Problema A
Alfajores

Seba es el gerente del Taller de Avioncitos de Papel (TAP), una empresa muy grande dedicada al arte de la papiroflexia. El TAP cuenta con un edificio muy grande que posee M oficinas. En la i-ésima oficina trabajan E_i empleados.

Debido a la gran demanda de su producto, Seba viaja constantemente. Al regresar de sus viajes, tiene por costumbre traer una gran caja de alfajores para compartir con sus empleados.

Para repartirlos, visita cada una de las M oficinas de la empresa en orden, desde la 1 hasta la M.

Cuando llega a la i-ésima oficina, reparte tantos alfajores como le sea posible en partes iguales entre los E_i empleados de la oficina. Luego de repartirles, toma la caja con los alfajores restantes y pasa con ellos a la siguiente oficina.

Una vez que ha visitado todas las oficinas, se sienta en su escritorio y disfruta de los alfajores restantes.

Seba tiene miedo de estar excediéndose con los dulces, y por eso necesita saber cuántos alfajores ha consumido. El problema es que no lleva el registro de la cantidad que quedó en la caja luego de la repartición correspondiente a cada viaje. Por suerte, cuenta con los N tickets correspondientes a las compras de alfajores, y como sabe cuántas personas trabajan en cada oficina, está seguro de que podrás calcular dichas cantidades por él.

Entrada

Una primera línea con dos enteros N y M ($1 \leq N, M \leq 10^5$), que representan la cantidad de viajes que hizo Seba y la cantidad de oficinas del TAP.

Una segunda línea con N enteros $A_1, A_2, \ldots, A_N$ ($1 \leq A_i \leq 10^9$), donde A_i es la cantidad de alfajores que compró en el i-ésimo viaje.

Finalmente, una tercera línea con M enteros $E_1, E_2, \ldots, E_M$ ($1 \leq E_i \leq 10^9$), donde E_i es la cantidad de personas que trabajan en la i-ésima oficina.

Salida

Una única línea con N enteros, las cantidades de alfajores que quedaron en la caja luego de la repartición correspondiente a cada uno de los viajes de Seba.

Ejemplo de entrada 1	**Ejemplo de salida 1**

`3 3`
`140 79 5`
`90 42 5`
	`3 2 0`

Ejemplo de entrada 2	**Ejemplo de salida 2**

`10 8`
`120 456 7458 84 123 84 213 185 987 654`
`97 73 61 41 52 23 11 7`
	`0 0 2 0 3 0 1 4 6 0`

En el primer ejemplo, Seba trae 79 alfajores en el segundo viaje y al pasar por el primer piso, como no hay suficiente para que cada empleado tome uno, ninguno recibe alfajores. Al pasar por el segundo piso, cada uno toma 1 alfajor, quedando 37 en la caja. Finalmente, llega al tercer piso donde cada empleando se lleva 7, quedando así 2 alfajores para Seba.

Problema B
Blanquinegro

Fernanda está jugando con fichas sobre un tablero. El tablero es una cuadrícula de $N \times N$ casillas, algunas pintadas de blanco y otras de negro. Cada casilla mide 1cm $\times$ 1cm.

Fernanda utilizará fichas de dominó rectangulares, de 1cm $\times$ 2cm. Las colocará en posición **horizontal**, de modo que ocupen **exactamente dos casillas** vecinas **horizontalmente** en el tablero.

El juego impone la regla de que solamente puede colocar una ficha de dominó sobre dos casillas negras. Las casillas blancas están completamente prohibidas. Además, nunca se puede ocupar una misma casilla con más de una ficha.

Dados los colores de las $N \times N$ casillas, ¿puedes ayudar a Fernanda indicando cuál es la máxima cantidad de fichas que puede colocar en el tablero?

Entrada

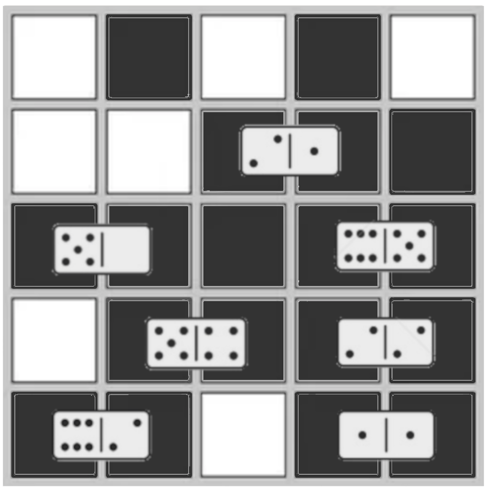
Una primera línea con un entero N $(1 \leq N \leq 50)$, el tamaño del tablero. Luego N líneas en las que se describe el tablero. La i-ésima línea contiene una cadena de N caracteres `N` o `B`, que indican los colores (`N` para negro y `B` para blanco) de las casillas en la i-ésima fila, en orden de izquierda a derecha.

Salida

Una única línea con un entero que indica la máxima cantidad de fichas que se puede colocar en el tablero.

Ejemplo de entrada 1	Ejemplo de salida 1
5 BNBNB BBNNN NNNNN BNNNN NNBNN	7

En el ejemplo, es posible colocar 7 fichas en el tablero como se muestra en la siguiente figura. No hay manera de colocar 8 o más fichas, por lo que la respuesta es 7.



Problema C
Cromático

Un grupo de N amigas va a pintar sus casas de violeta. Tienen N tarros de pintura azul y N de pintura roja. Van a distribuir la pintura de manera que cada una reciba un tarro de cada color. No es importante cuánta pintura de cada color reciben, pero quieren que la distribucion del total de pintura sea lo más equitativa posible. Es decir, quieren que la diferencia entre la persona que más pintura recibe y la que menos recibe sea lo más chica posible.

Dada la cantidad de pintura en cada tarro, ¿cuál es el mínimo de la diferencia?

Entrada

Una primera línea con un entero N ($2 \leq N \leq 10^5$), la cantidad de amigas.

Una segunda línea con N enteros: $A_1, \ldots, A_N$ ($1 \leq A_i \leq 10^9$), la cantidad de pintura en cada uno de los N tarros de pintura roja.

Finalmente, una tercera línea con N enteros: $B_1, \ldots, B_N$ ($1 \leq B_i \leq 10^9$), la cantidad de pintura en cada uno de los N tarros de pintura azul.

