

Coding Cup TecNM — A. Mensajes ocultos

Points	100	Memory limit	32 MiB
Time limit (case)	1s	Time limit (total)	1m0s

Descripción

En la materia de Fundamentos de programación del TecNM, un profe les dejó una tarea clásica, imprimir pirámides, triángulos y cuadrados de asteriscos. Para hacerlos sufrir un rato más, les pidió que descubrieran lo que les pide a partir de un mensaje cifrado.

El mensaje se compone de muchas líneas de caracteres del abecedario inglés en mayúsculas, no hay números ni caracteres especiales en este mensaje cifrado. Cada 16 letras, iniciando en la letra 16, está un carácter que forma parte del mensaje oculto.

Debes escribir un programa que lea un número N de líneas, donde $3 \leq N \leq 20$. Cada línea se compone de entre 1 y 100 caracteres. Cada 16 caracteres se encuentra una letra que debes tomar para formar el mensaje, excepto si la letra es una "X" y además se encuentra precedida por una letra "I" o por una letra "O".

Al finalizar, la palabra que se obtiene indica si debes imprimir un TRIANGULO, un CUADRADO o una PIRAMIDE (así, sin acentos). Después de la palabra vendrá una letra entre la A y la Z, esta letra indica la altura de la figura, por ejemplo, si el mensaje obtenido es "TRIANGULOD", entonces te están pidiendo pintar un TRIANGULO de tamaño 4. (A=1, B=2, C=3, D=4 etc).

Entonces, tu tarea solo es descubrir el mensaje y pintar una figura como la que se te pide con asteriscos y del tamaño indicado. Si el mensaje es diferente a alguna de las tres figuras, entonces solo debes imprimir el mensaje obtenido. Observa los ejemplos de entrada y salida.

Entrada

Un entero N que indica el número de líneas a leer, seguido de N líneas de texto de entre 1 y 100 caracteres.

Salida

Imprime un triángulo, rectángulo, o pirámide de la altura indicada en el mensaje. Si no se solicita una figura, entonces solo se imprime el mensaje.

Ejemplo

Entrada	Salida
7 AFDEERADARLFJOATFASFASFAYOTPKKJRVFDESQPPZAXSCDRICJIK HHYJUTFREDDADNMJJ LOKKIJUDDNXMQP LMJUTYHFGFFGD JJJJK OLIKIKIHURFDODLJLPROTTJLLRJTOIOEUTEOTEOTEORTUWELSDJOTEJ DCMCJDJDUIEKFXZOXDSC	* ** *** ****
7 DUISNISLTORTORTCMPOREGETPHARETRUUTAUCTORTRISTIDAEESTCLASSAPTENT DACITISOCIOSQUADRITORATORQUENTPEAC ONUBIANOSTRAPEDINCEPTOSHIMENAEOSPRAESE NTCOMMOIJULTRICESRUTRUMIXNECACPHARETRA DIXSNISLTORTOR TEMIXREGETPHARETRAUOX UCTORTRISTIQUEOXTCLASSAPTENT	***** ***** ***** ***** ***** ***** ***** ***** *****
10 ABCDEFGHJKLMNOP ABCDEFGHJKLMNOI ABCDEFGHJKLMNOR ABCDEFGHJKLMNOA ABCDEFGHJKLMNOM ABCDEFGHJKLMNOI ABCDEFGHJKLMNOD ABCDEFGHJKLMNOE ABCDEFGHJKLMNEE ABCDEFGHJKLMNOX	* *** ***** ***** *****
4 DUISNISLTORTORTIMPOREGETPHARETRTUTAUCTORTRISTIDSEESTCLASSAPTENT UACITISOCIOSQUADRITORATORQUENTPIXC ONUBIANOSTRAPOXINCEPTOSHIMENAOXSPRAESE NTCOMMOIXULTRICESRUTRUMOXNECACPHARETRA	ITSUR

Coding Cup TecNM — B. La fiesta de Filiberto

Points	100	Memory limit	256 MiB
Time limit (case)	5s	Time limit (total)	1m0s

Descripción

Este año Filiberto ofreció una fiesta debido a su cumpleaños en junio. Esta fiesta estaba pensada en tener un formato diferente a las fiestas que había organizado anteriormente. En este evento había mucha gente que quería jugar videojuegos que solo funcionan cuando se conectan a una red WIFI.

