles eran las estrellas que estaba mirando Ciro. El problema es que en cada imagen que obtienen de una configuración de estrellas, suelen aparecer muchos conjuntos de cuatro de ellas que son los vértices de un paralelogramo.

Una imagen del cielo estrellado lo representa mediante un conjunto de puntos en el plano cartesiano, cada uno de los cuales corresponde a una estrella. La siguiente figura muestra tres copias de la misma imagen, cada una ilustrando uno de los tres conjuntos de cuatro estrellas que son vértices de un paralelogramo, marcado en la figura con líneas sólidas.



Su tarea es salvaguardar el buen nombre de los astrónomos calculando, dada una imagen, la cantidad de paralelogramos que contiene.

Entrada

La primera línea contiene un entero N que indica la cantidad de estrellas en la imagen a analizar ($4 \leq N \leq 1000$). Las siguientes N líneas describen cada una de las estrellas en la imagen mediante dos enteros X_i e Y_i , indicando respectivamente las coordenadas X e Y de la estrella en el plano cartesiano ($-10^8 \leq X_i, Y_i \leq 10^8$ para $i = 1, 2, \dots, N$). Pueden suponer que no hay en la imagen dos estrellas en la misma posición.

Salida

Imprimir en la salida una línea conteniendo un único entero que representa la cantidad de conjuntos de cuatro puntos de la entrada que son vértices de un paralelogramo.

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

Entrada de ejemplo	Salida para la entrada de ejemplo
11 0 0 0 1 0 -1 1 0 1 1 1 -1 -1 0 -1 1 -1 -1 0 2 1 2	44

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

Entrada de ejemplo	Salida para la entrada de ejemplo
6 -100000000 100000000 100000000 -100000000 -100000000 -100000000 100000000 100000000 1 1 -1 -1	2

Problema D – Dominó fractal

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

Una pregunta que podemos hacernos si tenemos mucho tiempo libre (o si somos matemáticos, que es prácticamente lo mismo) es la siguiente: dado un tablero cuadrado de $N \times N$, ¿de cuántas formas podemos cubrirlo con piezas de dominó de modo que las piezas no se superpongan entre sí y no queden casilleros sin cubrir? Las piezas de dominó son rectángulos compuestos por dos casilleros adyacentes, que pueden ser horizontales (1×2) o verticales (2×1). Por ejemplo, hay sólo dos formas de cubrir un tablero de 2×2 , pero 12.988.816 formas de cubrir un tablero de 8×8 .

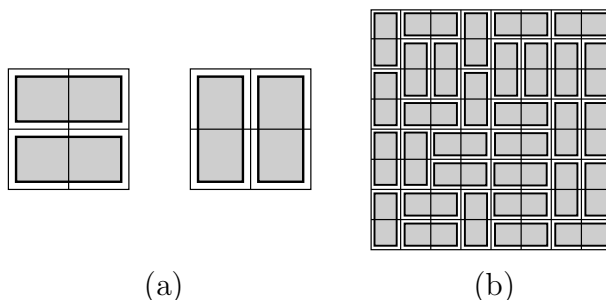


Figura 1: (a) Las dos formas de cubrir un tablero de 2×2 con piezas de dominó horizontales (izquierda) o verticales (derecha); (b) un posible cubrimiento de un tablero de 8×8 .

Para aquellos con aún más tiempo libre existe una variante de esta pregunta que consiste en considerar un tablero en el que algunos casilleros se consideran *ocupados*, de modo que debemos contar la cantidad de cubrimientos en los que las piezas de dominó cubren los casilleros no ocupados, sin superposiciones ni huecos. Por ejemplo, hay dos formas de cubrir un tablero de 3×3 en el que el casillero central está ocupado, y 75.272 formas de hacerlo en un tablero de 7×7 .