Salida

Una única línea con un entero que indica el menor valor posible de la diferencia entre la cantidad de pintura que recibe la persona que recibe más pintura, y la cantidad que recibe la persona que recibe menos.

Ejemplo de entrada 1	Ejemplo de salida 1

`2`
`1 2`
`1 2`
	`0`

Ejemplo de entrada 2	Ejemplo de salida 2

`3`
`1000000000 1 1000000000`
`1000000000 1 1000000000`
	`999999999`

En el primer ejemplo, la distribución óptima de pintura se puede obtener de la siguiente forma: La primera de las amigas recibe 1 unidad de pintura azul y 2 unidades de pintura roja, que en total son 3 unidades de pintura. La segunda recibe 1 unidad de pintura roja y 2 unidades de pintura azul, que en total son 3 unidades de pintura. Por lo tanto, la diferencia entre la que más pintura recibe y la que menos pintura recibe es de $3 - 3 = 0$.

Problema D

Decidiendo los problemas

Como parte de una nueva iniciativa, la AAPC (Asociación Argentina de Programación Competitiva) busca organizar varias competencias anuales. Realizar una *competencia* consiste en tomar *problemas* de K *niveles* distintos, numerados desde 1 hasta K en orden creciente de dificultad. Concretamente, toda competencia debe tener exactamente C_i problemas asignados al nivel i, para cada i entre 1 y K.

A su vez, cada problema tiene un cierto *grado de dificultad*. Que un problema tenga un grado de dificultad igual a d, significa que ese problema puede ser asignado en una competencia a cualquier nivel mayor o igual a d.

Naturalmente no se pueden repetir problemas en las competencias, por lo tanto un problema no puede ser asignado en más que una competencia, y además, dentro de una misma competencia solo puede ser asignado a un único nivel.

La AAPC solicitó la colaboración de sus socios para que propongan problemas. En total al juntar todas las propuestas de problemas, la asociación dispone de P_i problemas de grado de dificultad i que puede utilizar en las competencias que desea organizar, para cada i entre 1 y K inclusive.

Utilizando algunos de estos problemas (potencialmente todos, pero no necesariamente), ¿cuál es la mayor cantidad de competencias que puede armar la asociación?

Entrada

Una primera línea con un entero $K (1 \leq K \leq 10^5)$, la cantidad de niveles distintos que hay en las competencias que organiza la asociación.

Luego, una segunda línea con K enteros $C_1, C_2, \ldots, C_K$, donde cada $C_i (1 \leq C_i \leq 10^9)$ indica la cantidad de problemas asignados al nivel i que debe tener una competencia organizada por la asociación.

Finalmente, una tercera línea con K enteros $P_1, P_2, \ldots, P_K$, donde $P_i (0 \leq P_i \leq 10^9)$ denota la cantidad de propuestas de problemas de grado de dificultad i de las que dispone la asociación para conformar las competencias que desea llevar a cabo.

Salida

Una única línea con un entero, que representa la mayor cantidad de competencias que puede conformar la asociación respetando las cantidades requeridas por nivel utilizando los problemas que tiene a disposición.

Ejemplo de entrada 1	**Ejemplo de salida 1**

`5`
`2 2 2 2 2`
`100 0 0 0 0`
	`10`

Ejemplo de entrada 2	**Ejemplo de salida 2**

`5`
`2 3 2 2 2`
`10 0 9 3 1`
	`2`

Ejemplo de entrada 3	**Ejemplo de salida 3**

`5`
`2 3 2 2 2`
`0 100 7 0 6`
	`0`

Ejemplo de entrada 4	**Ejemplo de salida 4**

`5`
`2 3 2 2 2`
`10 0 7 0 6`
	`1`

En el primer ejemplo cada competencia consiste de 5 niveles, y debe tener 2 problemas asignados a cada nivel. Se dispone de 100 problemas de grado de dificultad igual a 1, que pueden ser asignados a cualquier nivel. Por lo tanto en total se pueden realizar 10 competencias con esos problemas.

Problema E

Encontrando progresiones

Catalina necesita ayuda con su tarea de Matemáticas, la cual consiste en encontrar todas las *progresiones aritméticas crecientes de números enteros* (*) que cumplen las siguientes condiciones:

- La progresión contiene al número A
- La suma de los términos de la progresión es S
- Todos los términos de la progresión se encuentran en el rango $[L, R]$

Catalina es una chica responsable y no quiere que alguien más resuelva la tarea en su lugar, pero le vendría bien un poco de ayuda. Lo que Catalina necesita es que le digas cuántas progresiones existen bajo dichas condiciones, para que así ella pueda saber cuándo dejar de buscar.

(*) Una *progresión aritmética creciente de números enteros* es una lista no vacía de n números enteros $x_1, x_2, \ldots, x_n$ donde se cumple que $x_i - x_{i-1} = D > 0$ para todo entero i tal que $2 \leq i \leq n$, donde D es un número entero positivo. Por ejemplo, $[7]$, $[1, 2, 3]$ y $[2, 4, 6, 8]$ son progresiones válidas, mientras que $[5, 1]$, $[2, 4, 8]$ y $[3, 6, 9, 6]$ no lo son.

Entrada

Una única línea con cuatro enteros A, S, L y R ($1 \leq L \leq A \leq R \leq 10^{12}$, $1 \leq S \leq 10^{18}$, $0 \leq R - L \leq 10^5$).

Salida

Una única línea con un entero indicando cuántas progresiones cumplen las condiciones que describe el enunciado.

Ejemplo de entrada 1	**Ejemplo de salida 1**
`5 15 1 10`	`6`

Ejemplo de entrada 2	**Ejemplo de salida 2**
`5 15 2 10`	`4`

Ejemplo de entrada 3	**Ejemplo de salida 3**
`7 30 3 26`	`4`

Ejemplo de entrada 4	**Ejemplo de salida 4**
`5 5 5 5`	`1`

Las 6 progresiones válidas para el primer ejemplo son: $[5, 10]$, $[1, 5, 9]$, $[2, 5, 8]$, $[3, 5, 7]$, $[4, 5, 6]$ y $[1, 2, 3, 4, 5]$

Problema F
Frío en la playa

Un grupo de amigos, ya retirados de la programación competitiva, decidieron improvisadamente tomarse una semana de vacaciones en la ciudad de la costa Mar de AJI. Como no miraron el pronóstico, al llegar se encontraron días nublados y ventosos. Pero las condiciones meteorológicas no iban a arruinarles sus vacaciones.