El modem de la conexión WIFI tiene una distancia límite, que significa que el internet estará disponible a lo más a una distancia d del modem (hacia cualquier dirección). Hubo algunos problemas con la conectividad durante la fiesta por lo cual Filiberto necesita de tu ayuda para solucionar el siguiente problema.

El organizara de nuevo una fiesta el siguiente año, pero quiere evitar a toda costa este tipo de inconvenientes. La fiesta se llevará a cabo en un apartamento de tamaño $n \times m$ (el departamento está representado como una matriz de este tamaño). Para poder jugar cualquier videojuego, cada persona debe de estar sentada en un punto p dentro del apartamento, el cual es conocido desde antes de organizar la fiesta.

Filiberto no sabe que personas irán a la fiesta por lo cual él te realizará distintas preguntas de la siguiente forma:

$k, u_1, u_2, u_3, \dots, u_k$ - lo cual representa una pregunta que inicia con un número k , significando que él va a preguntarte por k personas y los identificadores de cada una de ellas (los identificadores van de 1 a l).

Después de cada pregunta debes imprimir la mínima distancia d que el modem debe de soportar para que todas las personas incluidas en la pregunta actual puedan jugar cualquier juego sin ningún inconveniente.

El modem puede ser colocado en cualquier lugar dentro del departamento (incluso en puntos con coordenadas reales). El modem también se puede poner en la misma posición en donde una persona esta colocada.

Entrada

La primera línea de entrada contiene tres enteros n, m y l ($1 \leq n, m \leq 10^9, 1 \leq l \leq 18$) - representando el tamaño del apartamento y el máximo número de personas que podrían ir a la fiesta.

Las siguientes l líneas contienen dos enteros x_i y y_i ($1 \leq x_i \leq n, 1 \leq y_i \leq m$) - indicando el punto donde la i th persona estaría colocada dentro del apartamento.

La siguiente línea contiene un numero q ($1 \leq q \leq 3 * 10^5$) -el número de preguntas que Filiberto va a realizar.

Las siguientes q líneas contienen las preguntas con el siguiente formato:

$k, u_1, u_2, u_3, \dots, u_k$ ($1 \leq k \leq l, 1 \leq u_i \leq l \quad \forall i \mid 1 \leq i \leq k$) - lo cual representa una pregunta con un número k y que significa que él te preguntara por k personas y enseguida los identificadores de cada una de ellas. El identificador de una persona puede aparecer a lo más una vez en cada pregunta.

Salida

Por cada pregunta tienes que imprimir la mínima distancia d que el modem de la conexión WIFI debe soportar para que las k personas puedan jugar sin problemas considerando que puedes poner el modem en cualquier punto dentro del apartamento.

Tu respuesta será considerada correcta si el absoluto o error relativo no excede 10^{-9} .

Formalmente, sea tu respuesta a , y la respuesta del juez b . Tu respuesta será considerada correcta si y solo si $\frac{|a - b|}{\max(1, |b|)} \leq 10^{-9}$.

Ejemplos

Entrada	Salida
10 10 4	0.000000000
10 5	3.041381265
3 6	3.535533906
9 7	1.118033989
4 4	3.535533906
5	
1 4	
2 1 4	
3 1 2 3	
2 2 4	
3 1 2 4	
10 10 4	4.031128874
5 7	1.118033989
3 4	0.000000000
4 2	4.031128874
5 10	2.549509757
5	
4 4 2 1 3	
2 2 3	
1 2	
2 3 4	
2 3 1	

Coding Cup TecNM — C. Carrera Conmemorativa

Points	100	Memory limit	512 MiB
Time limit (case)	4s	Time limit (total)	1m0s

Descripción

Filiberto está planeando una carrera para celebrar el concurso "Coding Cup TecNM". En esta carrera, los participantes compiten yendo cuesta abajo sobre una montaña manejando un automóvil sobre un conjunto de carreteras disponibles. Como podría ser peligroso que existan autos yendo en direcciones opuestas, es por eso que los concursantes solo pueden conducir cuesta abajo en cada una de las carreteras y en una sola dirección. El extremo de una carretera es llamado una estación, y las carreteras solo pueden conectarse en dichas estaciones.

Un camino es una secuencia de carreteras donde la estación final de una carretera es la estación de inicio de la siguiente. Cada corredor debe escoger un camino antes de la carrera por el cual deberá seguir su trayectoria durante la competencia. Se dice que un corredor ha finalizado su carrera, cuando no tiene más carreteras por donde circular. La longitud de un camino es el número de carreteras en él.

