Torneo Argentino de Programación

30 de septiembre de 2017

- Departamento de Ciencias e Ingeniería de la Computación
  Universidad Nacional del Sur
- Escuela de Tecnologías de la Información y las Comunicaciones
  Universidad Nacional de Chilecito
- Facultad de Ciencias Exactas, Ingeniería y Agrimensura
  Universidad Nacional del Rosario
- Facultad de Ciencias Exactas y Naturales
  Universidad de Buenos Aires
- Facultad de Ciencias Exactas y Tecnología
  Universidad Nacional de Tucumán
- Facultad de Informática
  Universidad Nacional de La Plata
- Facultad de Informática
  Universidad Nacional del Comahue
- Facultad de Matemática, Astronomía, Física y Computación
  Universidad Nacional de Córdoba
- Facultad Regional Resistencia
  Universidad Tecnológica Nacional
- I. E. S. Número 7 Populorum Progressio
  In.Te.La. Jujuy

Sesión de Competencia

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

Información General

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

Entrada

1. La entrada se debe leer de la entrada estándar (standard input).
2. La entrada contiene un único caso de prueba, el cual se describe utilizando una cantidad de líneas que depende del problema. No hay otros datos en la entrada.
3. Cuando una línea de datos contiene varios valores, éstos se separan utilizando exactamente *un* espacio entre ellos. Ningún otro espacio aparece en la entrada. No hay líneas en blanco.
4. No hay letras con tildes, acentos, diéresis, ni otros signos ortográficos (ñ, Ã, é, Ì, ô, Ü, ç, etcétera).
5. Todas las líneas, incluyendo la última, tienen la marca usual de fin de línea.

Salida

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

Tiempo límite

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

Equipo de desarrollo
Nicolás Álvarez, Pablo Blanc, Brian Curcio, Franco Marino,
Miguel Maurizio, Federico Pousa, Fidel I. Schaposnik, Melanie Sclar,
Leopoldo Taravilse, Alfredo Umfurer, Pablo Zimmermann y Ariel Zylber

Problema A – Audición

Autor: Pablo Zimmermann - Universidad Nacional de Rosario

Eddard Stark: — *Tú te casarás con un gran señor y reinarás su castillo.*
Y tus hijos serán caballeros y príncipes y señores.
Arya Stark: — *No. Esa no soy yo.*

A pesar de que ahora, en un contexto de lucha por la igualdad de género, se recibe con agrado la actitud de Arya, en los tiempos de la abuela Amalia las mujeres no tenían muchas posibilidades de elegir qué querían hacer de sus vidas.

La abuela Amalia llegó a recibirse de profesora de piano, pero de adulta tuvo que dejar sus estudios para ocuparse de los quehaceres del hogar. Más de sesenta años después su nieto tecladista descubrió que ella, a pesar de no haber tocado el piano en todo ese tiempo, tenía la capacidad de reconocer una nota con solo oírla. Sin embargo se percató de que su oído escuchaba las notas más agudas de lo que en realidad eran.

En nuestro sistema tonal, un semitono es la diferencia mínima que puede haber entre una nota y otra. Una escala cromática es un conjunto ordenado de 12 notas, separadas entre sí por un semitono. Para este problema consideramos arbitrariamente las notas de la escala cromática que comienza en la nota “DO”. Las notas de esta escala, de más grave a más aguda, se llaman: “DO”, “DO#”, “RE”, “RE#”, “MI”, “FA”, “FA#”, “SOL”, “SOL#”, “LA”, “LA#” y “SI”. Esta escala se repite de manera cíclica, es decir, luego de cada nota “SI” se encuentra un nuevo “DO” más agudo a un semitono de distancia, y antes de cada “DO” hay un “SI” más grave, también a un semitono de distancia.

Cada vez que Amalia escucha una nota y dice cuál cree que es, su nieto sabe que está corrida S semitonos hacia las notas más agudas con respecto a lo que en realidad suena. Por ejemplo, si $S = 2$ y Amalia dice que escucha un “LA”, la nota que en realidad suena es dos semitonos más grave que la que dice escuchar, o sea un “SOL”.

Dada la nota que Amalia dice escuchar, escriban un programa que le ayude a su nieto a saber cuál es la nota real correspondiente.

Entrada

La entrada consiste en una única línea que contiene un entero S y una cadena N. El entero S representa el corrimiento de semitonos hacia las notas más agudas que escucha Amalia ($1 \leq S \leq 3$). La cadena N es una de las 12 notas de la escala cromática mencionada anteriormente: “DO”, “DO#”, “RE”, “RE#”, “MI”, “FA”, “FA#”, “SOL”, “SOL#”, “LA”, “LA#” y “SI”, y representa la nota que Amalia dice escuchar.

Salida

Imprimir en la salida una línea conteniendo una cadena que representa la nota real que suena sin el corrimiento de semitonos.