Para divertirse en la playa, decidieron organizar un torneo de *tejo* entre dos equipos, llamados `A` (equipo azul) y `R` (equipo rojo). El tejo se juega en una cancha de forma rectangular delimitada sobre la arena que tiene W centímetros de ancho y L centímetros de largo. Las esquinas de esta cancha tienen coordenadas $(0, 0)$, $(W, 0)$, $(0, L)$ y (W, L).

Inicialmente hay un *tejín* en la posición (T_x, T_y). Luego, cada equipo realiza N lanzamientos de discos (llamados *tejos*) buscando quedar lo más cercano al tejín posible (inclusive pudiendo quedar en la misma posición del tejín, situado por encima de él).

Al finalizar los N lanzamientos, el equipo que haya lanzado el tejo que quedó más cerca del tejín es el equipo ganador. Además, el equipo ganador recibe un punto por cada tejo que esté más cerca del tejín que el tejo del equipo rival que se encuentra más cercano al tejín. La distancia de un tejo al tejín se mide como la distancia euclídea del centro del tejín al centro del tejo.

Además de los lanzamientos de los equipos, se conoce la posición del centro del tejín. Se asegura que todos los lanzamientos se ubican dentro de los límites de la cancha (o sobre la línea) y que en la entrada no hay dos tejos a la misma distancia del tejín. Se te pide calcular qué equipo resultó ganador y cuántos puntos obtuvo.

Entrada

Una primera línea con un entero $N (1 \leq N \leq 1000)$, la cantidad de lanzamientos que realizó cada equipo.

Luego, una segunda línea con cuatro enteros W, L, T_x y T_y. Los primeros dos corresponden a las dimensiones en ancho y largo de la cancha $(1 \leq W \leq 10^4, 1 \leq L \leq 10^4)$, y los últimos dos a la ubicación del tejín $(0 \leq T_x \leq W, 0 \leq T_y \leq L)$.

Luego N líneas, cada una con dos enteros que describen cada lanzamiento del equipo `A`. La i-ésima línea contiene dos enteros X_i, Y_i donde X_i indica la ubicación en ancho del i-ésimo lanzamiento $(0 \leq X_i \leq W)$, e Y_i denota la ubicación en largo del i-ésimo lanzamiento $(0 \leq Y_i \leq L)$ del equipo `A`.

Finalmente siguen N líneas que describen de la misma forma los lanzamientos del equipo `R`.

Salida

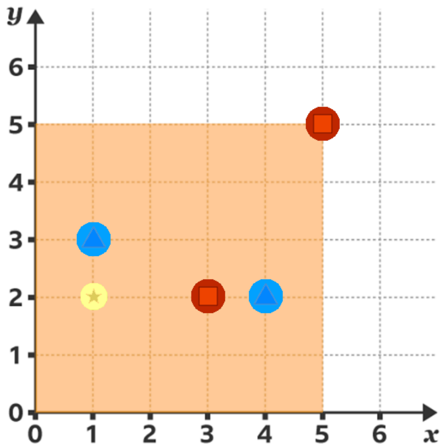
Una única línea con dos valores separados por un espacio. El primero debe ser `A` o `R` según qué equipo resultó ganador, y el segundo la cantidad de puntos que recibe dicho equipo al finalizar todos los lanzamientos.

Ejemplo de entrada 1	**Ejemplo de salida 1**

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

Ejemplo de entrada 2	**Ejemplo de salida 2**
3 10 10 0 5 0 0 0 2 0 4 0 1 0 3 0 5	R 1

Ejemplo de entrada 3	**Ejemplo de salida 3**
3 10 10 0 5 0 3 0 4 0 5 0 0 0 1 0 2	A 3



La imagen corresponde al primer ejemplo. Cada equipo realizó dos lanzamientos. Los tejos azules (con un triángulo en su interior) corresponden al equipo `A` en las posiciones $(1, 3)$ y $(4, 2)$. Los tejos rojos (con un cuadrado en su interior) al equipo `R`, en las posiciones $(3, 2)$ y $(5, 5)$. La cancha mide 5 centímetros de ancho y largo. El tejín (con una estrella en su interior) se encuentra en la posición $(1, 2)$. Las distancias de los lanzamientos al tejín son $1, 3, 2$, y 5, donde los primeros dos lanzamientos corresponden al equipo `A` y los últimos 2 al equipo `R`. El equipo `A` resulta ganador y recibe solamente 1 punto (corresponde al tejo que está a 1 centímetro de distancia al tejín).

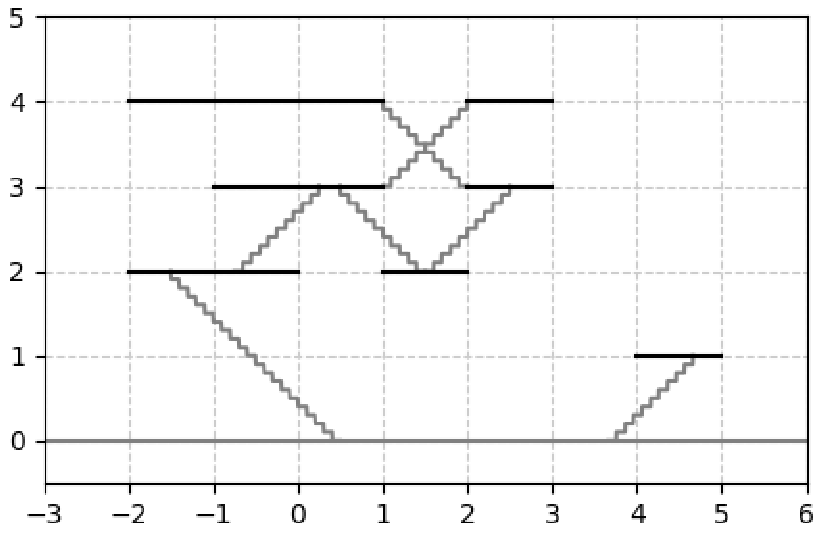
Problema G
Grandes Edificaciones

En Nlogonia, la construcción de la tan esperada Torre Alta y Prestigiosa (TAP) finalmente ha comenzado. Los andamios que serán utilizados por los trabajadores en las diversas operaciones requeridas para llevar a cabo esta obra monumental ya se encuentran instalados y listos para su uso. Cada andamio se puede modelar como un segmento horizontal (es decir, alineado con el eje x). El i-ésimo andamio se ubica a altura H_i sobre el suelo, y abarca las coordenadas sobre el eje x entre L_i y R_i inclusive.