El objetivo de esta carrera es permanecer en la competencia lo máximo posible -- esto es, encontrar un camino de longitud máxima. Cada corredor conoce el diseño completo de las estaciones y carreteras, y elegirá algún camino de longitud máxima (**puede elegir cualquier camino existente de máxima longitud**).

Filiberto está preocupado por desastres naturales (por ejemplo, avalanchas de tierra) que puedan bloquear las carreteras durante la competencia. Si una carretera está bloqueada, esto puede reducir la longitud de algunos caminos existentes. Si un competidor está siguiendo su trayectoria y se encuentra que la siguiente carretera que debe tomar está bloqueada, entonces deberá tomar una carretera alterna que maximice su estancia en la competencia.

Esto funciona de la siguiente manera: digamos que la carretera que va de la estación u a la estación v está bloqueada. Entonces el corredor seguirá su camino hasta la estación u , donde tendrá que modificar su trayectoria desde la estación u y tratar de seguir maximizando la longitud de su nuevo camino. Como cada competidor conoce el mapa completo de la montaña, ellos elegirán su nueva ruta de manera óptima para así maximizar la longitud de su nuevo camino.

Para entender el peor de los casos, Filiberto quiere saber la mínima longitud que el camino del ganador puede tener si a lo más unas de las carreteras esta bloqueada. Esto significa, ¿que entre todos los caminos de máxima longitud que los competidores pueden elegir y todas las posibles carreteras que pueden bloquearse, cual es la mínima longitud del camino del ganador que puede tener al finalizar la competencia si a lo más una de las carreteras es bloqueada?

Entrada

La primera línea de entrada contiene 2 enteros n y m ($2 \leq n \leq 10^5, 1 \leq m \leq \min(n(n-1)/2, 10^6)$) -- el número de estaciones y el número de carreteras en la montaña.

Las siguientes m líneas contienen 2 enteros cada una u y v ($1 \leq u, v \leq n, u \neq v$) representando una carretera que va de la estación u a la estación v , solo en esa dirección. Está garantizado que existirá a lo más una carretera entre cualquier par de estaciones.

Salida

Imprime una línea con un solo entero: la ruta de longitud mínima que los competidores pueden alcanzar si a lo más una de las carreteras está bloqueada.

Ejemplos

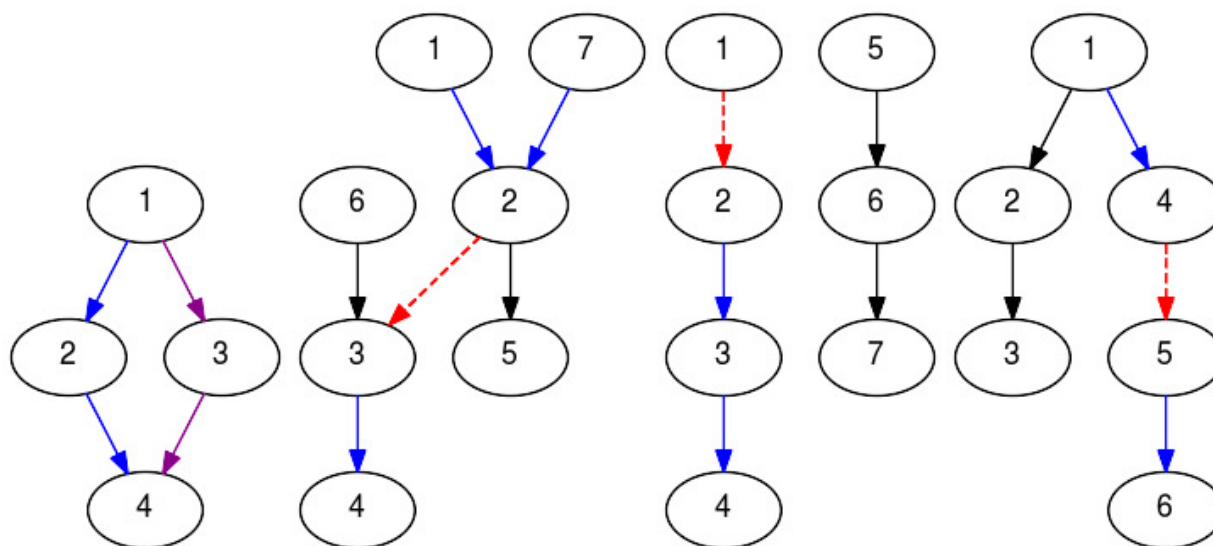
Entrada	Salida
4 4 1 2 1 3 3 4 2 4	2
7 6 1 2 2 3 2 5 6 3 7 2 3 4	2
7 5 1 2 2 3 3 4 5 6 6 7	0
6 5 1 2 1 4 2 3 4 5 5 6	1

Explicación de casos

En el segundo ejemplo, cada corredor comenzará desde las estaciones número 1 o 7. Si la carretera (2,3) está bloqueada, todos los corredores tendrán que cambiar de ruta en la estación 2 e ir hacia la estación 5.