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`2 LA`	`SOL`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`1 MI`	`RE#`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`1 DO`	`SI`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`3 SOL#`	`FA`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`3 DO#`	`LA#`

Problema B – Bergantín

Autor: Pablo Blanc - Universidad de Buenos Aires

La banda del pirata Raúl ha acechado los siete mares por muchos años, atacando innumerables barcos, buques, botes, bajeles, balandros, balsas e incluso algunos veleros. Los piratas saqueaban las embarcaciones principalmente interesados en llevarse doblones de oro, guardando los doblones robados en la bodega del barco.

Un día navegando las costas de Italia el pirata Raúl miraba las playas y reflexionaba. Había pasado prácticamente toda su vida en altamar. Cansado de esa vida decidió desarmar la banda. Bajó a la bodega y repartió los doblones de oro en N cofres. Cada uno de los N tripulantes de la embarcación, incluido él mismo, se llevaría un cofre.

Cuando informó la decisión a sus marineros, les preguntó cuántos doblones querían llevarse. Ahora Raúl sabe cuántos doblones tiene cada cofre y cuántos quiere cada marinero. Por supuesto que cualquier marinero está dispuesto a llevarse más doblones que los que pidió. Si Raúl no va a mover doblones entre los cofres, ¿de cuántas formas puede repartir los cofres entre los tripulantes, de modo tal que cada tripulante reciba un cofre con al menos tantos doblones de oro como pidió?

Entrada

La primera línea de la entrada contiene un entero N que representa la cantidad de cofres y piratas en el barco ($1 \leq N \leq 10^5$). La segunda línea contiene N enteros $P_1, P_2, \ldots, P_N$, indicando el entero P_i cuántos doblones de oro quiere el i-ésimo pirata ($1 \leq P_i \leq 10^9$ para $i = 1, 2, \ldots, N$). La tercera línea contiene N enteros $C_1, C_2, \ldots, C_N$, indicando el entero C_i cuántos doblones de oro contiene el i-ésimo cofre ($1 \leq C_i \leq 10^9$ para $i = 1, 2, \ldots, N$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la cantidad de formas en que pueden repartirse los cofres entre los piratas. Como la respuesta puede ser un número muy grande, solo deben imprimir el resto de su división por $10^9 + 7$.

Entrada de ejemplo	Salida para la entrada de ejemplo
2 2 2 1 1000000000	0

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

Entrada de ejemplo	Salida para la entrada de ejemplo
13 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	227020758

Página dejada casi totalmente en blanco en forma absolutamente intencional.

Problema C – Cebando mate

Autor: Nicolás Álvarez - Universidad Nacional del Sur

Carolina tiene la costumbre de juntarse todas las tardes a tomar mate con sus amigos. Como han vivido equivocados toda su vida, les gusta tomar mate dulce. Últimamente, se han preocupado por su ingesta calórica y han decidido probar un nuevo edulcorante cero calorías que salió al mercado: el Ingrediente Caramelizador de Productos Cebables (ICPC). El ICPC tiene la extraña propiedad de que al aplicarlo dura exactamente K cebadas endulzando el mate y luego se evapora completamente.

Carolina y sus amigos se ubican alrededor de una mesa circular, y se numeran del 0 al $N-1$ en el sentido de las agujas del reloj. Luego comienzan a tomar mate durante varias rondas. En cada ronda, ella ceba un mate para cada integrante, comenzando por la persona 0 y continuando en orden ascendente hasta llegar a la persona $N-1$. Por lo tanto, luego de que toma la persona $N-1$ es nuevamente el turno de la persona 0. Carolina decide una cantidad fija **entera** y positiva E_i de ICPC para agregar al mate antes de cebar a la persona i. La cantidad de ICPC que recibe cada persona en su mate será entonces la suma de lo agregado por la cebadora en las últimas K cebadas. Formalmente, la cantidad de ICPC que recibe la persona i a partir de la segunda ronda es

$$T_i = \sum_{d=0}^{K-1} E_{i-d \ (mod \ N)}$$

donde $x \ (mod \ N)$ es un entero entre 0 y $N-1$ que indica el resto de x en la división entera por N.

Por ejemplo, si la ronda constara de $N = 5$ amigos, la duración del edulcorante fuera de $K = 3$ cebadas y las cantidades de ICPC agregado fueran $E_0 = 10$, $E_1 = 4$, $E_2 = 0$, $E_3 = 2$ y $E_4 = 1$, entonces las cantidades de ICPC que recibirían los amigos serían $T_0 = 13$, $T_1 = 15$, $T_2 = 14$, $T_3 = 6$ y $T_4 = 3$.

Carolina conoce muy bien los gustos de sus amigos y quisiera complacerlos a todos. Dado un arreglo G_0, G_1, …, G_{N-1} con las cantidades de edulcorante que quieren recibir los N amigos, ustedes deben determinar si existe un arreglo E_0, E_1, …, E_{N-1} con las cantidades de ICPC a agregar antes de cebar a cada persona, tal que a partir de la segunda ronda todos los amigos estén satisfechos (esto es, $T_i = G_i$ para $i = 0, 1, \ldots, N-1$).