Sin embargo, aún no se ha determinado dónde irán las escaleras que servirán para el acceso de los operarios y transporte de los materiales. Por dicho motivo, nos han contratado para que llevemos a cabo esta importante tarea. Cada escalera se puede modelar como un segmento con un extremo superior y otro inferior. Llamamos base de una escalera a su extremo inferior. Las bases de las escaleras pueden estar apoyadas sobre el suelo o sobre un andamio, y sus extremos superiores deben estar apoyados sobre otro andamio.

Además, para que sean funcionales, las escaleras deben construirse de manera tal que sea posible acceder desde el suelo (modelado por el eje x, a altura 0) hasta cualquiera de los andamios, subiendo y bajando cuantas escaleras sean necesarias para ello. Dado que las escaleras deben usarse para transportar objetos pesados, se decidió que se instalen a 45°, de manera tal que si una escalera tiene el extremo superior a una altura D metros por encima de su base, su extremo superior tendrá una coordenada x que estará D metros a la izquierda o a la derecha de la coordenada x de su base.

Es posible instalar las escaleras de manera que sus segmentos se intersequen, pero por razones de seguridad está estrictamente prohibido pasar de una escalera a otra por fuera de un andamio. Es decir, una escalera solamente conecta entre sí los andamios en que se apoyan sus extremos (o el suelo con un andamio, en el caso de las escaleras con base en el suelo).



Como la empresa quiere ahorrar gastos, nos han pedido que determinemos el mínimo costo total de instalar todas las escaleras necesarias, teniendo en cuenta que el costo de una escalera es igual a la diferencia entre la altura de su extremo superior y la altura de su base.

Entrada

La primera línea contiene un solo entero N ($1 \leq N \leq 10^5$), la cantidad de andamios instalados en la TAP. Las siguientes N líneas contienen 3 enteros H_i, L_i y R_i ($1 \leq H_i \leq 10^9$, $-10^9 \leq L_i < R_i \leq 10^9$), siendo H_i la altura del i-ésimo andamio, mientras que L_i y R_i son la posición de los extremos izquierdo y derecho del mismo. Se garantiza que los andamios no comparten ningun punto.

Salida

Una sola línea con un entero igual al mínimo costo total de instalar todas las escaleras.

Ejemplo de entrada 1	**Ejemplo de salida 1**

`7`
`2 -2 0`
`3 -1 1`
`3 2 3`
`4 -2 1`
`4 2 3`
`2 1 2`
`1 4 5`
	`8`

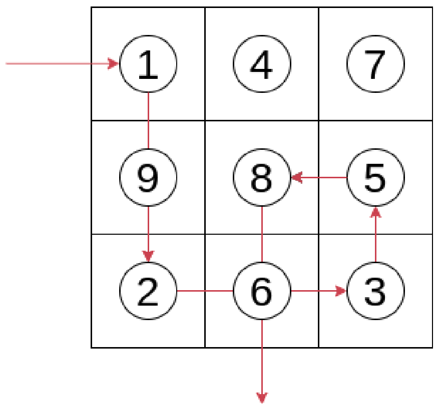
Una manera de obtener el resultado del ejemplo puede encontrarse en la imagen del enunciado, utilizando 6 escaleras de costo 1, y una escalera de costo 2. Notar que en dicha solución, las escaleras con base en los andamios ubicados a altura 3 no están conectadas entre sí.

Problema H

Habilidades robóticas

Franco está aprendiendo robótica y está muy contento porque ya construyó su primer robot. Su plan original era que este robot fuera capaz de jugar a la Rayuela, pero como esto es muy difícil, se conforma con una versión más simple de este juego.

Sobre el piso hay dibujado un tablero de $N \times N$ casillas. Cada una de estas casillas contiene un número entero entre 1 y N^2. Se sabe además que no hay números repetidos. El robot puede arrancar su recorrido en cualquier punto exterior al tablero y solamente puede moverse horizontalmente y verticalmente, es decir, en las direcciones paralelas a los bordes del tablero. Para cambiar su dirección de movimiento de vertical a horizontal (o viceversa) el robot debe estar en el interior de una de las casillas. El robot puede cambiar su dirección una vez como máximo en cada casilla. En ningún momento puede cambiar su dirección en 180°. El recorrido debe terminar en el exterior del tablero. Sean $c_1, c_2, c_3, \ldots, c_k$ los números escritos en las casillas en las que el robot cambió su dirección. Para que el recorrido sea válido, se debe cumplir que $c_1 < c_2 < c_3 < \ldots < c_k$. El puntaje de un recorrido es la cantidad de casillas en las que el robot cambió su dirección. A continuación se muestra un recorrido válido de un tablero de 3×3:



Ayudá a Franco a programar el robot de forma que el puntaje de su recorrido sea lo máximo posible.

Entrada

Una primera línea con un entero N ($2 \leq N \leq 1000$). Luego, la i-esima de las siguientes N líneas contiene N números enteros $A_{i,1}, A_{i,2}, \ldots, A_{i,N}$, que son los números escritos en la i-esima fila del tablero ($1 \leq A_{i,j} \leq N^2$). Se garantiza que todos los números son diferentes.

Salida

Una única línea con el mayor puntaje que puede conseguir el robot en su recorrido.

Ejemplo de entrada 1	Ejemplo de salida 1
2 1 2 3 4	3

Ejemplo de entrada 2	**Ejemplo de salida 2**

`3`
`1 4 7`
`9 8 5`
`2 6 3`
	`5`

En el segundo ejemplo, un posible recorrido del robot que consigue el puntaje máximo puede encontrarse en la imagen del enunciado. Observar que el robot no cambia de dirección en las casillas con los números 6 y 9.