En este problema vamos a ocuparnos de una variante de esta variante de la pregunta original, pensada específicamente para aquellos de ustedes con exceso de tiempo libre. Nuevamente consideramos un tablero de $N \times N$ en el que algunos casilleros pueden estar ocupados, pero queremos ahora cubrirlo con *piezas de dominó fractales de orden K* .

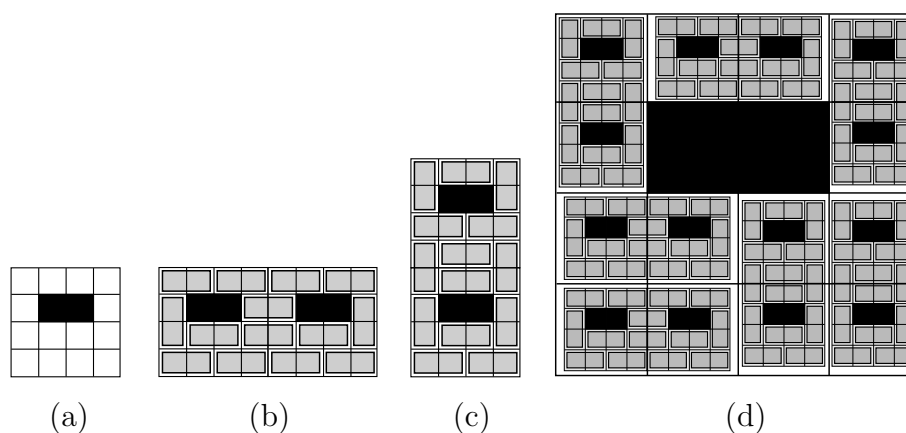


Figura 2: (a) Un tablero de 4×4 con dos casilleros ocupados; (b) una de las 89 piezas de dominó horizontales de orden uno para el tablero (a); (c) una de las 52 piezas de dominó verticales de orden uno para el tablero (a); (d) un cubrimiento del tablero (a) con las piezas de dominó fractales de orden uno de (b) y (c).

Dado un tablero de $N \times N$, una pieza de dominó fractal de orden $K = 0$ es una pieza de dominó usual, es decir un rectángulo de 1×2 ó de 2×1 casilleros. Una pieza de dominó fractal de orden $K > 0$ consiste en una pieza de dominó en la que cada casillero es una copia del tablero de $N \times N$ dado, y la pieza en su conjunto está cubierta por piezas de dominó fractales de orden $K - 1$.

Nótese que en general para $K > 1$ puede haber más de una pieza de dominó fractal de orden $K - 1$ de cada tipo (horizontal o vertical). En este caso, suponemos que al momento de utilizar una pieza ésta se elige con probabilidad uniforme entre todas las posibles piezas de dominó fractales del tipo y orden deseado. Del mismo modo, en caso de que exista más de un cubrimiento posible para un dado tablero o pieza de dominó fractal, vamos a suponer que todos ellos son igualmente probables.

Al cubrir un tablero con piezas de dominó fractales de orden K se utiliza una dada cantidad de piezas de orden K , una cantidad mayor de piezas de orden $K - 1$, una cantidad aún mayor de piezas de orden $K - 2$, y así siguiendo. Esto continúa hasta llegar a una cantidad posiblemente enorme de piezas de orden cero, es decir, piezas usuales sin nada adentro. Por ejemplo, se ve en la Figura 2d que el tablero cubierto con piezas de dominó fractales de orden uno contiene siete de esas piezas y 98 piezas de orden cero.

La tarea en este problema es calcular la proporción esperada de piezas de dominó fractales de orden cero (piezas usuales) que son verticales, si suponemos que todos los cubrimientos de un tablero dado con piezas de dominó fractales de un orden K dado son igualmente probables.

Entrada

La primera línea contiene dos enteros N y K , que representan el tamaño del tablero y el orden de las piezas de dominó fractales que queremos usar para cubrirlo, respectivamente ($2 \leq N \leq 8$ y $0 \leq K \leq 10^9$). Las siguientes N líneas contienen N enteros cada una, siendo el j -ésimo número en la i -ésima línea un 1 si el casillero en la fila i y la columna j del tablero está ocupado, y un 0 caso contrario. El tablero dado no está totalmente ocupado, y siempre es posible cubrirlo usando piezas de dominó.