Entrada

La primera línea de la entrada contiene dos enteros N y K, que representan la cantidad de amigos y la duración del ICPC, respectivamente ($1 \leq N \leq 1000$ y $1 \leq K \leq N$). La segunda línea contiene N enteros $G_0, G_1, \ldots, G_{N-1}$, siendo G_i para $i = 0, 1, \ldots, N-1$ la cantidad de edulcorante que quiere recibir la persona i ($0 \leq G_i \leq 10^6$).

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si es posible satisfacer a todos los amigos. El carácter debe ser una '`S`' si es posible, y una '`N`' en caso contrario.

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`5 3` `13 15 14 6 3`	`S`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`3 2` `2 3 7`	`N`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`5 2` `1 1 1 1 1`	`N`

Problema D – Diferencia cambiaria

Autor: Leopoldo Taravilse - Universidad de Buenos Aires

Por muchos años en Nlogonia no estuvo permitida la compra o venta de dólares de Cuadradonia. Recientemente el banco central de Nlogonia decidió liberarlas, pero como esta medida es muy novedosa aún no existe consenso en las distintas ciudades de Nlogonia sobre cuánto vale un dólar cuadradonio en pesos nlogones.

Una página de internet acaba de publicar el precio de compraventa por peso nlogón en cada ciudad del país. Como hay diferencias de precio importantes entre las distintas ciudades, tuvimos una gran idea: viajar en auto comprando y vendiendo dólares cuadradonios y pesos nlogones para incrementar nuestra riqueza. Nlogonia es un país con N ciudades numeradas del 1 al N y conectadas por M rutas. Cada ruta conecta dos ciudades distintas y puede recorrerse en ambos sentidos, no existiendo dos rutas que conecten el mismo par de ciudades. Afortunadamente el precio del combustible no varía de una ciudad a otra, de modo que sabemos de antemano cuánto cuesta recorrer cada ruta. El combustible debe pagarse sí o sí en pesos nlogones, por supuesto que antes de transitar la ruta correspondiente.

Vamos a iniciar nuestro recorrido en la ciudad número 1. Nos gustaría saber cuál es la mínima cantidad entera de pesos nlogones que necesitamos tener al comenzar nuestro viaje para poder incrementar nuestra riqueza tanto como queramos. Como no somos millonarios, si comenzar con 10^6 pesos nlogones no es suficiente consideraremos que este objetivo es imposible.

Entrada

La primera línea contiene dos enteros N y M, que representan la cantidad de ciudades y de rutas de Nlogonia, respectivamente ($1 \leq N \leq 200$ y $1 \leq M \leq N(N-1)/2$). La segunda línea contiene N números reales $C_1, C_2, \ldots, C_N$, dados con exactamente dos dígitos luego del punto decimal ($0.1 \leq C_i \leq 10$ para $i = 1, 2, \ldots, N$). Para $i = 1, 2, \ldots, N$, el número C_i indica que en la i-ésima ciudad podemos cambiar 1 peso nlogón por C_i dólares cuadradonios, o viceversa C_i dólares cuadradonios por 1 peso nlogón. Nótese que en cada ciudad podemos cambiar cualquier cantidad real de pesos nlogones por dólares cuadradonios o viceversa, siempre que se cumpla la proporción correspondiente.

Cada una de las siguientes M líneas contiene tres enteros A, B y P, indicando que existe una ruta que conecta las ciudades A y B, siendo P el costo en pesos nlogones del combustible necesario para recorrerla ($1 \leq A, B \leq N$ con $A \neq B$ y $1 \leq P \leq 1000$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la cantidad mínima de pesos nlogones con los que necesitamos empezar el viaje para poder incrementar nuestro patrimonio tanto como queramos por medio de la compraventa de dólares cuadradonios. Si esto no es posible empezando con a lo sumo de 10^6 pesos nlogones, entonces la respuesta debe ser -1.

Entrada de ejemplo	Salida para la entrada de ejemplo
2 1 0.70 1.40 1 2 1	5

Entrada de ejemplo	Salida para la entrada de ejemplo
3 2 1.00 2.00 3.00 1 2 10 2 3 10	41

Entrada de ejemplo	Salida para la entrada de ejemplo
3 3 0.50 0.50 0.50 1 2 1 1 3 1 2 3 1	-1

Entrada de ejemplo	Salida para la entrada de ejemplo
3 1 1.00 0.70 1.40 2 3 1	-1

Problema E – Enseñando al nieto

Autor: Pablo Zimmermann - Universidad Nacional de Rosario

Nadie entiende el motivo de la predilección de Laino por su nieto Eduardo, por sobre sus otros nietos Emilio y Epifanía. Lo cierto es que los regalos de Eduardo siempre son los más grandes, y recibe también las enseñanzas de Laino sobre su pasión, los juegos de mesa. Eduardo quiere aprender a jugar al Truco, pero como es muy chico y el juego es complejo, Laino creó un juego similar pero más sencillo para enseñarle didácticamente a su nieto preferido. Lo llamó el EduarTruco.

El EduarTruco se juega con siete cartas distintas numeradas del 1 al 7. Antes de iniciar el juego se reparten tres cartas a cada jugador, que las mantiene ocultas de su contrincante. Queda de este modo una carta en el mazo, desconocida por ambos jugadores.