Otro posible recorrido que también consigue el puntaje máximo se obtiene cambiando de dirección en las casillas $1, 2, 3, 5$ y 9.

Problema I
Integración regional

La lejana y moderna ciudad de Extremópolis se encuentra justo en la triple frontera entre las naciones de Circulonia, Cuadradonia y Triangulonia.

Por esta razón es que en la ciudad existen muchos edificios. Estos edificios son de solamente 3 tipos posibles:

1. La base del edificio tiene forma de círculo
2. La base del edificio tiene forma de cuadrado (**no necesariamente alineado a los ejes**). Un cuadrado es un cuadrilátero con los 4 lados iguales y los 4 ángulos rectos.
3. La base del edificio tiene forma de triángulo (no necesariamente alineado a los ejes)

Poseés un plano detallado de la ciudad, en el cual se pueden apreciar los círculos, cuadrados y triángulos de las bases de los distintos edificios. Dos de estas figuras nunca se intersecan ni se tocan, ni siquiera en un vértice, ya que son edificios diferentes.

El Instituto de Coordinación y Planificación de Circulonia (ICPC) planea abrir dos nuevas oficinas de negocios: una deberá estar ubicada en un edificio de base triangular, y la otra en un edificio de base cuadrada. Con esto el instituto busca mejorar sus relaciones comerciales con los países vecinos.

Otra propiedad extrema de Extremópolis es que se ubica sobre el ecuador, posee un clima muy seco y prácticamente sin días nublados. Por lo tanto el nivel de radiación ultravioleta del sol que experimentan sus habitantes es extremadamente alto. Teniendo esto en cuenta, el ICPC desea seleccionar los dos edificios donde abrirá las nuevas oficinas de manera tal que para viajar caminando de una a la otra, sea necesario caminar al sol lo menos posible.

Extremópolis no tiene calles ni obstáculos que obstruyan el paso, por lo cual es posible caminar por cualquier punto de la ciudad libremente. Todos los edificios están diseñados con una planta baja espaciosa y sin obstáculos, por lo que incluso se puede caminar libremente por toda la base de los edificios (es decir, por todo el interior del círculo, del cuadrado, o del triángulo respectivo), y estos tramos de caminata por dentro de los edificios no cuentan para el cómputo de la distancia total caminada bajo el sol.

Tu tarea consiste en computar la mínima distancia que es necesario caminar al sol para viajar entre ambas oficinas, si se abren en los edificios ideales para minimizar esa distancia.

Entrada

Una primera línea con tres enteros C, Q, T: la cantidad de edificios con base en forma de círculo ($0 \leq C$), con forma de cuadrado ($1 \leq Q \leq 10^5$) y con forma de triángulo ($1 \leq T \leq 10^5$) respectivamente.

Además se sabe que $(T + C)(Q + C) \leq 10^6$.

Cada una de las siguientes C líneas describe un edificio con base en forma de círculo mediante 3 enteros x, y, r. El correspondiente círculo tiene centro en (x, y) y radio r.

Cada una de las siguientes Q líneas describe un edificio con base en forma de cuadrado mediante 4 enteros x_1, y_1, x_2, y_2. El correspondiente cuadrado tiene un par de vértices **opuestos** en (x_1, y_1) y en (x_2, y_2). Se garantiza que son dos puntos distintos del plano, es decir $(x_1, y_1) \neq (x_2, y_2)$.

Cada una de las siguientes T líneas describe un edificio con base en forma de triángulo mediante 6 enteros $x_1, y_1, x_2, y_2, x_3, y_3$. El correspondiente triángulo tiene sus tres vértices en los puntos (x_1, y_1), (x_2, y_2) y (x_3, y_3). Se garantiza que son tres puntos distintos del plano, y que no están alineados (el triángulo no es degenerado).

Los radios r y todas las coordenadas x, y son números enteros entre 1 y 10^9 inclusive.

Se garantiza que no habrá dos edificios cuyas bases se intersequen, ni siquiera en un punto en su frontera.

Salida

Una única línea con un único número: la mínima distancia que es posible caminar al sol cumpliendo lo pedido.

Esta respuesta será aceptada si tiene un error relativo o absoluto menor o igual a 10^{-6}.

Formalmente, sea a tu respuesta y b la respuesta del jurado. Entonces, tu respuesta será aceptada si y solo si $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Ejemplo de entrada 1	Ejemplo de salida 1

`2 1 1`
`10 1 5`
`20 1 4`
`1 1 4 4`
`24 5 28 50 45 4`
	`3.656854249`

Ejemplo de entrada 2	Ejemplo de salida 2

`0 1 1`
`1 1 4 4`
`5 1 5 10 10 1`
	`1.0`

Ejemplo de entrada 3	Ejemplo de salida 3

`0 2 2`
`1 1 10 2`
`3 12 2 20`
`50 10 50 100 100 10`
`450 410 450 4100 4100 410`
	`40.792156108`

En los primeros dos ejemplos hay un único edificio de base cuadrada, y un único edificio de base triangular, por lo que esos deben ser elegidos necesariamente para abrir las oficinas.

En el primer ejemplo, para caminar al sol la menor distancia posible, es conveniente pasar en el camino por los edificios con base circular.

Notar que en el tercer ejemplo los cuadrados no están alineados a los ejes.

Problema J

Juglar en peligro

En el reino de Grafonia existen N ciudades, que se identifican con enteros consecutivos desde 1 hasta N inclusive. Grafonia es un reino muy peculiar pues tiene dos capitales: La ciudad 1 es la capital real de Grafonia, y la ciudad N es la capital constitucional.

En Grafonia existen también M carreteras bidireccionales. Cada carretera conecta en forma directa exactamente dos ciudades diferentes. La carretera i-ésima conecta en forma directa las ciudades A_i y B_i.

