



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema A - Asombroso League of Legends

Límite de tiempo: 3 segundos

Problema

League of Legends es un juego clasificación MOBA (Multiplayer Online Battle Arena), en el que combaten varios campeones en enfrentamientos épicos, en los que el equipo que juegue mejor en equipo gana.

En éste problema nos centraremos en 2 campeones, Lux y Volibear; Volibear está persiguiendo a Lux, la cuál tiene poca vida, tal que si Volibear alcanza a Lux, ésta morirá sin poder hacer nada al respecto.

Afortunadamente, Lux tiene un hechizo que le permite hacer que Volibear deje de moverse durante cierto tiempo(T); el hechizo lo puede usar cada C segundos (a C también se le llama “enfriamiento” de hechizo).

Tu tarea consiste en, sabiendo las velocidades de los campeones (Lux y Volibear), el tiempo que el hechizo de Lux deja atrapado a Volibear, y el enfriamiento del hechizo de Lux, determinar si Lux será alcanzada por Volibear o no.

Entrada

La primera línea contendrá un número N ($0 < N \leq 100$), siendo N el número de casos de prueba. Cada una de las siguientes N líneas tendrá 4 enteros, $V1$, $V2$, T y C , ($0 < V1, V2 \leq 500$) ($0 < T, C \leq 10$) las velocidades de Volibear, Lux, el tiempo que el hechizo de Lux atrapa a Volibear, y el enfriamiento de su hechizo respectivamente.

(Las velocidades $V1$ y $V2$ están dadas en unidades/segundo, y T, C están dados en segundos. También supondremos que Volibear y Lux empiezan en puntos diferentes, Lux está adelante de Volibear, y ambos corren en línea recta)

Salida

Para cada caso de prueba, se tendrá que imprimir una línea, imprimiendo “Se muere” si Volibear alcanza a Lux, y “Se salva” en caso contrario.

Entrada Ejemplo

```
2
10 300 1 2
300 100 1 10
```

Salida Ejemplo

```
Se salva
Se muere
```



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema B - Bobby el Minotauro

Límite de tiempo: 3 segundos

Problema

Desafortunadamente para el “flojo Mau”, un día se topó con el malvado minotauro devorador de hombres conocido como ... Bobby, y cayó directo en su trampa. Bobby no es cualquier minotauro, es un minotauro que habla, pero el sólo entiende frases que son palíndromos. Mau se dió cuenta que si le hablaba con palíndromos a Bobby, él lo dejaría ir. Afortunadamente Mau tiene consigo un teléfono inteligente con un programa que identifica frases palindrómicas, el cual tú codificaste para él ... ¿o no?.

Entrada

Te serán dadas muchas frases. Cada frase sólo contendrá letras mayúsculas de la ‘A’ hasta la ‘Z’ y los siguientes caracteres: ‘.’, ‘,’, ‘!’, ‘?’. El final de la entrada será una línea conteniendo la palabra “HECHO”, que no deberá ser procesada. Cada frase tendrá a lo más 200 caracteres.

Salida

Para cada frase imprime una línea con la palabra “NO SERAS COMIDO” si la frase es un palíndromo, o “OH NO!” si no es un palíndromo.

Entrada Ejemplo

```
ROMA TIBI SUBITO MOTIBUS IBIT AMOR.  
ME DEJARIAS IR?  
ARRIBA LA BIRRA  
TRAIGAN AL MINOTAURO!  
HECHO
```

Salida Ejemplo

```
NO SERAS COMIDO  
OH NO!  
NO SERAS COMIDO  
OH NO!
```

Anotaciones

Un palíndromo es una frase que se lee igual de atrás hacia adelante y de adelante hacia atrás. Tienes que determinar si son palíndromos o no, ignorando signos de puntuación.

Modificado de un problema original de UVA Online Judge



Problema C - Coleccionista

Límite de tiempo: 3 segundos

Problema

Hernán es un joven apasionado por los videojuegos, le gustan de todo género, desde shooters, plataformas, carreras y estrategia hasta llegar a los de aventura. Uno de sus juegos favoritos es Pokémon. Pokémon es un juego en el que tu objetivo es conseguir todos los monstruos disponibles (llamados pokémon) y las medallas de todos los gimnasios para así demostrar que has vencido a los mejores.