El objetivo del juego consiste en ganar al menos dos de una serie de tres rondas. Se dice que un jugador "es mano" de una ronda si es el que juega primero en dicha ronda. Las tres rondas se juegan sucesivamente, siendo el jugador que ganó una ronda el que será mano en la siguiente. Como Eduardo está aprendiendo a jugar, su abuelo siempre lo deja ser mano en la primera ronda.

En cada ronda, cada jugador usa una de sus cartas que aún no haya utilizado. La persona que es mano comienza por colocar su carta sobre la mesa de modo tal que su oponente pueda verla. Luego, éste debe colocar la suya encima, y el ganador de la ronda resulta ser el jugador que colocó la carta con el número más bajo.

Concluidas las tres rondas cada jugador ha utilizado sus tres cartas, de modo que el juego acaba y se determina el ganador. Como un ejemplo, si Eduardo recibe las cartas 7, 1 y 2 puede asegurarse ganar las rondas en las que juegue las cartas con valores 1 y 2, sin importar las cartas que haya recibido Laino. Por lo tanto, puede asegurarse de ganar la mayoría de las tres rondas, y de este modo el juego.

Eduardo es muy inteligente para su edad, y ya se dio cuenta de que como hay sólo siete cartas puede deducir casi con seguridad las cartas de su abuelo. También sabe que es importante el orden en el cuál tira sus cartas. Sin embargo, todavía le cuesta saber si posee una estrategia ganadora independientemente de las cartas que haya recibido su abuelo o de cómo él juegue. ¿Pueden ayudarlo?

Entrada

La entrada consiste en una línea con tres enteros distintos C_1, C_2 y C_3, que representan las tres cartas que recibe Eduardo al iniciar el juego ($1 \leq C_1, C_2, C_3 \leq 7$).

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si Eduardo puede asegurarse ganar al menos dos de las tres rondas sin importar qué cartas tenga ni cómo juegue su abuelo Laino. El carácter debe ser una '`S`' en caso de que esto sea posible, y una '`N`' caso contrario.

Entrada de ejemplo	Salida para la entrada de ejemplo
`7 1 2`	`S`

Entrada de ejemplo	Salida para la entrada de ejemplo
`6 2 4`	`N`

Entrada de ejemplo	Salida para la entrada de ejemplo
`3 5 1`	`S`

Problema F – Festival

Autor: Nicolás Álvarez - Universidad Nacional del Sur

Este año llega nuevamente a la Argentina el famoso festival de LeoPablozza. Leo y Pablo están planeando asistir, pero les gustaría saber de antemano cuánto van a disfrutar del show. Sólo están interesados en los recitales que se desarrollan en el escenario principal, y lógicamente no puede haber más de un recital simultáneamente en dicho escenario. Cada recital tiene una hora de comienzo y finalización, que por supuesto los músicos respetan a rajatabla.

Leo y Pablo han asignado a cada recital un valor entero que depende de cuánto les gusta el músico o banda que toca, y que puede ser negativo si se trata de músicos que detestan. La felicidad que les va a provocar asistir al festival en un cierto rango horario es igual a la suma de los valores asignados a los recitales para los que escucharán al menos un fragmento del mismo.

Los organizadores del festival pueden ir reprogramando los recitales, decidiendo cancelar algunos o agregar otros nuevos. A este dúo dinámico le gustaría saber cuál es la felicidad que le provocará ir en los rangos horarios que tiene disponibles, que también varían a medida que se aproxima la fecha del festival. ¿Pueden ayudarlos con esta tarea?

Entrada

La primera línea contiene un entero Q, que representa la cantidad de consultas a ser procesadas ($2 \leq Q \leq 2 \times 10^5$). Cada una de las siguientes Q líneas contiene la descripción de una consulta, que puede ser de cualquiera de los siguientes tres tipos:

- Un 1 seguido de tres enteros I, F y V representa que los organizadores deciden agregar un nuevo recital que inicia al comenzar el minuto I y finaliza al comenzar el minuto F, siendo V el valor asignado al recital por Leo y Pablo ($1 \leq I \leq F \leq 10^5$ y $-10^4 \leq V \leq 10^4$).

  Inicialmente no hay ningún recital programado, y cada recital recibe al momento de ser agregado un número identificatorio que será el menor número entero positivo que no haya sido utilizado aún como número identificatorio.

- Un 2 seguido de un entero K representa que los organizadores deciden cancelar el recital con el número identificatorio K.

- Un 3 seguido de dos enteros A y B representa que Leo y Pablo tienen disponible el rango horario entre el comienzo de los minutos A y B ($1 \leq A < B \leq 10^5$). Quieren saber cuánta felicidad les provocaría asistir al festival en dicho rango horario, es decir la suma de los valores de los recitales para los que escucharían al menos un minuto.

Todas las consultas de la entrada son válidas, esto es, en ningún momento habrá dos recitales programados simultáneamente, ni se cancelarán recitales no programados aún o ya cancelados anteriormente. Se asegura que en la entrada hay al menos una consulta de tipo 3.