En el tercer caso de ejemplo, la longitud mínima que se puede lograr ocurre bloqueando el primer camino del único camino de longitud máxima que los corredores pueden elegir. Bloquear cualquier otra carretera resulta en una ruta más larga.

En el último ejemplo solo hay un camino de longitud máxima, pero la mejor opción es bloquear el camino (4,5). Si bloqueamos el camino (1,4), los corredores pueden cambiar de ruta antes y alcanzar un camino de longitud 2.



Coding Cup TecNM — D. Red de mensajería TecNM

Points	100	Memory limit	32 MiB
Time limit (case)	1s	Time limit (total)	1m0s

En el Tecnológico Nacional de México los profesores interesados por optimizar recursos para hacer más y mejores concursos de programación han creado una red de mensajería interna en la cual intercambian material didáctico, libros y equipo tecnológico durante los viajes que hacen entre diferentes ciudades por conferencias y talleres.

En esta red cada profesor se registra listando las ciudades por las cuales viaja frecuentemente. Posteriormente cualquier profesor puede pedirle a miembros de la red que lleven un paquete en su siguiente viaje a otra ciudad en la que el paquete es entregado ya sea a su destinatario final o a otro profesor de la red.

Los paquetes que se envían en esta red no son urgentes. Lo único que importa es que lleguen a su destino independientemente del tiempo que tome.

Ejemplo, si en la red se hubieran registrado tres profesores que viajan entre las siguientes ciudades:

- Profesor 1: Orizaba, Pachuca
- Profesor 2: Pachuca, Celaya, Acapulco
- Profesor 3: Campeche, Chetumal

Sería posible mandar un paquete de Orizaba a Acapulco a través de este sistema:

- Remitente le da el paquete a Profesor 1 en Orizaba
- Profesor 1 lleva el paquete a Pachuca en su siguiente conferencia y se lo entrega a Profesor 2
- Profesor 2 lleva el paquete a Acapulco en su siguiente taller y se lo entrega al destinatario

Sin embargo no es posible usar la red para enviar un paquete de Orizaba a Chetumal. Lamentablemente se tiene que usar la mensajería normal y usar dinero del presupuesto.

Eres el encargado de escribir un programa que determine si es posible que un paquete llegue de un origen a un destino usando esta red.

Se te da una lista representando a los profesores registrados en la red con las ciudades que frecuentan.

Se te da una ciudad origen y una ciudad destino y en dado caso de ser posible usar la red para enviar el paquete imprime "posible", de otra forma escribe "imposible".

Input

- Un entero, N , indicando la cantidad de profesores registrados
- N renglones representando a los profesores registrados en la red. Cada renglón seguirá el formato M CIUDAD_1 CIUDAD_2 ... CIUDAD_M donde:
 - M es la cantidad de ciudades que el profesor frecuenta separadas por un espacio
 - CIUDAD_1, CIUDAD_2, ..., CIUDAD_M son los nombres de las M ciudades
- Una cadena *ORIGEN* que representa el nombre de la ciudad origen
- Una cadena *DESTINO* que representa el nombre de la ciudad destino

Notas

- $1 \leq N \leq 100$
- $1 \leq M \leq 100$
- Los nombres de las ciudades no tienen espacios (son una sola palabra) y usan sólo letras minúsculas y dígitos ([a-z0-9]).

Output

Imprime `posible` si se puede utilizar la red para enviar el paquete, de otra forma imprime `imposible`.