Salida

Imprimir en la salida una línea conteniendo un racional que representa la proporción esperada de piezas de dominó fractales de orden cero que son verticales, cuando consideramos que todos los cubrimientos del tablero dado con piezas de dominó fractales de orden K son igualmente probables. Imprimir el resultado utilizando exactamente 5 dígitos luego del punto decimal, redondeando de ser necesario.

Entrada de ejemplo	Salida para la entrada de ejemplo
4 2 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0	0.50750

Problema E — Erdős et al.

AUTOR: PABLO A. HEIBER - UNIVERSIDAD DE BUENOS AIRES

Paul Erdős fue un matemático húngaro del siglo XX que alcanzó los más altos niveles de reconocimiento. Tal es así que es considerado un honor no solamente haber escrito un artículo con él, sino incluso haber compartido la autoría de un trabajo con uno de sus coautores. Más aún, escribir un artículo con alguien que escribió un artículo con alguien que escribió un artículo con Erdős es una aspiración importante de muchos jóvenes investigadores.

Una consecuencia natural de la mencionada asignación de honores, dada en el contexto de las ciencias formales, es el surgimiento de los *números de Erdős*. Erdős es el único con número de Erdős 0, y para cualquier otra persona p , su número de Erdős n queda definido por la secuencia de artículos a_1, \dots, a_n más corta tal que Erdős es uno de los autores de a_1 , p es uno de los autores de a_n , y entre cualquier par de artículos consecutivos de la secuencia a_i y a_{i+1} hay al menos un autor en común. Si no existe ninguna secuencia de artículos que vincule a Erdős con p de este modo, decimos que el número de Erdős de p está *indefinido*.

Su tarea en este problema es calcular el número de Erdős basados únicamente en los autores de los artículos de un corpus recibido como entrada. Lamentablemente, la ciencia actual requiere a los científicos y científicas aumentar muy rápido el número de artículos que escriben, lo que provocó que tanto la cantidad de artículos como la cantidad de autores por artículo sean enormes. Por supuesto, esta realidad es un escollo que una solución a este problema debe poder manejar.

Entrada

La primera línea contiene dos enteros A y N , donde A representa la cantidad de artículos del corpus a analizar y N la cantidad de personas que aparecen en el mismo (siendo $2 \leq A, N \leq 10^5$). Las personas se identifican con enteros entre 1 y N , siendo siempre Erdős la persona identificada por el número 1.

Las siguientes A líneas describen un artículo cada una. Cada una de estas líneas comienza con un entero C que representa la cantidad de autores del artículo ($2 \leq C \leq 10^5$), y tiene luego C enteros distintos P_1, P_2, \dots, P_C que representan los identificadores de los autores del artículo correspondiente ($1 \leq P_i \leq N$ para $i = 1, 2, \dots, C$). La suma de la cantidad de autores C de todos los artículos no supera 10^5 .

Salida

Imprimir en la salida tres enteros D , M y S , donde D debe representar la cantidad de personas del corpus que tienen un número de Erdős definido, M el máximo número de Erdős de una de esas personas y S la suma de todos los números de Erdős de las personas con dicho número definido.

Entrada de ejemplo	Salida para la entrada de ejemplo
3 5 2 1 5 3 5 2 3 3 3 2 4	5 3 8

Entrada de ejemplo	Salida para la entrada de ejemplo
5 11 2 10 11 4 1 3 5 7 6 2 3 4 5 6 7 4 3 5 7 9 3 8 1 5	9 2 12

Entrada de ejemplo	Salida para la entrada de ejemplo
6 31 9 1 2 3 15 20 25 30 9 10 10 2 25 7 9 3 11 12 13 14 4 10 11 12 13 14 4 16 17 18 19 5 2 5 7 9 21 22 23 24 26 27 28 29 7 3 22 6 21	29 4 63

