
Competencia de Programación

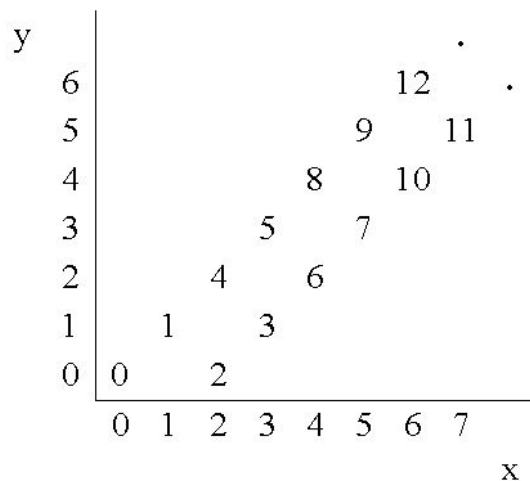
TecnoMate

Nivel “AEDD”

31 de Octubre de 2014

Problema 1: Number Steps (1112. NSTEPS)

Comenzando desde el punto (0,0) en el plano, hemos escrito todos los enteros no negativos 0,1, 2,... como se muestra en la figura. Por ejemplo, 1, 2, y 3, han sido escritos en los puntos (1,1), (2,0), y (3, 1) respectivamente, y de la misma forma continúa este patrón.



Ud. tiene que escribir un programa que lea las coordenadas (x, y) de un punto, y escriba el número (si existe) que ha sido escrito en ese punto. Las coordenadas (x, y) en la entrada están en el rango 0...10000.

Entrada

La primer línea de la entrada es N, el número de casos de prueba para este problema. En cada una de las N líneas siguientes, están x e y, representando las coordenadas (x, y) de un punto.

Salida

Para cada punto en la entrada, escribir el número escrito en ese punto, ó "No Number" si no corresponde.

Ejemplo

Input:

3
4 2
6 6
3 4

Output:

6
12
No Number

Problema 2: Desde Hasta (4762. DESDEHAS)

Mou y Larry desarrollaron un método para encriptar mensajes. Primero, deciden secretamente el número de columnas, y escriben el mensaje (sólo letras) hacia abajo por columnas, rellenando con letras al azar para completar un arreglo rectangular de letras. Por ejemplo, si el mensaje es "There's no place like home on a snowy night" y hay 5 columnas, Mo escribe:

```
t o i o y  
h p k n n  
e l e a i  
r a h s g  
e c o n h  
s e m o t  
n l e w x
```

Observar que Mo incluye sólo letras y las escribe a todas en minúsculas. En este ejemplo, Mou usó la letra 'x' para rellenar el mensaje hasta completar un rectángulo, aunque el podría haber usado cualquier letra. Mo luego envía el mensaje a Larry escribiendo las letras en cada fila, alternativamente de izquierda a derecha y de derecha a izquierda. Así, el rectángulo anterior se encripta como:

```
toiynnkpheleaigshareconhtomesnlewx
```

Tu trabajo es recuperar para Larry el mensaje original (junto con todas las letras de relleno) a partir del mensaje encriptado.

Entrada

Hay varios casos de entrada. Cada caso consiste de dos líneas. La primer línea contiene un entero en el rango 2..20 que indica el número de columnas usadas. La línea siguiente es un string de hasta 200 letras minúsculas. El último caso de entrada viene seguido por una línea que contiene únicamente un 0, indicando fin de la entrada.

Salida

Cada caso de entrada tiene que generar una línea de salida, con el texto original del mensaje, sin espacios.

Ejemplo

Input:

```
5  
toiynnkpheleaigshareconhtomesnlewx  
3  
ttyohhieneesiaabss  
0
```

Output:

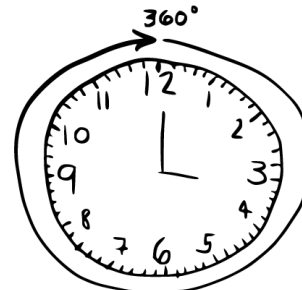
```
theresnoplacelikehomeonasnowynightx
```

thisistheeasyoneab

Problema 3: Reloj (15989. AI_REL)

Los relojes mecánicos utilizan manecillas para mostrar la hora los minutos y los segundos.