Ejemplos

Entrada	Salida	Descripción
3 2 orizaba pachuca 3 pachuca celaya acapulco 2 campeche chetumal orizaba acapulco	posible	Es posible enviar el paquete usando al primer y segundo profesor
3 2 orizaba pachuca 3 pachuca celaya acapulco 2 campeche chetumal orizaba chetumal	imposible	No hay forma en la que el paquete llegue de Orizaba a Chetumal usando la red de profesores

Coding Cup TecNM — E. Instancias de clases

Points	100	Memory limit	32 MiB
Time limit (case)	3s	Time limit (total)	1m0s

Descripción

ITSURITO está programado en java, él es muy cuidadoso al programar, es tan cuidadoso que siempre declara una sola variable de referencia por línea y en la misma línea siempre crea la instancia de la clase y la asigna a su variable. Por ejemplo, para declarar sus variables y crear las instancias de clase lo hace de la siguiente manera:

```
Integer miVariable = new Integer("5");
```

ITSURITO te ha pedido que dado una lista de variables de referencia declaradas e inicializadas y una lista de asignaciones entre las diferentes variables de referencia, le indiques cuántos objetos siguen siendo referenciados en el código.

NOTA: una variable de referencia almacena una dirección que apunta a una instancia que esta en memoria.

Entrada

Un número D, que indica el número de líneas que declaran variables de referencia y crean una instancia, seguido de un número A que indica el número de líneas que contienen asignaciones. En seguida D líneas donde cada D_i contiene una cadena que representa una línea de código que realiza una declaración e instancia de una clase, seguido de A líneas de código que realizan una asignación de variables.

La línea de código puede contener caracteres del alfabeto inglés (a-z y/o A-Z), dígitos del 0 al 9 y los siguientes signos: ";", ".", "=", y espacios en blanco.

Salida

Un número N, que indica cuantas instancias están referenciadas al finalizar.

Ejemplo

Al final de la siguiente lista solo quedan 2 instancias referenciadas.

```
Integer valor1 = new Integer("8"); <--Aquí hay una variable de referencia que apunta a una instancia
```

```
Integer valor2 = new Integer("9"); <-- Aquí se agrega una instancia referenciada por su variable, para dar un total de 2 instancias referenciadas
```

```
String valor3 = new String("uno más uno"); <-- Aquí ya hay 3 instancias referenciadas cada uno por su variable de referencia.
```

```
valor1 = valor2; <-- Aquí al finalizar se pierde la referencia a la instancia que es apuntada por el valor2, por lo tanto, solo hay dos instancias que siguen siendo referenciadas.
```

Entrada	Salida	Descripción
3 2 Integer estatura = new Integer("5"); Integer edad = new Integer("55"); Integer codigo = new Integer("66"); estatura = edad; codigo = estatura;	1	<p>En la línea 3,4 y 5 se crean 3 instancias y 3 variables de referencias apuntando cada una a una instancia.</p> <p>En la línea 6 la variable de referencia estatura ahora va a apuntar a la instancia referenciada por la variable edad, por lo que la instancia que estaba siendo apuntada por estatura se pierde ya que no queda ninguna variable apuntando a ella, quedando dos variables de referencia apuntando a la misma instancia es decir estatura y edad apuntan a la misma instancia y codigo a otra instancias diferente.</p> <p>En la línea 7 la variable de referencia codigo apuntara a la instancia referenciada por estatura, que es la misma instancia que esta referenciada por edad por lo que al final quedarán 3 variables de referencia apuntando a la misma instancia.</p>
6 0 String primer = new String("Solo soy quien te busca"); String segunda = new String("que a veces piensa"); String tercera = new String("que te encuentre"); String cuarta = new String("y otras tantas pienso"); String quinta = new String("que ya te perdi"); String sexta = new String("DMO");	6	
3 2 Double numero1 = new Double("5.5"); Double numero2 = new Double("7.5"); Double numero3 = new Double("8.5"); numero1 = numero2; numero2 = numero3;	2	<p>En la línea 3,4 y 5 se crean 3 instancias y 3 variables de referencias apuntando cada una a una instancia.</p> <p>En la línea 6, la variable de referencia numero1, es asignada a apuntar a la instancia que es referenciada por la variable numero2, por lo que la instancia que estaba apuntada por el numero1 se pierde y ya hay dos variables de referencia apuntando a la misma instancia estas variables son numero1 y numero2.</p> <p>En la línea 7, la variable de referencia numero2 ahora va a apuntar a la instancia a la que apunta la variable de referencia numero3, por lo que al final hay 2 objetos que son referenciados, ya que la instancia creado en la línea 3 ya no es referenciada.</p>

Límites

- D es mayor que 0 y menor que 100
- A es mayor o igual que 0 y menor o igual que 25

Coding Cup TecNM — F. Contando Estudiantes

Points	100	Memory limit	64 MiB
Time limit (case)	2s	Time limit (total)	20s

Descripción

Para el concurso "Coding Cup TecNM" se han inscrito una cantidad impresionante de estudiantes. Han llegado muchas peticiones de personas que quieren participar en este gran concurso y es muy difícil llevar estadísticas del evento.