Problema F — Fertilización asistida de cadenas

AUTOR: MARIO YNOCENTE CASTRO

Las cadenas son como las plantas, para crecer precisan de mucho amor y cuidado. En este problema vamos a seguir la evolución de un jardín con N cadenas durante T temporadas. Las cadenas del jardín están numeradas del 1 al N , e inicialmente están todas vacías. Cada temporada realizamos dos actividades distintas en nuestro jardín, a saber:

- Al comenzar la temporada, podemos *podar* el jardín eliminando los C caracteres de la derecha de cada una de las N cadenas del jardín.
- Luego de realizada la poda, *fertilizamos* el jardín de modo que cada una de las N cadenas crece, agregándosele un carácter (posiblemente distinto para cada cadena) a la derecha.

Al final de cada temporada un buen jardinero de cadenas siempre se toma un momento para contemplar su obra. Para ello, elegimos un número P del 1 al N y nos detenemos a apreciar la belleza de la cadena que queda en la posición P cuando ordenamos las N cadenas del jardín alfabéticamente de menor a mayor (desempatando arbitrariamente poniendo primero las cadenas identificadas por números más chicos).

Estos momentos de contemplación deberían ser un merecido descanso para el jardinero, por lo que no queremos perder el tiempo ordenando las cadenas del jardín para identificar aquella que deseamos observar. ¿Pueden ayudarnos a encontrarla?

Entrada

La primera línea contiene dos enteros N y T , que representan la cantidad de cadenas del jardín y la cantidad de temporadas que seguimos su evolución, respectivamente (con $2 \leq N \leq 100$ y $1 \leq T \leq 10^4$). Las siguientes T líneas describen una temporada cada una, en el orden en el que se van sucediendo.

La descripción de cada temporada consiste de un número entero C , una cadena S y otro número entero P ($1 \leq P \leq N$). El número C es no negativo, y representa la cantidad de caracteres que se eliminan durante el período de poda al comenzar la temporada. C siempre será menor o igual que la longitud de las cadenas al comenzar la temporada correspondiente, y puede ser nulo en caso de que en dicha temporada no se poda el jardín. La cadena S contiene exactamente N caracteres s_1, s_2, \dots, s_N , siendo el i -ésimo carácter s_i el que debe agregarse a la derecha de la cadena identificada por el número i (s_i es una letra minúscula del alfabeto inglés para $i = 1, 2, \dots, N$). Finalmente, el entero P representa la posición de la cadena que deseamos contemplar al terminar la temporada, cuando ordenamos las N cadenas del jardín como indica el enunciado.

Salida

Imprimir en la salida T líneas, una por cada temporada que se describe en la entrada. La i -ésima línea debe contener el número que identifica a la cadena que deseamos observar al finalizar la i -ésima temporada, para $i = 1, 2, \dots, T$.

Entrada de ejemplo	Salida para la entrada de ejemplo
2 4	1
0 aa 1	2
0 ba 1	1
1 ba 2	2
2 aa 2	