El reloj es de forma circular y todo círculo tiene 360 grados sexagesimales. Se desea conocer en qué posición en grados sexagesimales se encuentran las manecillas del reloj en un determinado tiempo.



Entrada

La entrada consiste en dos números enteros h y m ($0 \leq h < 24$) ($0 \leq m < 60$) donde h es la hora, m los minutos de un determinado tiempo.

Salida

Imprimir en qué ángulo se encuentran las manecillas del reloj en el tiempo ingresado, especificando el ángulo del horero y el minuterero en ese orden y separados por un espacio.

Ejemplos

Input:

23 59

Output:

330 354

Input:

12 00

Output:

0 0

Input:

16 47

Output:

120 282

Problema 4: Zig Zag (15990. AI_ZZ)

Hay un rumor de que en tu ciudad existe un tesoro enterrado desde hace siglos. Muchas personas venden mapas diciendo que te llevará directo al tesoro, lamentablemente no puedes saber si un mapa es verdadero o falso pero existe otra curiosidad más, se dice que el camino al tesoro tiene forma de zig zag.

Como deseas encontrar el tesoro antes que otros has comprado todos los distintos mapas que están a la venta, pero son realmente muchos, tu tarea es identificar que mapas contienen un camino con forma de zig zag.

Entrada

Un mapa se ha dividido en pequeños sectores cuadrados, por lo tanto tienes como entrada dos números enteros n y m ($3 \leq n, m \leq 100$) y ($n \leq m$), donde n es el alto total del mapa y m el ancho total del mapa. Luego n filas siguen conteniendo m números donde 0 representa la ausencia de un camino y 1 la presencia de un camino en el mapa, un camino puede estar unido de forma vertical, horizontal y diagonal.

Salida

Imprimir 'YES' si el camino tiene forma de zig zag y recorre todo el mapa empezando siempre desde la esquina superior izquierda, caso contrario imprimir 'NO'.

Ejemplo

Input:	Output:
3 10 1 0 0 0 1 0 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0 0 1 0 0 0 1 0 0 0	YES
3 5 0 1 0 0 0 0 0 1 0 1 0 0 0 1 0	NO
3 5 1 1 1 0 1 0 1 0 1 0 1 0 1 0 0	YES
3 3 0 0 0 1 1 1 0 0 0	NO
3 9 1 0 0 0 0 0 0 0 1	NO

0 1 0 1 0 1 0 1 0 0 0 1 0 1 0 1 0 0	
--	--

Problema 5: El juego de las fracciones (16416. AI_JUE)

Alice y Bob se divierten con el siguiente juego, cada uno escribe en un papel una acordada cantidad de fracciones en tan solo 5 segundos, El ganador es la persona que escribió mas fracciones irreducibles. Por ejemplo si la cantidad acordada de fracciones es 3 y Alice escribió: $1/12$, $7/12$ y $3/12$, Bob escribió: $3/7$, $2/8$ y $10/5$. Gana Alice porque tiene dos fracciones irreducibles $1/12$ y $7/12$ y Bob solo tiene una $3/7$.

Ayuda a Alice y Bob a decidir quien es el ganador.

Input

La entrada contiene varios casos de prueba cada uno descrito en tres líneas. La primera línea contiene un número entero N indicando el número de fracciones acordado entre Alice y Bob escritos en un papel ($1 \leq N \leq 100$). La segunda línea contiene N fracciones X_i separados por un espacio cada uno con un numerados n_i y denominador d_i , representando las fracciones que Alice escribió n_i/d_i ($1 \leq n_i, d_i \leq 100$ para $1 \leq i \leq N$). La tercera línea contiene N fracciones Y_i separados por un simple espacio, representando las fracciones que Bob escribió ($1 \leq Y_i \leq 100$ para $1 \leq i \leq N$). La entrada termina con $N = 0$.

Output

Para cada caso de prueba imprimir una sola línea "Alice" si Alice gana el juego, o la cadena "Bob" si Bob gana el juego, en el caso de empate imprimir "=".