Es por ello que el profesor German Gutiérrez necesita tu ayuda para resolver este problema. Faltan m minutos para que comience el concurso. Por cada minuto que pasa pueden pasar dos acciones: Un estudiante puede registrarse para participar en el concurso o el profesor German querrá saber cuántas personas registradas en el evento tienen una edad menor o igual a un número dado x .

Es por esto que se te ha dado esta tarea para poder ayudar al profesor German y dejarlo libre para que pueda ocupar ese tiempo en otras actividades y así poder ofrecerte un evento de mejor calidad.

Entrada

La primera línea de entrada contiene un número entero m ($2 \leq m \leq 10^5$) - el número de minutos que faltan para que el concurso comience.

Las siguientes m líneas contienen 2 enteros cada una p y x ($1 \leq p \leq 2, 1 \leq x \leq 10^6$).

Si $p = 1$, es una pregunta de tipo 1 y significa que un nuevo estudiante con edad x se ha registrado para participar en el concurso.

Si $p = 2$, es una pregunta de tipo 2 y significa que el profesor German quiere saber cuántos estudiantes registrados tienen edad menor o igual que x .

Nota que puede haber varios estudiantes con la misma edad en el concurso. Y si, los estudiantes para este concurso pueden vivir muchísimos años.

Salida

Para cada pregunta del tipo 2 tienes que imprimir en una sola línea cuantas personas registradas para el concurso tienen edad menor o igual que x .

Ejemplos

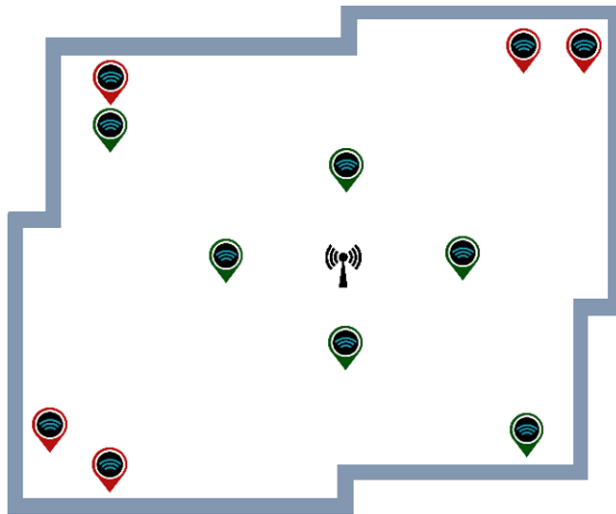
Entrada	Salida
10	1
1 1	2
2 5	3
1 2	4
2 5	5
1 3	
2 5	
1 4	
2 5	
1 5	
2 5	
5	0
1 100	2
1 200	
1 300	
2 20	
2 250	

Coding Cup TecNM — G. Verificador de alcance

Points	100	Memory limit	32 MiB
Time limit (case)	1s	Time limit (total)	1m0s

Descripción

José Rómulo Sosa Ortiz es un reconocido empresario dedicado a la implementación de soluciones en domótica, es decir, se encarga de implementar sistemas capaces de automatizar una vivienda o edificación de cualquier tipo, aportando servicios de gestión energética, seguridad, bienestar y comunicación. Todos los dispositivos que se usan para generar esta automatización requieren estar interconectados para que el sistema de domótica funcione de manera óptima, por ello, José dedica mucho tiempo a realizar la instalación del sistema de comunicación, puesto que antes de iniciar debe realizar muchas pruebas con diversas propuestas de distribución de los dispositivos receptores, ya que tiene que asegurarse de que todos los receptores estén dentro del alcance de la antena emisora. En cada una de sus propuestas José genera una gráfica como la que sigue:



Debido a que esto le lleva mucho tiempo, José requiere que le ayudes a probar cada una de sus propuestas de distribución indicándole qué receptores quedan dentro o fuera del alcance. José te brindará el alcance de la antena emisora, cuantos dispositivos receptores requiere colocar y su ubicación respecto a la antena emisora, y a partir de estos datos le presentará el análisis de dicha distribución. Debes considerar que José te proporcionará coordenadas de los dispositivos receptores asumiendo que el dispositivo emisor se coloca en el origen. José quiere que en la representación gráfica marques con una "x" aquellos receptores que queden fuera del alcance de la antena emisora, con un "+" si el receptor se encuentra dentro del rango de difusión y la antena emisora con un "#", por otro lado, también desea que le muestres con una "o" aquellos espacios que no cuentan con un dispositivo (emisor o receptor). Un dispositivo receptor está dentro del rango de la antena emisora, si la distancia Euclidiana entre ambos es menor o igual al alcance de la antena emisora. Es decir, la distancia directa entre ambos dispositivos:



Debe considerarse que cada cuadrante debe mostrarse de forma rectangular, considerando como tope el punto más alejado en X y Y de cada cuadrante, de manera que no se requiere rellenar todo el cuadrante con "o" solamente las requeridas. Se asegura que en el origen nunca se colocará un dispositivo receptor. Nota: solo son permitidos los espacios que sean estrictamente requeridos para el acomodo de la salida.

Entrada

La primera línea contendrá un entero **A** que representa el alcance de la antena emisora, la segunda línea contendrá un entero **N** el cual permitirá saber cuántos dispositivos receptores se requieren colocar, seguido de **N** pares de enteros que representan las coordenadas **X** y **Y** donde José plantea ubicar cada dispositivo receptor.

Salida

La salida deberá presentarse con la siguiente notación: "o" para las ubicaciones sin receptor, "x" para los receptores fuera de alcance, "+" para los receptores dentro del alcance y "#" para indicar la ubicación del dispositivo emisor. Cada cuadrante toma el eje de la derecha para representar a los dispositivos que quedan en las posiciones $X = 0$ ó $Y = 0$, quedando como: Cuadrante (+,+) toma dentro de su representación las ubicaciones (+,0). Cuadrante (+,-) toma dentro de su representación las ubicaciones (0,-). Cuadrante (-,-) toma dentro de su representación las ubicaciones (-,0). Cuadrante (-,+) toma dentro de su representación las ubicaciones (0,+).

Ejemplo

Entrada	Salida	Descripción
2 6 5 6 -1 0 3 -4 -6 -1 -7 2 -6 4	oooox ooooo oxoooooooooooo oooooooooooooooo xoooooooooooooooo oooooooooooooooo ooooo+#ooooo xoooooooooooo oooo oooo oooox	Con un alcance de 2, solamente un dispositivo se encuentra dentro del alcance, además como puedes observar la salida solo imprime hasta donde se encuentra el dispositivo más alejado del origen, generando impresiones rectangulares y aplicando de forma distinta para cada cuadrante.
3 6 1 3 -5 0 3 0 -1 -5 -5 -5 3 3	xox ooo ooo xoooo#oo+ ooooo ooooo ooooo ooooo ooooo xoooox	Únicamente un dispositivo dentro del alcance, además de que solo hay dispositivos en el cuadrante (+,+) y (-,-).
6 3 -3 -3 -1 -5 -5 -5	ooooo# ooooo ooooo oo+oo ooooo xooo+	2 dispositivos dentro del alcance y la distribución se encuentra solo 1 cuadrante.

Límites

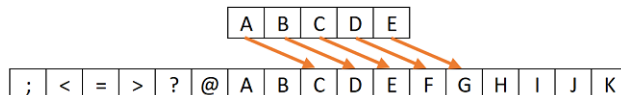
- $1 \leq N \leq 100$
- $1 \leq A \leq 90$
- $-60 \leq X, Y \leq 60$

Coding Cup TecNM — H. Mensajes secretos

Points	100	Memory limit	32 MiB
Time limit (case)	1s	Time limit (total)	1m0s

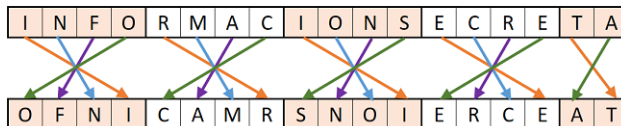
Descripción

Juan, María y Carlos son un grupo de amigos que participarán en el CodingCup TecNM y como un ejercicio de entrenamiento se han puesto el reto de generar un algoritmo sencillo de encriptación que les permita mandarse mensajes públicamente, pero que no sea tan fácil que sus compañeros descifren su contenido. A Juan se le ha ocurrido que pueden hacer un corrimiento de caracteres de forma que al colocar una frase el texto no tenga un sentido directo, pues se reemplazarán los caracteres por el carácter equivalente haciendo un corrimiento positivo o negativo en la tabla ASCII, por ejemplo: si el algoritmo genera una nueva versión de la cadena con un corrimiento de 2, la A en su versión encriptada sería una C, la B una D, la C una E y así sucesivamente:



Juan espera que de acuerdo a los corrimientos para encriptar y descifrar, se utilicen los valores ASCII del 33 al 122, puesto que se considera como base el rango de las letras mayúsculas (65 a 97), cuyo corrimiento puede ser positivo ($97+32=122$) o negativo ($65-32=33$).