Entrada de ejemplo	Salida para la entrada de ejemplo
26 26	1
0 abcdefghijklmnopqrstuvwxyz 1	1
1 bcdefghijklmnopqrstuvwxyz 2	1
1 cdefghijklmnopqrstuvwxyzab 3	1
1 defghijklmnopqrstuvwxyzabc 4	1
1 efghijklmnopqrstuvwxyzabcd 5	1
1 fghijklmnopqrstuvwxyzabcde 6	1
1 ghijklmnopqrstuvwxyzabcdef 7	1
1 hijklmnopqrstuvwxyzabcdefg 8	1
1 ijklmnopqrstuvwxyzabcdefgh 9	1
1 jklmnopqrstuvwxyzabcdefghi 10	1
1 klmnopqrstuvwxyzabcdefghij 11	1
1 lmnopqrstuvwxyzabcdefghijk 12	1
1 mnopqrstuvwxyzabcdefghijkl 13	1
1 nopqrstuvwxyzabcdefghijklm 14	1
1 opqrstuvwxyzabcdefghijklmn 15	1
1 pqrstuvwxyzabcdefghijklmno 16	1
1 qrstuvwxyzabcdefghijklmnop 17	1
1 rstuvwxyzabcdefghijklmnopq 18	1
1 stuvwxyzabcdefghijklmnopqr 19	1
1 tuvxyzabcdefghijklmnopqrs 20	1
1 uvwxyzabcdefghijklmnopqrst 21	1
1 vwxyzabcdefghijklmnopqrstu 22	1
1 wxyzabcdefghijklmnopqrstuv 23	1
1 xyzabcdefghijklmnopqrstuvw 24	1
1 yzabcdefghijklmnopqrstuvwx 25	1
1 zabcdefghijklmnopqrstuvwxy 26	1

Problema G — Ganar, gustar, golear y galantear

AUTOR: LEOPOLDO TARAVILSE - UNIVERSIDAD DE BUENOS AIRES

Germán y Gianina se reúnen todos los fines de semana a jugar al fútbol con sus amigos. Este sábado, tanto Germán como Gianina están lesionados, por lo que van a formar parte del partido como directores técnicos de los dos equipos.

Para elegir los equipos van a utilizar el siguiente mecanismo: de los N amigos de Germán y Gianina van a elegir $N/2$ jugadores cada uno (N es par), siendo el proceso de selección mediante $N/2$ turnos. En cada turno, cada uno elige un jugador distinto de los que aún no tienen equipo asignado, y se lo asigna a su equipo. Germán ganó el tradicional pan y queso, así que puede elegir primero en todos los turnos. Sin embargo, para hacerse el galante, Germán puede decidir en cada turno dejar que Gianina elija primero en ese turno.

Por ejemplo, si $N = 6$ hay $N/2 = 3$ turnos. Supongamos que Germán decide ceder el lugar a Gianina en el segundo y el tercer turno, pero no en el primero. En ese caso, el orden de selección de jugadores sería:

$$\underbrace{\text{Germán} - \text{Gianina}}_{\text{turno 1}} - \underbrace{\text{Gianina} - \text{Germán}}_{\text{turno 2}} - \underbrace{\text{Gianina} - \text{Germán}}_{\text{turno 3}}.$$

Como todos sabemos, el fútbol es un deporte que se rige por la lógica. Todos los amigos de Germán y Gianina tienen un puntaje que indica qué tan bien juegan al fútbol. Si la suma de los puntajes del equipo de Germán es mayor a la suma de los puntajes del equipo de Gianina, entonces gana el equipo de Germán. Si la suma de puntajes es igual para ambos, el partido termina en empate. En cualquier otro caso, gana el equipo de Gianina.

Germán quiere hacerse el galante, pero más aún le interesa ganar el partido. Él quiere ceder su turno la mayor cantidad de veces posible de modo tal que de todas formas pueda ganar el partido. Gianina también quiere ganar el partido, por lo tanto siempre que les toca elegir un jugador disponible, ambos eligen de forma de maximizar sus posibilidades.

Entrada

La primera línea contiene un entero N con la cantidad de jugadores para elegir (siendo $2 \leq N \leq 1000$ con N par). La segunda línea contiene N enteros P_1, P_2, \dots, P_N indicando los puntajes de cada jugador ($1 \leq P_i \leq 1000$ para $i = 1, 2, \dots, N$).

Salida

Imprimir en la salida una línea conteniendo un único entero que representa cuántas veces Germán puede ceder el turno de modo tal de asegurarse de que su equipo va a ganar el partido. En caso de que Germán no pueda asegurarse la victoria, se deberá imprimir el entero -1 .