Llamamos G_0 al mapa de Grafonia, con sus N ciudades y sus M carreteras. Con el fin de preparar planes de contingencia especiales y aumentar la seguridad del reino, el rey está interesado en estudiar lo que pasaría si el reino perdiera ciudades sucesivamente como consecuencia de posibles catástrofes. En el escenario hipotético que el rey desea estudiar, serán eliminadas K ciudades en orden una tras otra: $C_1, C_2, \ldots, C_K$. Comenzando con el mapa G_0, tras eliminar cada una de ellas, se obtienen los nuevos mapas $G_1, G_2, \ldots, G_K$ respectivamente. Es decir, G_i se obtiene eliminando la ciudad C_i (junto con cualquier carretera que incida en esa ciudad) del mapa G_{i-1}, para cada $1 \leq i \leq K$. Ninguna de las ciudades eliminadas será capital.

Para cada $0 \leq i \leq K$, un camino capitalino en G_i es una sucesión de ciudades del mapa G_i que comienza en la ciudad 1, termina en la ciudad N, y de modo tal que ciudades consecutivas en la sucesión están conectadas en forma directa por una carretera. La longitud de un camino capitalino es la cantidad de ciudades en la sucesión. Si existe un camino capitalino en G_i, el rey define $D(i)$ como la longitud mínima entre todos los caminos capitalinos en G_i. Se sabe que en G_0 existe un camino capitalino, por lo que $D(0)$ está definido.

Para cada $0 \leq i \leq K$, el rey dice que G_i está roto si no existe camino capitalino en G_i, o bien si existe pero $D(i) > D(0)$, es decir, la longitud mínima de un camino capitalino en el mapa G_i es estrictamente mayor que en G_0.

Para cada $0 \leq i \leq K$ tal que G_i no está roto, el rey dice que una ciudad v ($2 \leq v \leq N-1$) del mapa G_i es crítica, si el mapa que se obtendría al eliminar v de G_i estaría roto.

Eres un juglar en la corte del rey, y para tu desgracia también eres la única persona que sabe programar en todo el reino. Para evitar el castigo del implacable rey grafonio, debes escribir un programa que para cada $0 \leq i \leq K$ determine la cantidad de ciudades críticas en el mapa G_i, o bien indique que G_i está roto.

Entrada

Una primera línea con tres enteros N, M, K ($3 \leq N \leq 10^5$, $1 \leq M \leq 2 \cdot 10^5$, $1 \leq K \leq N-2$).

Luego M líneas que describen las carreteras. La i-ésima línea contiene los dos enteros A_i y B_i ($1 \leq A_i < B_i \leq N$). Estas M líneas son todas distintas entre sí.

Luego K líneas con las ciudades a eliminar, en el orden en que se van eliminando. Es decir, la i-ésima línea contiene el entero C_i ($2 \leq C_i \leq N-1$). Estas K ciudades son todas distintas entre sí.

Salida

Una única línea con $K+1$ enteros, que indican lo pedido por el rey para cada mapa.

Más precisamente, para cada $0 \leq i \leq K$, el i-ésimo entero (contando desde 0) deberá ser:

- -1, si el mapa G_i está roto
- La cantidad de ciudades críticas en el mapa G_i, si el mapa G_i no está roto

Ejemplo de entrada 1	**Ejemplo de salida 1**

`4 5 2`
`1 2`
`2 4`
`2 3`
`1 3`
`3 4`
`3`
`2`
	`0 1 -1`

Ejemplo de entrada 2	**Ejemplo de salida 2**

`6 5 2`
`1 2`
`2 3`
`2 5`
`3 5`
`3 6`
`4`
`5`
	`2 2 2`

Ejemplo de entrada 3	**Ejemplo de salida 3**

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

1. En el primer ejemplo, los caminos capitalinos de menor longitud en G_0 son $1 \to 2 \to 4$ y $1 \to 3 \to 4$. Si se eliminara una ciudad no capital, ya sea la 2 o la 3, seguiría quedando al menos un camino capitalino con esa mínima longitud, por lo tanto en G_0 no hay ciudades críticas. En G_1 ya se ha eliminado la ciudad 3, por lo que si en G_1 se eliminara la ciudad 2 ya no quedaría ningún camino entre capitales y quedaría un mapa roto. Por esta razón, la ciudad 3 es crítica en G_1, y hay exactamente una ciudad crítica en G_1. Finalmente, en G_2 ya se han eliminado ambas ciudades (la 2 y la 3) y no queda ningún camino capitalino en G_2, que está roto.

2. En el segundo ejemplo, 2 y 3 son las únicas ciudades críticas en G_0, G_1 y G_2.

3. En el tercer ejemplo, 2 es la única ciudad crítica de G_0, y G_1, G_2, G_3 están rotos. Notar que en G_1 existe un camino capitalino $1 \to 3 \to 4 \to 5$, pero su longitud (y la de cualquier otro) es mayor que la del camino capitalino de mínima longitud en G_0, que es $1 \to 2 \to 5$.

Problema K
Kitties

Marcos es veterinario y los gatos son su animal favorito. Tanto es así, que desde hace unos años que tiene un local comercial llamado Kitties, donde vende todo tipo de cosas para gatos. En su local, Marcos tiene una pantalla de alto H_1 y ancho W que utiliza para mostrar un sitio web que él mismo programó, con distintas fotos de gatitos que se van moviendo y renovando con el tiempo.

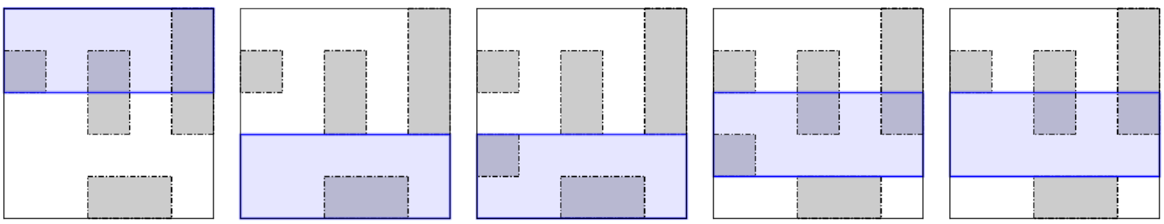
Este sitio web consiste de un panel verticalmente desplazable (o "scrolleable"), que tiene el mismo ancho W que la pantalla, pero tiene una mayor altura $H_2 > H_1$ (por eso es verticalmente desplazable).