Aunque a Hernán le encantan todas las versiones de este juego, desde Rojo y Azul hasta X y Y, él no siempre quiere capturar a todos los pokémon disponibles en algunas versiones, ya sea porque considera que algunos son débiles o pequeños, e incluso feos.

Debido a esto (y a que no le gusta hacer algo tan fácil), te ha pedido que le ayudes con una sencilla tarea.

Dados el número total de pokémon disponibles en cada versión y dos listas de pokémon, la primera será la lista de los que ya posee y la segunda la lista de los que no le interesan, determina las siguientes dos cosas:

- Cuántos y cuáles pokémon le faltan por capturar (solo de los que le interesan).
- Cuantos y cuales pokémon que posee puede cambiar, ésto es, los pokémon que ya tiene y no le interesan.

Los pokémon se identificarán por enteros para facilitar su procesamiento.

Es sabido que Hernán no posee mas de una especie del mismo pokémon por versión.

Entrada

La entrada consta de varios casos de prueba.

Cada caso consta de 3 líneas, la primera línea será un entero N , ($10 \leq N \leq 1000$) que indica el total de pokémon en la versión del juego.

La segunda línea contiene un entero A , ($0 \leq A \leq N$) el número de pokémon que Hernán ha capturado en esa versión, seguido por una lista de A enteros que representan a estos pokémon. Todos estos números están separados por un espacio.

La tercera línea contiene un entero X , ($0 \leq X \leq N$) el total de pokémon que no le interesan a Hernán, seguido por X enteros que representan a los mismos. Todos estos números están separados por un espacio.

Para facilitar su identificación, los pokémon estan representados por números desde 1 hasta N .

Salida

Para cada caso imprime dos líneas:

- La primera línea será un entero F que representará el número de pokémon que le interesa capturar a Hernán y aún no tiene, seguido por F enteros que representan estos pokémon impresos en orden ascendente.
- La segunda constará de un entero C que representará el número de pokémon que Hernán puede cambiar, seguido por C enteros que serán los pokémon cambiables impresos en orden ascendente.

Entrada ejemplo

```
11
5 3 5 9 10 11
6 2 3 4 5 9 11
```

Salida ejemplo

```
4 1 6 7 8
4 3 5 9 11
```

Maximiliano Vera Luna - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema D - Dominó

Límite de tiempo: 1 segundo

Problema

Mauricio tiene N piezas de domino en fila. Cada pieza se divide en dos partes iguales en tamaño, - la superior y la inferior -. Cada una de las partes contiene un número del 1 al 6.

A Mauricio le encantan los números pares, así que quiere que la suma de los números de las mitades superiores y la suma de los números de las mitades inferiores sea par.

Para lograr eso, Mauricio puede rotar las piezas de dominó 180 grados. Después de una rotación las mitades cambian de lugares. Esta acción toma 1 segundo. Ayuda a Mauricio a averiguar el tiempo mínimo que debe gastar girando piezas de dominó para que se cumpla lo que él quiere.

Entrada

La primera línea contiene un entero T ($1 < T < 50$), el número de casos de prueba. Para cada uno de los siguientes T casos, hay una línea que contiene un entero N ($1 \leq N \leq 100$), el número de piezas de dominó que tiene Mau. Las siguientes N líneas contienen dos enteros X_i, Y_i , ($1 \leq X_i, Y_i \leq 6$) cada una, separados por un espacio. El número X_i es el que está escrito inicialmente en la mitad superior del i -ésimo dominó, mientras que Y_i es el que está escrito en la mitad inferior.

Salida

Para cada caso imprime sólo una línea, esta deberá decir "Caso T: " seguido por el mínimo de segundos requeridos para realizar la tarea. T es el caso actual empezando a numerarlos desde 1. Si resulta imposible lograr tal tarea, imprime un '-1' (sin comillas).

Entrada Ejemplo

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

Salida Ejemplo

```
Caso 1: 0
Caso 2: -1
```



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema E - Esto es fácil

Límite de tiempo: 5 segundos

Problema

Este problema es fácil, tienes que buscar 2 enteros en una lista, tal que sumados den otro número.

Entrada

La primera línea tendrá un número T que representa el número de casos de prueba.

Las siguientes $3T$ líneas contendrán los casos de prueba, cada caso de prueba tendrá 3 líneas, la primera línea tendrá un número N ($2 \leq N \leq 10000$) que representa la cantidad de números que tendrá el arreglo, la segunda línea tendrá N números enteros K_i ($1 \leq i \leq N$) ($0 \leq |K_i| \leq 10000$) los números sobre los cuáles tienes que buscar la pareja de números, y la tercera línea tendrá un número entero M ($0 \leq |M| \leq 20000$).