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

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

Entrada de ejemplo	Salida para la entrada de ejemplo
4 60 95 100 65	0

Entrada de ejemplo	Salida para la entrada de ejemplo
10 10 10 10 10 10 10 10 10 10 10	-1

Problema H – Hora pico

AUTOR: NICOLÁS ÁLVAREZ - UNIVERSIDAD NACIONAL DEL SUR

Nlogonia es una ciudad muy organizada, en la que las viviendas de sus habitantes están todas ubicadas en el extremo Este de la ciudad y sus lugares de trabajo están ubicados en el extremo Oeste.

En cada jornada laboral, los habitantes se dirigen a sus lugares de trabajo en el Oeste y al terminar el día retornan a sus hogares en el Este. Para transportarse dependen del sistema ferroviario de Nlogonia.

La compañía de trenes ofrece N servicios distintos. Cada uno de ellos hace un recorrido que une una estación en el Oeste con una estación en el Este. Existen exactamente N estaciones en el Oeste y N estaciones en el Este. En ambos extremos las estaciones están numeradas de 1 a N siguiendo el orden Norte-Sur. La estación más al Norte se identifica con el número 1 y la más al Sur con el número N . Cada estación de ambos extremos pertenece exactamente a un servicio.

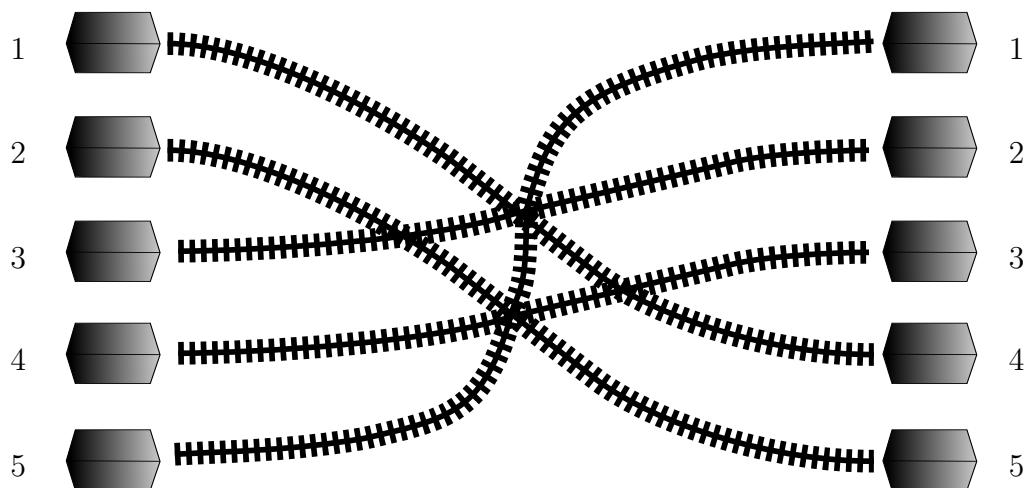


Figura 1: Situación con $N = 5$ estaciones que corresponde a la primera entrada de ejemplo.

Cuando finaliza la jornada laboral, los Nlogoninos salen de sus lugares de trabajo, ansiosos por volver a sus casas. Esto da lugar a un intenso tráfico en el sistema ferroviario conocido como “tráfico de hora pico”.

Como se puede apreciar en la figura anterior, algunos servicios de tren se cruzan en sus recorridos. Los recorridos de dos servicios de tren se cruzan si y sólo si el orden relativo de los trenes en la dirección Norte-Sur es distinto al salir de la estación del Oeste que al llegar a la estación del Este.

Para evitar accidentes, la compañía de trenes de Nlogonia decidió planificar la salida de los trenes en turnos de manera que en cada turno no salgan dos trenes cuyos recorridos se crucen. Para que esto no genere demoras, desean planificar los turnos de modo tal que la cantidad total de turnos sea la mínima posible. Por suerte, en la hora pico sale exactamente un tren de cada servicio.