Salida

Imprimir en la salida una línea por cada consulta de tipo 3, conteniendo un entero que representa el valor de la felicidad de Leo y Pablo correspondiente al rango horario consultado.

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`7`	`0`
`3 10 15`	`3`
`1 1 5 4`	`0`
`1 7 10 -1`	`4`
`3 4 8`	
`3 5 6`	
`2 2`	
`3 4 8`	

Problema G – Grand Slam

Autor: Federico Pousa - Universidad de Buenos Aires

Rogelio es un gran jugador de tenis, de hecho es el mejor jugador de tenis de toda la historia. Rogelio siempre está buscando nuevos desafíos. Este año decidió que va a participar del Grand Slam Argentino de Tejo, organizado por los *Tejistas Asociados Profesionales* (TAP). Para Rogelio siempre fue muy importante manejar el aspecto psicológico de los deportes, por lo que para poder ganar su primer Grand Slam de Tejo quiere interiorizarse un poco más en el deporte. Rogelio sabe que en el tenis no existen demasiadas opciones diferentes para el desarrollo de un set, pero le gustaría conocer qué sucede en el tejo para saber a cuántas situaciones diferentes tendrá que enfrentarse eventualmente. Específicamente, Rogelio quiere saber, dado el puntaje final de un partido, de cuántas maneras diferentes puede desarrollarse el mismo.

El Tejo es un juego que consta de N rondas entre dos equipos. El puntaje final de cada equipo es la suma de los puntajes que obtuvieron respectivamente en cada ronda. El puntaje de un equipo en una ronda es un número entero entre 0 y M inclusive, y por cómo son las reglas de este juego, en cada ronda al menos uno de los dos equipos tiene puntaje 0. Dados P_1 y P_2, los puntajes finales de los equipos en un partido de tejo, Rogelio quiere saber de cuántas maneras se puede desarrollar el partido. Dos formas de desarrollo del partido se consideran distintas si y solo si en alguna de las rondas el puntaje de alguno de los equipos difiere.

Entrada

La entrada consiste de una única línea que contiene cuatro enteros N, M, P_1 y P_2. El entero N indica la cantidad de rondas del partido ($1 \leq N \leq 100$). El entero M indica el puntaje máximo posible en una ronda para un equipo ($1 \leq M \leq 100$). Los enteros P_1 y P_2 representan los puntajes finales de los equipos para un partido ($1 \leq P_1, P_2 \leq 1000$).

Salida

Imprimir en la salida una línea conteniendo un entero que indica la cantidad de formas diferentes en las que se puede desarrollar el partido. Como la respuesta puede ser un número muy grande, solo se debe imprimir el resto de su división por 10^9+7.

Entrada de ejemplo	Salida para la entrada de ejemplo
`3 2 2 2`	`12`

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

Entrada de ejemplo	Salida para la entrada de ejemplo
`5 12 3 2`	`200`

Página dejada casi totalmente en blanco en forma absolutamente intencional.

Problema H – Horizonte Montañoso II

Autor: Pablo Blanc - Universidad de Buenos Aires

Horacio y Hernán gustan mucho de recorrer la Cordillera de los Andes, apreciando la agradable vista de las montañas en el horizonte. Este año han decidido participar de un importante torneo de programación y por ello no han podido disfrutar de los paisajes que tanto aman. Con un poco de nostalgia se dispusieron a mirar en la computadora las fotos de viajes pasados, esperando ver muchas montañas.

Las fotos digitales son grillas cuadriculadas cuyas celdas se conocen como píxeles. Lamentablemente las fotos de Horacio y Hernán están dañadas, y donde antes estaban los bellos paisajes de montaña ahora solo hay píxeles blancos y negros. Hay píxeles blancos donde antes estaba el cielo y negros donde estaban las montañas. De este modo, en cada columna de píxeles se ve de abajo hacia arriba un conjunto de píxeles negros seguido de otro de píxeles blancos. En la Figura 1 se presenta una foto y la correspondiente cuadrícula, donde podemos identificar más claramente los píxeles.

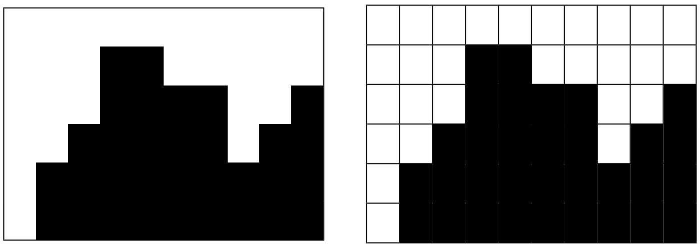

Figura 1: Una foto dañada.

En las fotos cada montaña ocupa cierta cantidad de columnas consecutivas. De izquierda a derecha, las cantidades de píxeles negros de las mismas forman una sucesión no decreciente seguida de otra sucesión no creciente. El paisaje visible en las fotos se forma al superponer varias montañas. Por ejemplo, el paisaje de la foto de la Figura 1 puede estar formado por las tres montañas de la Figura 2. Sin embargo, observamos que también puede deberse a solamente dos montañas como las de la Figura 3.