Salida

Se imprimirán T líneas, una por cada caso de prueba, con la palabra "SI" si es que existen 2 números K_i y K_j tales que $K_i + K_j = M$, y "NO" en caso contrario.

Entrada Ejemplo

```
2
3
1 2 3
4
5
6 7 8 9 10
11
```

Salida Ejemplo

```
SI
NO
```

Sergio Adrián Lagunas Pinacho - Grupo de Algoritmia Avanzada y Programación Competitiva



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema F - Falsa Simulación

Límite de tiempo: 1 segundo

Problema

El siguiente problema consta de una simulación muy concreta... ¿o no?. Tendrán una matriz M de tamaño $1234(\text{filas}) \times 5678(\text{columnas})$. Está inicialmente llena con enteros desde el $1, 2, \dots, 1234 \times 5678$ en orden ascendente de izquierda a derecha y de arriba a abajo. (La primera fila está llena con los enteros $1, 2, \dots, 5678$; la segunda con los enteros $5679, 5680, \dots, 11376$, etc) Existen 4 tipos de comandos:

- “R x y” intercambia la x -ésima y y -ésima fila de M .
- “C x y” intercambia la x -ésima y y -ésima columna de M .
- “Q x y” imprime $M(x, y)$ (El entero que está en la posición (x, y) en la matriz M).
- “W z” imprime x e y donde $z = M(x, y)$.

Entrada

Te será dada una lista de comandos válidos, uno por línea. La entrada termina con el fin de archivo.

Salida

Cada que recibas el comando “Q x y” imprime una línea con el valor $M(x, y)$, y cada que recibas el comando “W z” imprime una línea con los valores x e y separados por un espacio, la posición en la que se encuentra el número z .

Entrada Ejemplo

```
R 1 2
Q 1 1
Q 2 1
W 1
W 5679
C 1 2
Q 1 1
Q 2 1
W 1
W 5679
```

Salida Ejemplo

5679

1

2 1

1 1

5680

2

2 2

1 2

Sphere Online Judge



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema G - Geómetra Hermann

Límite de tiempo: 3 segundos

Problema

Durante el siglo XIX el matemático Hermann Minkowski investigó acerca de un tipo de geometría no euclidiana, llamada geometría de taxi. En la geometría de taxi la distancia entre dos puntos $T_1(x_1, y_1)$ y $T_2(x_2, y_2)$ es definida como $D(T_1, T_2) = |x_1 - x_2| + |y_1 - y_2|$. Cualquier otra definición se define igual que en la geometría euclidiana, incluyendo la definición del círculo: Un círculo es el conjunto de todos los puntos en un plano a una distancia dada(radio) de un punto dado(centro del círculo). Nosotros estamos interesados en la diferencia de las áreas de dos círculos con radio R , uno de los cuáles está dado en un espacio normal(euclidean) y el otro en una geometría de taxi. Dicha tarea ha sido encomendada a tí.

Entrada

Cada línea tendrá un solo número entero $1 \leq R \leq 10000$. Deberás leer hasta el final del archivo.

Salida

Para cada caso deberás imprimir dos líneas, la primera contendrá el area del círculo con radio R en una geometría euclidiana y la segunda línea será el area del círculo con radio R en una geometría de taxi. Tu salida deberá ser redondeada a 4 lugares decimales. Además puedes asumir de manera segura de usando valores flotantes de precisión doble y π igual 3.141592653589793 será suficiente.

Entrada Ejemplo

21
9384
887

Salida Ejemplo

1385.4424
882.0000
276646940.0487
176118912.0000
2471707.7105
1573538.0000



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema I - Investigando laberintos

Límite de tiempo: 3 segundos

Problema

Un laberinto es representado en una cuadrícula de dos dimensiones como está ilustrado en la figura siguiente.

```
XXXXXXXXXXXXXXXXXXXXX
X  X  X  X  X  X
X      X  X  X
X  X  X  X  X  X
XXXXXX XXX XXXXXXXXXX
X  X  X  X  X  X
X  X      *      X
X  X  X  X  X  X
XXXXXXXXXXXXXXXXXXXXX
```

Cada punto de la cuadrícula es representado por un carácter. Un carácter espacio () representa los lugares en donde puedes caminar.