Siguiendo con el ejemplo de la figura, el servicio que parte de la estación 5 se cruza con todos los restantes y por lo tanto deberá planificarse un turno exclusivamente para su salida. Los otros cuatro servicios no pueden salir en un único turno ya que existen cruces entre ellos. Sin embargo, es posible agruparlos en dos turnos, uno para los trenes que salen de las estaciones 1 y 2, y otro para los trenes que salen de las estaciones 3 y 4. De este modo, hace falta planificar un total de 3 turnos para evitar todo riesgo de colisiones.

Ustedes serán los encargados de esta tarea: dadas las descripciones de los N servicios de tren de Nlogonia, deben determinar cuál es la mínima cantidad de turnos en los que se pueden planificar las salidas en la hora pico evitando todo riesgo de accidentes.

Entrada

La primera línea contiene un entero N indicando la cantidad de servicios de tren en Nlogonia ($1 \leq N \leq 10^5$). La segunda línea contiene N enteros distintos E_1, E_2, \dots, E_N ($1 \leq E_i \leq N$), indicando que el tren que sale de la estación i en el Oeste debe llegar a la estación E_i en el Este, para $i = 1, 2, \dots, N$.

Salida

Imprimir en la salida una línea conteniendo un único entero que representa la cantidad mínima de turnos en la que se pueden planificar los servicios.

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

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

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

Problema I – Intervalos engrapados

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

Dos números naturales n y m son coprimos si su máximo divisor común es el número 1. En otras palabras, n y m son coprimos si no existe ningún entero $d > 1$ tal que d divide exactamente tanto a n como a m . Un conjunto finito de dos o más números naturales consecutivos es un “intervalo engrapado” si no hay en él ningún número que sea coprimo con todos los demás números del conjunto.

Dado un rango $[A, B]$, queremos contar la cantidad de intervalos engrapados totalmente contenidos en él. Esto es, queremos saber cuántos pares (a, b) distintos existen tales que $A \leq a < b \leq B$ y el conjunto $\{a, a + 1, \dots, b\}$ es un intervalo engrapado.

Entrada

La primera línea contiene un entero P que indica la cantidad de preguntas que deben responder ($1 \leq P \leq 1000$). Cada una de las P líneas siguientes describe una pregunta, y contiene dos enteros A y B que representan los extremos del rango $[A, B]$ dentro del cual queremos contar intervalos engrapados ($1 \leq A \leq B \leq 10^7$).

Salida

Imprimir en la salida P líneas, cada una con un número entero. Para $i = 1, 2, \dots, P$, el número en la i -ésima línea representa la cantidad de intervalos engrapados totalmente contenidos en el rango $[A, B]$ correspondiente a la i -ésima pregunta.

Entrada de ejemplo	Salida para la entrada de ejemplo
4	1
2184 2200	0
2185 2200	0
2184 2199	13
1 100000	

Problema J – Jurado despistado

AUTOR: LEOPOLDO TARAVILSE - UNIVERSIDAD DE BUENOS AIRES

Este problema fue agregado a último momento reemplazando a otro que el jurado no pudo terminar de preparar a tiempo. No vamos a contarles mucho sobre el problema original porque pensamos usarlo para el torneo del año que viene. Sólo mencionaremos que se llama “Jugando con listas” y trata sobre listas de números enteros. La entrada del problema consistía en L listas de enteros dadas en la entrada de la siguiente manera:

- una línea con un entero N_1 indicando la cantidad de números de la primera lista;
- N_1 líneas con un entero cada una representando los números de la primera lista;
- una línea con un entero N_2 indicando la cantidad de números de la segunda lista;
- N_2 líneas con un entero cada una representando los números de la segunda lista;
- ...
- una línea con un entero N_L indicando la cantidad de números de la última lista;
- N_L líneas con un entero cada una representando los números de la última lista.

En un lamentable accidente, el jurado Joaquín borró la primera línea de cada archivo de entrada (la que contiene N_1). Necesitamos su ayuda para restaurar los archivos de entrada y usar el problema el año que viene. Dado el archivo sin la primera línea, les pedimos que nos digan cuáles son los posibles valores de N_1 tales que, agregando una línea con ese valor, el archivo resultante es una entrada válida según la descripción dada arriba.