María cree que la propuesta generada por Juan es demasiado simple y se le ha ocurrido trabajar con cadenas invertidas, podría solamente invertirse la cadena, pero María cree que eso también sería muy sencillo de percibir, por lo que se le ha ocurrido tomar fragmentos de la cadena, de forma que si se invierta pero en grupos de N caracteres hasta invertir toda la cadena y así lograr que la cadena resultante quede un poco más complicada de descifrar. María propone que si la cadena fuera "INFORMACIONSECRETA", y si se toman grupos de 4, la operación de reversión quede como se muestra a continuación:



Finalmente, Carlos propone que se fusionen ambas propuestas y que adicionalmente al corrimiento se maneje la cadena en grupos invertidos, por otro lado, Carlos propone que la cadena a encriptar siempre se interprete en mayúscula y que ahora para hacer más fácil el proceso de encriptar a pesar de que el usuario la indique en minúsculas. Y al descifrar se tomará la cadena tal como la indique el usuario.

Entrada

La entrada estará conformada por 2 líneas:

- La primera contendrá la cadena *F* a encriptar o descifrar que contiene solo letras (a-z y A-Z), de la cual el primer carácter indicará con una "e" o "E" si se encriptará la cadena, o con una "d" o "D" si la cadena se descifrá.
- La segunda contendrá primero la letra "g" seguida de un número entero *G* que indicará el tamaño de grupo para la reversión de la cadena (este nunca será más grande que el tamaño de la cadena a encriptar o descifrar) separado por un espacio de la letra "c" seguida por un número entero *C* que indicará el corrimiento positivo o negativo mediante el cual se obtendrá el carácter encriptado o descifrado. Debes considerar que tanto para encriptar como para descifrar debes indicar el mismo tamaño de grupo como el mismo corrimiento.

Salida

La cadena encriptada o descifrada de acuerdo al tamaño de grupo de reversión y al corrimiento.

Ejemplo

Entrada	Salida	Descripción
eEsteeselpimerconcurso de programacion del TecNM g6 c8	[MM\[MUQZXTMKVWKZMMLW[Z]IZOWZXVWQKIUKM\TMLUV	Encriptación de la cadena en grupos de reversión de 6 caracteres y un corrimiento en la tabla ASCII de 8 posiciones.
eInformacion secreta g4 c2	QHPKECOTUPQKGTEGCV	Encriptación de la cadena en grupos de reversión de 4 caracteres y un corrimiento en la tabla ASCII de 2 posiciones.
ea g1 c10	K	
DOMRVCRJCLCCSLGBLMACRNSAEQCBKLAP?PDGAL? g6 c-2	TEXTOQUEENELCONDINGCUPTECNMDESCIFRARAN	

Límites

- $1 \leq G \leq$ Tamaño de la cadena $F-1$
- $-32 \leq c \leq 32$
- La longitud de la cadena F puede ser de hasta 401 caracteres

Coding Cup TecNM — I. Aprendiendo arreglos

Points	100	Memory limit	32 MiB
Time limit (case)	1s	Time limit (total)	1m0s

Descripción

En tus primeros días en el TecNM te han enseñado a programar lo básico del tema de arreglos. Tu primer tarea consiste en encontrar la cantidad de números que hay en un rango. Como entrada se tendrán N números y un rango. Lo único que debe hacer el programa es calcular la cantidad de números que hay dicho rango.

Entrada

Leer primero N, donde $1 \leq N \leq 100$. El dato N indica la cantidad de números que se leerán desde el teclado. En las siguientes N líneas se encuentran los datos de entrada. Al finalizar se tienen dos números enteros A y B que indican el rango.

$0 \leq A \leq B \leq 1000$

Salida

Imprima la cantidad de números que existen en el rango desde A hasta B.

Ejemplos

Entrada	Salida
10 1 2 3 4 5 6 7 8 9 10 0 2	2
5 3 25 100 23 28 20 30	3