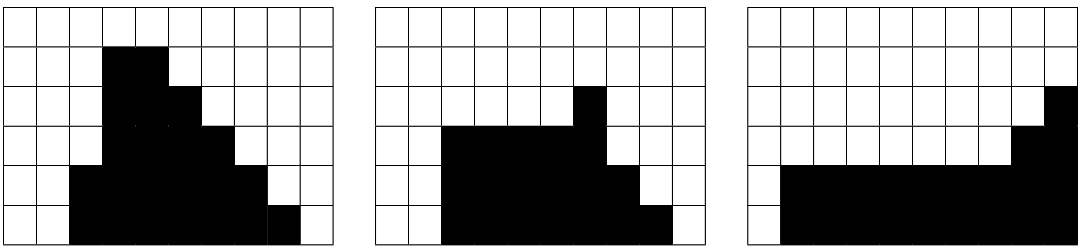

Figura 2: Tres montañas que superpuestas forman el paisaje de la Figura 1.

A causa del daño que sufrieron las fotos, Horacio y Hernán ya no pueden distinguir las montañas que se superponen en las mismas. Sin embargo en algunas fotos dañadas sí se puede distinguir que hay varias montañas. Por ejemplo, en la foto de la Figura 1 podemos estar seguros de que hay al menos dos montañas.

Para evitar que Horacio y Hernán se depriman aún más, vamos a mostrarles que en algunas de sus fotos se puede asegurar que hay múltiples montañas. Dada una foto,

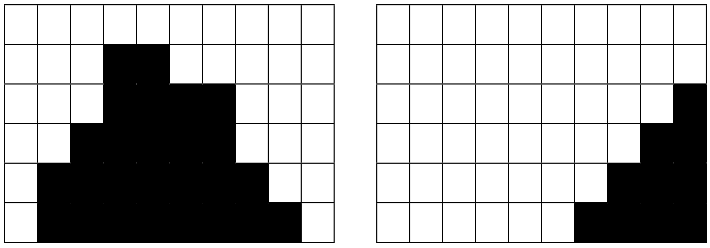

Figura 3: Dos montañas que superpuestas también forman el paisaje de la Figura 1.

¿cuántas montañas podemos afirmar con certeza que aparecen en la foto formando el paisaje?

Entrada

La primera línea de la entrada contiene un entero N, que representa la cantidad de columnas en la foto ($2 \leq N \leq 10^5$). La segunda línea contiene N enteros $X_1, X_2, \ldots, X_N$, representando el entero X_i la cantidad de píxeles negros en la i-ésima columna de la foto, de izquierda a derecha ($0 \leq X_i \leq 10^9$ para $i = 1, 2, \ldots, N$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la mayor cantidad de montañas que podemos afirmar con certeza que aparecen en la foto.

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`10` `0 2 3 5 5 4 4 2 3 4`	`2`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`2` `0 1000000000`	`1`

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

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`3` `0 0 0`	`0`

Entrada de ejemplo	**Salida para la entrada de ejemplo**
`2` `1 1`	`1`

Problema I – Imitaciones

Autor: Leopoldo Taravilse - Universidad de Buenos Aires

Irene, la presidenta del Instituto para la Conservación de Polígonos y Curvas (ICPC), acaba de convocar a la comisión directiva del ICPC para una reunión urgente. Aparentemente, un polígono muy valioso que se encontraba en exhibición en el hall del Teatro de Áreas y Polígonos (TAP) habría sido robado.

La noticia llegó a Irene gracias a que uno de los empleados del TAP notó que el polígono en exposición era distinto al que había visto hacía unos días. Irene le pidió a los directores del TAP que hicieran una investigación, y ellos con su habitual eficiencia pudieron determinar que efectivamente no es el mismo que habían colocado allí originalmente. Temen entonces que haya sido reemplazado por una imitación. Sin embargo, es posible que por cuestiones climáticas como el viento o la humedad el polígono se haya rotado, desplazado y/o redimensionado (esto es, que las longitudes de sus lados se hayan multiplicado todas por una misma constante, manteniéndose los ángulos del polígono en su correspondiente orden).

Como miembros de la comisión directiva del ICPC, su tarea es determinar si el polígono es efectivamente el original, habiendo posiblemente sufrido una rotación, desplazamiento y/o redimensionamiento. Para ello, disponen de las coordenadas en el plano cartesiano de los vértices del polígono original, así como también las del polígono que se encuentra en este momento en el hall del TAP. ¿Pueden cumplir con tan importante tarea?

Entrada

La primera línea de la entrada contiene un entero N que representa la cantidad de vértices de ambos polígonos ($3 \leq N \leq 2000$). Cada una de las siguientes N líneas contiene dos enteros $X_{1,i}$ y $Y_{1,i}$, indicando las coordenadas del i-ésimo vértice del polígono original, en sentido horario. Finalmente, cada una de las siguientes N líneas contiene dos enteros $X_{2,i}$ y $Y_{2,i}$, indicando las coordenadas del i-ésimo vértice del polígono que se encuentra actualmente en exposición, también en sentido horario. Todas las coordenadas satisfacen $-10^4 \leq X_{1,i}, Y_{1,i}, X_{2,i}, Y_{2,i} \leq 10^4$, y en ninguno de los dos polígonos hay vértices que formen parte de un lado con extremos en otros dos vértices. Notar que los polígonos de la entrada pueden tener lados que se intersequen entre sí, es decir, pueden ser polígonos no simples.