Entrada

La primera línea contiene un entero T ($1 \leq T \leq 10^5$) que indica la cantidad de líneas que quedaron en el archivo de entrada. Cada una de las T líneas siguientes contiene un entero R_i que representa el número que quedó en la i -ésima línea del archivo de entrada ($1 \leq R_i \leq 10^5$ para $i = 1, 2, \dots, T$).

Salida

Imprimir en la salida una línea para cada valor posible valor de N_1 . Imprimir todas las posibles respuestas en orden creciente.

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

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

Problema K — Kit de encriptación

AUTOR: WALTER ERQUÍNIGO - FACEBOOK

Karina está desarrollando un poderoso protocolo de encriptación. Para darle difusión, quiere implementar un kit de herramientas para utilizarlo. En particular, necesita una herramienta que le permita hacer una operación complicada sobre cadenas de forma eficiente.

Las posiciones dentro de una cadena las numeramos de izquierda a derecha con naturales consecutivos entre 1 y la longitud de la cadena. La operación que Karina necesita implementar se especifica mediante 4 posiciones dentro de la cadena $i \leq j < k \leq l$ y consta de 3 pasos:

1. intercambiar la subcadena que va desde la posición i hasta la j , inclusive, con la subcadena que va desde la posición k hasta la l , inclusive;
2. dar vuelta ambas subcadenas individualmente;
3. cambiar todos los caracteres de ambas subcadenas por el siguiente carácter en el alfabeto. Es decir, las “a” se convierten en “b”, las “b” en “c”, etc. Para esta operación consideramos al alfabeto como circular, de modo que una “z” se convierte luego de la operación en una “a”.

Por ejemplo, supongamos que tomamos la cadena “alazareselfacil” y las posiciones $i = 3$, $j = 5$, $k = 8$ y $l = 15$. Las dos subcadenas involucradas son “aza” y “selfacil”. Luego del primer paso (intercambio), la cadena queda así “alselfacilreaza”. Luego del segundo paso (dar vuelta) queda “allicaflesreaza”. Finalmente, luego del tercer paso (pasar las letras a la siguiente) se obtiene “almjdbgmftrebab”.

El protocolo desarrollado por Karina aplica la operación descrita arriba sobre una cadena, luego aplica otra operación sobre el resultado, y así sucesivamente, siempre aplicando la nueva operación sobre el último resultado obtenido. Karina necesita saber como queda la cadena, pero lamentablemente sus rudimentarios conocimientos del lenguaje R sólo son suficientes para implementar un algoritmo muy poco eficiente. Para implementar una versión que pueda manejar de forma eficiente muchas operaciones sobre cadenas largas es que los necesita a ustedes.

Entrada

La primera línea contiene una cadena inicial S , que contiene entre 2 y 10^5 letras minúsculas del alfabeto inglés, y un entero N , que representa la cantidad de operaciones a realizar ($1 \leq N \leq 10^5$).

Las siguientes N líneas contienen cada una la descripción de una operación, dada mediante cuatro enteros I , J , K y L , que representan las 4 posiciones de la operación explicada arriba ($1 \leq I \leq J < K \leq L \leq |S|$, donde $|S|$ representa la longitud de la cadena S).

Salida

Imprimir en la salida una línea conteniendo una cadena que representa la cadena resultante luego de aplicar las operaciones dadas en la entrada de forma sucesiva.

Entrada de ejemplo	Salida para la entrada de ejemplo
alazareselfacil 1 3 5 8 15	almjdbgmftrebab

Entrada de ejemplo	Salida para la entrada de ejemplo
alazareselfacil 2 3 5 8 15 3 5 8 15	alcbcf Sugnbgekn

Entrada de ejemplo	Salida para la entrada de ejemplo
aa 1 1 1 2 2	bb

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