Las paredes del laberinto son representadas por letras mayúsculas (de A a Z), esto es, lugares donde no puedes caminar.

Tu tarea es, dado un punto de inicio dentro del laberinto, representado por un asterisco (*), debes de marcar todos los posibles lugares a donde puedes llegar caminando con el carácter gato (#). El asterisco debe ser reemplazado por el carácter gato.

La forma en que puedes caminar es 4 conexidad, esto significa que solo puedes moverte hacia arriba, abajo, izquierda o derecha.

La cuadrícula presentada anteriormente quedaría:

```
XXXXXXXXXXXXXXXXXXXXX
X###X###X###X  X  X
X#####X  X  X
X###X###X###X  X  X
XXXXXX#XXX#XXXXXXX
X  X###X###X###X
X  X#####X
X  X###X###X###X
XXXXXXXXXXXXXXXXXXXXX
```

Entrada

La primera línea de entrada es T , el número de casos. La primera línea de cada caso son dos enteros, N y M , el número de renglones y el número de columnas de la cuadrícula respectivamente.

Las siguientes N líneas contienen M caracteres, las cuales representan al laberinto.

Salida

N líneas de M caracteres cada una, las cuales representa a la cuadrícula marcada en la forma en que se indicó. Imprime una línea en blanco al final de cada caso.

Entrada Ejemplo

```
2
5 5
XXXXX
X*  X
X   X
XX  XX
XXXXX
5 9
AAAAAAAAA
C*  D   B
C   D   B
C   D   B
EEEEEEEE
```

Salida Ejemplo

```
XXXXX
X###X
X###X
XX#XX
XXXXX

AAAAAAAAA
C###D   B
C###D   B
C###D   B
EEEEEEEE
```



Facultad de Estudios Superiores

Acatlán

Centro de Desarrollo Tecnológico
Departamento de Servicios de Cómputo

Problema H - Historia de los relojes

Límite de tiempo: 3 segundos

Problema

El interés medieval en contribuciones mecánicas es bien ilustrado por la invención del reloj mecánico, el más viejo el cual es controlado por contrapesos, y un brazo oscilante con un engranaje. Ésto fué en 1386.

Los relojes controlador por resortes aparecieron por la mitad del siglo 15, haciendo posible construir mecanismos más compactos, y preparando el camino para el reloj portátil.

Los primeros relojes de péndulo controlador por resortes fueron comúnmente puestos en pequeñas repisas de pared, y después sobre repisas. Muchos relojes de soporte tenían un cajoncito para guardar la llave del vidrio. Los relojes de soporte más recientes, hechos después de 1660, tenían diseño arquitectónico, con pilares a los lados.

El los siglos 17 - 18 en francia, los relojes de mesa se convirtieron en un objeto de diseño monumental.

Uno de los primeros relojes atómicos fué un reloj controlado por amoniaco . Éste fué hecho en 1949 en el "National Bureau of Standards", Washington, D.C.

La historia de los relojes es fascinante, pero no tiene que ver con éste problema. En éste problema te darán el ángulo entre las manecillas de un reloj en forma de círculo, como el que se muestra en la figura, y tienes que indicar si existe una hora del día tal que el ángulo entre ambas manecillas sea el ingresado por el usuario.

Para éste problema, supondremos que el reloj tiene 60 marcas, una para cada minuto, y que tanto la manecilla de los minutos, tanto como la de las horas sólo pueden estar apuntando a alguna de las marcas.



Entrada

Está dada por varias líneas, cada una describiendo un caso de entrada. Cada línea contiene un entero A que representa el ángulo entre las manecillas ($0 \leq A \leq 180$).

Salida

Para cada caso de entrada imprime una línea con la palabra “Caso i: ” (i es el número de caso), seguido por un espacio, seguido por un caracter. Si existe al menos una hora del día tal que el ángulo mínimo entre las manecillas del reloj sea exactamente A grados, imprime el caracter será ”Y”. En caso contrario será ”N”.

Entrada Ejemplo

```
90
65
66
67
128
0
180
```

Salida Ejemplo

```
Caso 1: Y
Caso 2: N
Caso 3: Y
Caso 4: N
Caso 5: N
Caso 6: Y
Caso 7: Y
```

Sergio Adrián Lagunas Pinacho - UNAM FESA