Salida

Imprimir en la salida una línea conteniendo una cadena que representa si el segundo polígono puede obtenerse aplicando un desplazamiento, rotación y/o redimensionamiento del primer polígono. Si esto es posible, la cadena impresa debe ser "`MISMO`", caso contrario debe ser "`OTRO`".

Entrada de ejemplo	**Salida para la entrada de ejemplo**
5 0 0 0 2 1 2 1 1 2 0 2 4 4 3 3 1 4 -2 0 0	MISMO

Entrada de ejemplo	**Salida para la entrada de ejemplo**
3 0 0 0 2 1 0 0 0 0 1 0 -2	OTRO

Entrada de ejemplo	**Salida para la entrada de ejemplo**
4 0 0 0 1 0 2 1 0 2 0 1 1 0 2 4 6	OTRO

Entrada de ejemplo	**Salida para la entrada de ejemplo**
3 0 0 0 1 2 0 0 0 0 1 -2 0	OTRO

Problema J – Joaquín Jacinto

Autor: Leopoldo Taravilse - Universidad de Buenos Aires

Una de las tareas más importantes a la hora de organizar nuestro tan querido Torneo Argentino de Programación consiste en seleccionar los problemas para utilizar en la prueba. Afortunadamente contamos generalmente con muchas propuestas, entre las que elegimos los problemas más interesantes.

Una de estas propuestas recibidas en 2014 fue el problema *Jugando con listas*. Casi todos los miembros del jurado pensaban que era un buen problema, pero Joaquín no estaba de acuerdo e intencionalmente modificó los archivos de los casos de prueba para que no pudieran ser utilizados. Intentamos volver a utilizar este problema en 2015 y 2016, pero no fue posible porque Joaquín volvió a integrar el jurado, usando su segundo nombre, Jacinto, para no ser descubierto. ¡Recién este año descubrimos que Joaquín y Jacinto eran de hecho la misma persona!

Ahora queremos saber cuán grave fue nuestra omisión durante todo este tiempo. Todos los años la cantidad de jurados del TAP es la misma, más allá de que puedan ser o no las mismas personas. Recopilamos los nombres de los M jurados de cada una de las N ediciones anteriores del TAP, y queremos determinar si es posible que a lo largo de las N ediciones todos los jurados hayan sido siempre los mismos. Sabemos que cada jurado tiene exactamente dos nombres, y puede utilizar cualquiera de ellos cada año (o, posiblemente, siempre el mismo). También sabemos que los nombres de los jurados son todos distintos.

Por ejemplo, si hubo $N = 2$ TAPs con $M = 2$ jurados cada uno, siendo sus nombres Fidel y Ramiro en el primero, e Iván y Augusto en el segundo, es posible que se trate de solamente dos personas. En efecto, Fidel e Iván pueden ser los dos nombres de una misma persona, al igual que Ramiro y Augusto. Si, en cambio, hubo $N = 4$ TAPs con $M = 2$ jurados cada uno, siendo sus nombres Fidel y Ramiro en el primero, Iván y Augusto en el segundo, Iván y Ramiro en el tercero, y Fidel e Iván en el cuarto, ya no es posible que se trate de sólo dos personas. Esto es porque Iván comparte TAPs con Fidel, Ramiro y Augusto, de modo que debe ser una persona distinta a todos ellos. Por lo tanto, la única opción restante es que ellos tres sean una misma persona, pero esto es imposible porque dijimos que todos los jurados tienen exactamente dos nombres.

Entrada

La primera línea de la entrada contiene dos enteros N y M, que representan la cantidad de ediciones del TAP y la cantidad de jurados en cada TAP, respectivamente ($1 \leq N \leq 100$ y $1 \leq M \leq 20$). Cada una de las siguientes N líneas describe una edición distinta del TAP, y contiene los M nombres de los jurados de dicha edición. Todos los nombres de la entrada son cadenas compuestas por a lo sumo 10 caracteres del alfabeto inglés de letras minúsculas.

Salida

Imprimir en la salida una línea conteniendo un carácter que representa si es posible que todos los TAPs hayan tenido a los mismos jurados. El carácter debe ser una '`S`' si esto es posible, y una '`N`' caso contrario.