Inicialmente, el panel contiene N fotos de gatitos ubicadas en distintas posiciones. Más adelante aparecerán K nuevas fotos, de modo que existen $N + K$ posibles fotos distintas en total. La i-ésima de estas fotos se identifica con el entero positivo i, llamado su id. Las fotos con id desde 1 hasta N son las iniciales, y las fotos con ids $N + 1$ hasta $N + K$ serán agregadas más adelante. Para cada i desde 1 hasta $N + K$, la foto con id i abarca una porción rectangular del panel definida por los siguientes valores:

- U_i: la distancia desde el borde **superior** del panel al borde **superior** de la foto
- D_i: la distancia desde el borde **superior** del panel al borde **inferior** de la foto
- L_i: la distancia desde el borde **izquierdo** del panel al borde **izquierdo** de la foto
- R_i: la distancia desde el borde **izquierdo** del panel al borde **derecho** de la foto

Dado que el panel es más alto que la pantalla, en todo momento un subrectángulo del mismo será visible. De manera similar a las fotos, dicho subrectángulo se define por los valores s_U, s_D, s_L y s_R.

Inicialmente, el *desplazamiento* del sitio web es $x = 0$, es decir que en la pantalla de Marcos estará visible el subrectángulo del panel dado por $s_U = 0$, $s_D = H_1$, $s_L = 0$, $s_R = W$. Sin embargo, el sitio web está programado para que cada tanto, algunas fotos nuevas aparezcan, otras desaparezcan y el desplazamiento cambie.



Marcos tiene miedo de que estos cambios programados para que ocurran automáticamente afecten la eficiencia del sitio web, dado que las fotos se van cargando y quitando de memoria según sean visibles o no. Y por eso necesita tu ayuda.

Dados Q cambios que ocurren en el sitio web en orden, para cada uno de ellos tenés que responder las siguientes dos preguntas:

1. ¿Cuántas fotos que **no** eran visibles (parcial o totalmente) antes de este cambio, **sí** lo serán luego del cambio?
2. ¿Cuántas fotos que **sí** eran visibles (parcial o totalmente) antes de este cambio, **no** lo serán luego del cambio?

Entrada

Una primera línea con los cinco enteros N, Q, H_1, H_2 y W ($1 \leq N, Q \leq 10^5$, $1 \leq H_1, H_2, W \leq 10^9$, $H_1 < H_2$) explicados en el enunciado.

Luego siguen N líneas. La i-ésima de estas líneas contiene cuatro enteros U_i, D_i, L_i y R_i, que describen la ubicación en el panel de la foto con id i.

Luego siguen Q líneas que representan los cambios que ocurren en el sitio web en orden. Cada una de estas Q líneas tiene un caracter seguido por una cantidad de números enteros, respetando alguna de las siguientes formas:

- `A` j U_j D_j L_j R_j: indicando que aparece una nueva foto con id j, ubicada en el panel según indican U_j, D_j, L_j y R_j. El i-ésimo cambio de este tipo en la entrada tendrá $j = N + i$, y habrá K cambios de este tipo en total.

- `D` j ($1 \leq j \leq N + K$): indicando que la foto con id j desaparece (se garantiza que dicha foto se encuentra presente en el panel).

- `M` x ($0 \leq x \leq H_2 - H_1$): indicando que el desplazamiento es modificado a tomar el valor x, es decir que en la pantalla se visualizará el subrectángulo del panel definido por $s_U = x$, $s_D = x + H_1$, $s_L = 0$, $s_R = W$

Se garantiza que ningún par de fotos se solapan ni se tocan, y para todo i desde 1 hasta $N + K$, se cumple $0 \leq U_i < D_i \leq H_2$, $0 \leq L_i < R_i \leq W$.

Salida

Q líneas, una por cada cambio dado en la entrada, con dos enteros respondiendo a las dos preguntas planteadas en el enunciado, en el orden que dichas preguntas fueron hechas.

Ejemplo de entrada 1	**Ejemplo de salida 1**

`4 4 2 5 5`
`1 2 0 1`
`1 3 2 3`
`0 3 4 5`
`4 5 2 4`
`M 3`
`A 5 3 4 0 1`
`M 2`
`D 5`
	`1 3`
	`1 0`
	`2 1`
	`0 1`

Las cinco imágenes del enunciado reflejan la configuración inicial del ejemplo y la configuración posterior a cada uno de los cambios, en orden. Notar que el rectángulo azul en cada imágen representa el subrectángulo del panel visible en la pantalla, mientras que los rectángulos grises corresponden a las fotos.

Los ids de las fotos visibles en cada uno de estos cinco momentos son: $[1, 2, 3]$, $[4]$, $[4, 5]$, $[2, 3, 5]$ y $[2, 3]$

Problema L

Local y visitante

Dos equipos, "Arquimedianos F.C." y "Pitagóricos F.C.", disputan la final de un torneo de fútbol jugando una **serie a dos partidos**. Cada partido finaliza con una cierta cantidad de goles realizados por cada equipo.

Al cabo de estos dos partidos, un equipo es ganador de la serie si realizó **en total** más goles que el rival. En caso contrario, si finalizan con la misma cantidad de goles al cabo de ambos partidos, la serie se debe definir en un tercer partido de desempate.

Conociendo los resultados de cada uno de estos dos partidos (es decir, cuántos goles realizó cada equipo en cada partido), se te pide que escribas un programa que determine qué equipo ganó la serie, o si la serie debe ir a un partido de desempate.

Entrada

Una primera línea con dos enteros $A_1, P_1 (0 \leq A_1, P_1 \leq 31)$, que denotan la cantidad de goles que realizaron "Arquimedianos F.C." y "Pitagóricos F.C." en el primer partido, respectivamente.

Luego, una segunda línea similar a la primera con dos enteros $A_2, P_2 (0 \leq A_2, P_2 \leq 31)$, la cantidad de goles que realizaron en el segundo partido "Arquimedianos F.C." y "Pitagóricos F.C." respectivamente.

Salida

Una única línea con un único caracter. En caso de que el equipo "Arquimedianos F.C." resulte ganador de la serie, se deberá escribir `A`. Si el equipo que resulta ganador en la serie es el equipo "Pitagóricos F.C." debes imprimir `P`, y si se requiere de un partido desempate para definir la serie debes imprimir el caracter `D`.