Ejemplo:

Input:

```
3
1/12 7/12 3/12
3/7 2/8 10/5
5
2/4 6/3 8/2 1/3 15/3
3/5 9/11 4/2 56/4 33/3
1
7/2
4/5
0
```

Output:

```
Alice
Bob
=
```

Problema 6: Cola de Primos (16417. AI_COL)

Estás en el medio de una cola para comprar entradas de un concierto pero existen muchas personas, por lo mismo el día anterior los organizadores decidieron vender las entradas solo a personas que estén en medio de dos números primos. Los organizadores repartieron números acorde a la llegada a la fila, si tu número es el 4 y las personas que tiene el número 3 delante tuyo y el número 5 detrás tuyo están presentes tu puedes comprar la entrada pero si la persona con el número 5 no esta presente y tu estas en medio de 3 y 6 ellos no te venderán la entrada porque 6 no es un número primo.

Input

La entrada consiste en tres líneas, la primera línea contiene un número entero N indicando el número de personas en la cola ($3 \leq N \leq 100000$). La segunda línea contiene N números enteros ordenados o desordenados X_i separados por un simple espacio, cada uno representa a una persona ($1 \leq X_i \leq 100000$ para $1 \leq i \leq N$). La tercer línea contiene un número entero Q ($1 \leq Q \leq 100000$) que representa tu numero en la cola.

Output

Imprimir una sola línea conteniendo la palabra "Yes" si puedes comprar entradas, o la palabra "No" si no puedes comprar entradas.

Ejemplos

Input:	Output:
5 1 2 4 3 5 4	Yes
6 4 2 9 7 3 6 6	No
6 4 2 9 7 3 6 2	No
3 5 7 2 5	No

Problema 7: Batuke

Batuke es un perro con comportamiento repetitivo que tiene una rutina particular cuando sale a dar un paseo por su barrio. El barrio también es particular; una matriz cuadrada de $N \times N$, que Batuke sabe recorrer con una rutina en espiral: cuando comienza hace: 1 casilla a derecha, una abajo, luego dos a izquierda, dos arriba, luego 3 a derecha, 3 abajo, luego 4

A Batuke, Lucas (su dueño) lo lleva en auto a la casilla de partida, y Batuke siempre sigue su rutina, hasta que recorre todas las casillas de su barrio.

Por ejemplo si su barrio es de tamaño $N=4$, el mismo se numera de la siguiente forma:

1	2	3	4	y el recorrido de Batuke, partiendo de la casilla 1,1 (la casilla superior izquierda es la 0,0), es: 6,7,11,10,9,5,1,2,3,4,8,12,16,15,14,13 En este caso su rutina lo hace caminar 16 casillas.
5	6	7	8	
9	10	11	12	
13	14	15	16	

Pero Batuke no distingue los límites del barrio y nunca se sale de su rutina (aunque tenga que caminar más para recorrer todas las casillas de su barrio). Si la casilla de partida es la 2,2; el recorrido en espiral (sólo se muestran las casillas del barrio) es el siguiente:

11,12,16,15,14,10,6,7,8,13,9,5,1,2,3,4. En este caso su rutina lo hace caminar 24 casillas.

Lucas sabe que estudiás ISI, y te pide que dado un barrio de N filas x columnas, y una casilla donde él deja a Batuke, le informés el recorrido en espiral, y el total de cuerdas que caminó Batuke.

Entrada:

La entrada consiste de tres números enteros: N, F, C . $N, 2 < N \leq 10$ indica la cantidad de filas que tiene el barrio (con celdas numeradas de 1 a $N \times N$, por filas y de izquierda a derecha). F y C , indican la fila y columna de la casilla donde Batuke comienza su recorrido.

Salida:

La salida consiste de una línea conteniendo una lista con las casillas del barrio ordenadas según el recorrido, y separadas por un espacio en blanco. La línea siguiente contiene un número entero que muestra la cantidad de cuerdas que caminó Batuke.

Input	Output
4 1 1	6 7 11 10 9 5 1 2 3 4 8 12 16 15 14 13 16
4 3 3	16 15 11 12 14 10 6 7 8 13 9 5 1 2 3 4 46