Entrada de ejemplo	**Salida para la entrada de ejemplo**

`2 2`
`fidel ramiro`
`ivan augusto`
	`S`

Entrada de ejemplo	**Salida para la entrada de ejemplo**

`4 2`
`fidel ramiro`
`ivan augusto`
`ivan ramiro`
`fidel ivan`
	`N`

Entrada de ejemplo	**Salida para la entrada de ejemplo**

`5 3`
`fidel ramiro joaquin`
`fidel ramiro joaquin`
`ivan ramiro jacinto`
`fidel joaquin augusto`
`ivan augusto jacinto`
	`S`

Problema K – Katsura da

Autor: Franco Marino - Universidad Nacional de Córdoba

Katsura tiene una secuencia de N números enteros no negativos, $S_1, S_2, \ldots, S_N$. Sobre esta secuencia, él puede realizar la siguiente operación cualquier número de veces (posiblemente cero):

- Elegir dos elementos S_i y S_j con $i \neq j$ y efectuar los reemplazos $S_i \to 2 \times S_i$ y $S_j \to \lfloor S_j/2 \rfloor$.

Por $\lfloor x/2 \rfloor$ denotamos la división entera por 2. Por ejemplo, $\lfloor 0/2 \rfloor = 0$, $\lfloor 1/2 \rfloor = 0$, $\lfloor 2/2 \rfloor = 1$, $\lfloor 3/2 \rfloor = 1$, $\lfloor 10/2 \rfloor = 5$ y $\lfloor 17/2 \rfloor = 8$.

Su tarea es determinar la mínima cantidad de operaciones que necesita hacer Katsura para obtener una secuencia en la que todos los elementos sean números pares.

Por ejemplo, para la secuencia $S_1 = 1$, $S_2 = 1$ y $S_3 = 1$ se requieren al menos dos operaciones. Una posible primera operación sería elegir los elementos $S_2 = 1$ y $S_1 = 1$, y realizar los reemplazos $S_2 \to 2 \times S_2$ y $S_1 \to \lfloor S_1/2 \rfloor$, obteniendo la secuencia $S_1 = 0$, $S_2 = 2$ y $S_3 = 1$. Para la segunda operación, se podrían elegir los elementos $S_2 = 2$ y $S_3 = 1$, realizando entonces los reemplazos $S_2 \to 2 \times S_2$ y $S_3 \to \lfloor S_3/2 \rfloor$ para obtener la secuencia $S_1 = 0$, $S_2 = 4$ y $S_3 = 0$, cuyos elementos son todos pares.

Entrada

La primera línea de la entrada contiene un entero N que representa la cantidad de elementos de la secuencia ($2 \leq N \leq 3 \times 10^5$). La segunda línea contiene N enteros $S_1, S_2, \ldots, S_N$, representando la secuencia que tiene Katsura ($0 \leq S_i \leq 10^9$ para $i = 1, 2, \ldots, N$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la mínima cantidad de operaciones que necesita hacer Katsura para obtener una secuencia en la que todos los elementos sean números pares.

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

Entrada de ejemplo	Salida para la entrada de ejemplo
2 6 2	0

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

Página dejada casi totalmente en blanco en forma absolutamente intencional.

Problema L – LOST

Autor: Pablo Blanc - Universidad de Buenos Aires

Leia y Luke se sienten perdidos, mañana es su cumpleaños y aún no tienen listos los souvenirs para sus invitados. Tienen muchas fichas con la forma de alguno de los tetrominoes de la Figura 1 (un tetrominoe es la forma geométrica compuesta por cuatro cuadrados conectados entre sí por algunos de sus lados). Van a repartir bolsitas con cuatro tetrominoes cada una, y quieren hacerlo de forma tal que con los cuatro tetrominoes de una bolsita sea posible armar un cuadrado formado por 4×4 cuadrados menores. Para hacerlo es posible rotar las fichas y/o darlas vuelta para obtener la ficha espejada.

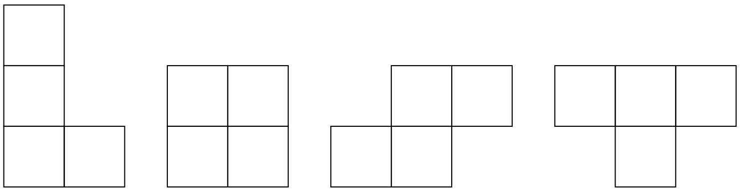

Figura 1: Cuatro tetrominoes que, de izquierda a derecha, tienen la forma de las letras L, O, S y T.

Leia y Luke ya contaron cuántas fichas tienen de cada tipo. ¿Pueden ayudarlos a determinar cuál es el número máximo de bolsas que pueden armar con ellas?

Entrada

La entrada consiste de una única línea conteniendo cuatro enteros L, O, S y T, que representan respectivamente la cantidad de fichas de tipo L, O, S y T que tienen Leia y Luke ($0 \leq L, O, S, T \leq 10^9$).

Salida

Imprimir en la salida una línea conteniendo un entero que representa la máxima cantidad de bolsas que pueden armar Leia y Luke con las fichas que tienen.

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

Entrada de ejemplo	Salida para la entrada de ejemplo
`4 4 4 4`	`4`

Entrada de ejemplo	Salida para la entrada de ejemplo
`1000000000 0 0 0`	`250000000`

Entrada de ejemplo	Salida para la entrada de ejemplo
`99 99 99 99`	`98`

Página dejada casi totalmente en blanco en forma absolutamente intencional.