Ejemplo de entrada 1	**Ejemplo de salida 1**
`3 1` `1 1`	`A`

Ejemplo de entrada 2	**Ejemplo de salida 2**
`4 3` `1 3`	`P`

Ejemplo de entrada 3	**Ejemplo de salida 3**
`2 4` `2 0`	`D`

En el primer ejemplo "Arquimedianos F.C." realizó 3 goles en el primer partido y 1 gol en el segundo partido, totalizando 4 goles en la serie. Por otro lado "Pitagóricos F.C." realizó 1 gol en ambos partidos, totalizando 2 goles en la serie. Por lo tanto "Arquimedianos F.C." resultó el equipo ganador de la serie y la respuesta es `A`.

Problema M

Múltiples descargas

CDescarga es un gestor de descargas escrito en lenguaje C. Es decir, un programa capaz de automáticamente descargar de internet un listado de N archivos provistos por el usuario, todos a la vez.

Cada archivo que se va a descargar puede estar priorizado o no. A diferencia de otros gestores mucho más complejos y con reglas más justas y mejores, la regla que utiliza CDescarga es muy simple: en cada momento, de la velocidad de descarga total de T megabytes por segundo (MB/s) que tiene la conexión del usuario, utilizará el 75% de esa velocidad para descargar los archivos priorizados, y el 25% para descargar los archivos no priorizados. Dentro de cada grupo, la velocidad de descarga asignada se reparte por igual.

Por ejemplo, si CDescarga debe descargar los archivos:

- A, priorizado, de tamaño 10 MB
- B, no priorizado, de tamaño 100 MB
- C, priorizado, de tamaño 20 MB
- D, priorizado, de tamaño 200 MB
- E, no priorizado, de tamaño 300 MB

Y la velocidad de la conexión del usuario es 40 MB/s, entonces inicialmente CDescarga:

- Descarga A, con velocidad 10 MB/s
- Descarga B, con velocidad 5 MB/s
- Descarga C, con velocidad 10 MB/s
- Descarga D, con velocidad 10 MB/s
- Descarga E, con velocidad 5 MB/s

Ya que de los 40 MB/s, el 75% es 30 MB/s y eso se utiliza para los 3 archivos priorizados, descargando así cada uno a 10 MB/s. Mientras que el 25% de 40 MB/s es 10 MB/s, y eso se utiliza en total para los 2 archivos no priorizados, resultando en 5 MB/s para cada uno.

Tras 1 segundo de descargar, se completa la descarga del archivo A, y la nueva situación de los que quedan es:

- Descarga B (ya van 5 MB descargados), con velocidad 5 MB/s
- Descarga C (ya van 10 MB descargados), con velocidad 15 MB/s
- Descarga D (ya van 10 MB descargados), con velocidad 15 MB/s
- Descarga E (ya van 5 MB descargados), con velocidad 5 MB/s

Y así se sigue hasta completar todas las descargas. Si en algún momento hay solamente archivos priorizados, o solamente archivos no priorizados, entonces la velocidad de descarga se reparte por igual entre todos los archivos restantes.

Tu tarea consiste en escribir un programa que calcule el tiempo total que utilizará CDescarga para descargar de internet todos los archivos.

Entrada

Una primera línea con dos enteros N y T ($1 \leq N \leq 50$, $1 \leq T \leq 1024$), la cantidad de archivos a descargar y la velocidad de descarga de la conexión del usuario en MB/s.

Luego N líneas, cada una de las cuales describe uno de los N archivos. La i-ésima línea contiene una letra mayúscula C_i ($C_i \in \{\texttt{P}, \texttt{N}\}$) y un entero X_i ($1 \leq X_i \leq 512$), que indican respectivamente si el i-ésimo archivo es priorizado (letra `P`) o no priorizado (letra `N`), y el tamaño en MB.

Salida

Una única línea con un único número que indica el tiempo total en segundos hasta que CDescarga complete la descarga de los N archivos.

Esta respuesta será aceptada si tiene un error relativo o absoluto menor o igual a 10^{-4}.

Formalmente, sea a tu respuesta y b la respuesta del jurado. Entonces, tu respuesta será aceptada si y solo si $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-4}$.

Ejemplo de entrada 1	**Ejemplo de salida 1**

`5 40`
`P 10`
`N 100`
`P 20`
`P 200`
`N 300`
	`15.75`

Ejemplo de entrada 2	**Ejemplo de salida 2**
`1 1` `N 64`	`64`

Ejemplo de entrada 3	**Ejemplo de salida 3**
`1 113` `P 355`	`3.1415929204`

Problema N

Número de la suerte

Matías tiene un mazo de N cartas. Cada una tiene escrito un número entero en ella. Como su número de la suerte es el cinco, quiere formar varios grupos con estas cartas de forma que en cada uno de estos grupos la suma de los números de las cartas que lo conforman sea múltiplo de cinco. Ninguna carta puede estar en dos o más grupos. Puede ocurrir que alguna carta no pertenezca a ningún grupo. ¿Cuál es la mayor cantidad de grupos que puede formar Matías?

Entrada

Una línea con un entero N ($1 \leq N \leq 2 \cdot 10^5$), la cantidad de cartas del mazo de Matías.

Luego, una segunda línea con N enteros $C_1, C_2, \ldots, C_N$. Cada uno indica el número escrito en la i-ésima carta ($1 \leq C_i \leq 10^9$).

Salida

Una única línea con un único entero, que representa la mayor cantidad de grupos de cartas que Matías puede formar.

Ejemplo de entrada 1	**Ejemplo de salida 1**
`6` `33 21 66 8 1 108`	`1`

Ejemplo de entrada 2	**Ejemplo de salida 2**
`9` `1 6 7 10 16 18 41 42 77`	`3`

Ejemplo de entrada 3	**Ejemplo de salida 3**
`2` `19 7`	`0`

En el primer ejemplo, Matías puede formar un grupo con las siguientes cuatro cartas: $33, 21, 8$ y 108. Su suma es $33 + 21 + 8 + 108 = 170$ que es múltiplo de cinco. No pueden formarse dos grupos de cartas cuya suma sea múltiplo de cinco.

En el segundo ejemplo, Matías puede formar tres grupos de la siguiente forma:

- El primer gurpo contiene las cartas $1, 6, 41$ y 77.
- El segundo grupo contiene las cartas 7 y 18.
- El tercer grupo contiene la carta 10.