

# SAN FRANCISCO



## Problemas

Categoría

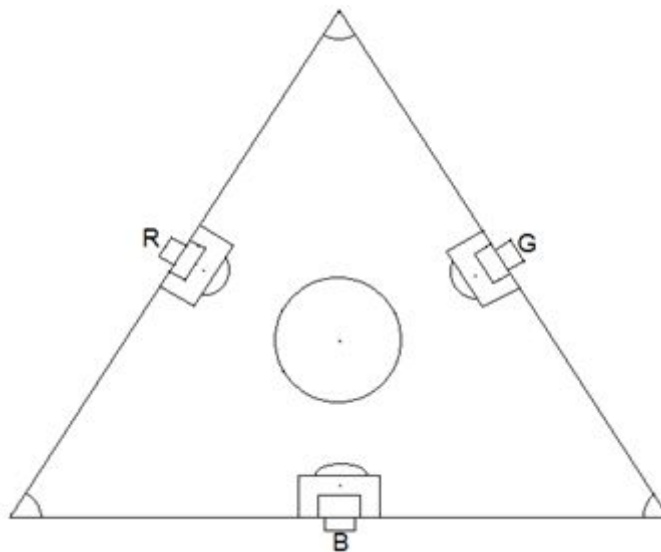
# A

22 de noviembre de 2018

## A. Tribol

En la ciudad de Triangulândia, el sueño más grande de sus habitantes era construir un estadio de fútbol, pero allí todos los terrenos son triangulares. Si se fuera a construir un estadio rectangular, una gran parte del terreno se desperdiciaría. Entonces, los hermanos Hipo y Tenusa tuvieron una gran idea: crear un nuevo deporte, derivado del fútbol, pero que se juegue en un campo triangular, y bautizaron al nuevo juego como Tribol. Las reglas eran simples: tres equipos juegan al mismo tiempo: el Equipo Rojo (Red), Equipo Verde (Green) y Equipo Azul (Blue).

- Cada juego tiene una duración de 30 minutos.
- El equipo que anota un gol en el arco del oponente que se ubica a su derecha, anota un gol normal.
- El equipo que anota un gol en el arco del oponente que se ubica a su derecha, anota un gol que vale doble.
- Si los tres equipos anotan la misma cantidad de goles, hay un trempate.
- Si dos de los equipos que anotan más goles tienen la misma cantidad, hay un empate, y el resultado del juego se decide por penales.
- Gana el equipo que hace mayor cantidad de goles. Los equipos se organizan en el campo como se muestra en la imagen siguiente.



Por ejemplo:

- Si el Equipo Verde (Green) anota un gol en el arco del Equipo Azul (Blue) y otro gol en el arco del Equipo Rojo (Red), obtiene un total de dos goles.
- Si el Equipo Azul (Blue) anota dos goles en el arco del Equipo Verde (Green), obtiene un total de dos goles.

- Si el Equipo Azul (Blue) anota dos goles en el arco del Equipo Rojo (Red), obtiene un total de dos goles.

## Entrada

Existirán varios casos de prueba. Primero se ingresará un entero **C** que representa la cantidad de casos de prueba. Cada caso de prueba comienza con un entero **P** que representa la cantidad de goles convertidos durante el partido. Seguirán **P** líneas cada una con dos letras **M** y **S**, indicando respectivamente la letra (R, G o B) del equipo que convirtió el gol y la letra del equipo al cual se le convirtió,

## Salida

Para cada caso de prueba imprimir el nombre del equipo ganador (red, green o blue) o el resultado del juego (empate o trempate) en base a los goles convertidos.

Ejemplo de entrada	Ejemplo de salida
<pre>3 3 G R R B B G 5 G B G R B R B G R G 1 B G</pre>	<pre>trempate empate blue</pre>

## B. Equilibrio de Paréntesis I

Considerando una expresión con paréntesis, imprima un mensaje informando si la cantidad de paréntesis es correcta o incorrecta, sin considerar el resto de la expresión. Ejemplo:

$a+(b*c)-2-a$	es correcto
$(a+b*(2-c)-2+a)*2$	es correcto
$(a*b-(2+c)$	es incorrecto
$2*(3-a))$	es incorrecto
$)3+b*(2-c)($	es incorrecto

Resumiendo, todos los paréntesis cerrados debe tener un paréntesis abierto y no es posible un paréntesis de cierre sin un paréntesis abierto previamente, y la cantidad de paréntesis cerrados y abiertos deben ser iguales.

### Entrada

El archivo de entrada contiene **N** expresiones ( $1 \leq N \leq 10000$ ), cada una con hasta 1000 caracteres.

### Salida

La salida debe mostrar **correct** or **incorrect** para cada caso de prueba de acuerdo con la reglas anteriores.

Ejemplo de entrada	Ejemplo de salida
$a+(b*c)-2-a$	correct
$(a+b*(2-c)-2+a)*2$	correct
$(a*b-(2+c)$	incorrect
$2*(3-a))$	incorrect
$)3+b*(2-c)($	incorrect

## C. One-Two-Three

Tu hermano pequeño acaba de aprender a escribir *one*, *two*, *three* (uno, dos, tres), en Inglés. Él ha escrito varias de estas tres palabras en un papel, y es tu tarea reconocerlas. Debes tener en cuenta que tu hermano es sólo un niño, por lo cual comete pequeños errores: para cada palabra habrá al menos una letra mal escrita. El largo de la palabra siempre es correcto. Se garantiza que cada letra que él escribió está en minúscula y que cada palabra posee una única interpretación posible.

### Entrada

La primer línea contiene un entero **N** que indica el número de palabras que tu pequeño hermano ha escrito. Cada una de las **N** líneas siguientes contiene una palabra escrita enteramente en letras minúsculas. Cada palabra satisface los requerimientos enunciados anteriormente: al menos una letra será incorrecta, pero el largo de la palabra siempre será el correcto. No habrá más de 1000 palabras en la entrada.

### Salida

Para cada caso de prueba, imprimir el valor numérico de la palabra.

Ejemplo de entrada	Ejemplo de salida
3 owe too theee	1 2 3

## D. Máquina de Café

El nuevo edificio de la Asociación Organizada de Responsables Programadores (AORP) tiene 3 pisos. En ciertas épocas del año, los empleados de la AORP beben mucho café. Por ello, la dirección de la AORP decidió regalarles una nueva máquina de espresso. Esta máquina debe ser instalada en uno de los 3 pisos, pero la instalación debe ser hecha de forma que las personas no pierdan mucho tiempo subiendo y bajando escaleras.

Cada empleado de la AORP bebe 1 café expreso por día. Él necesita ir desde el piso donde trabaja hasta el piso donde está la máquina y volver a su puesto de trabajo. Toda persona tarda 1 minuto para subir o bajar un piso. Como la AORP se preocupa mucho de la eficiencia, quiere posicionar la máquina para minimizar el tiempo total gastado subiendo y bajando escaleras.

Tu tarea es ayudar a la dirección a posicionar la máquina para minimizar el tiempo total gastado por los empleados subiendo y bajando escaleras.

### Entrada

La entrada consiste en 3 números,  $A_1, A_2, A_3$  ( $0 \leq A_1, A_2, A_3 \leq 1000$ ), uno por línea, donde  $A_i$  representa el número de personas que trabajan en el  $i$ -ésimo piso.

### Salida

El programa debe imprimir una única línea de salida conteniendo el número total de minutos a que serán gastados, teniendo en cuenta a todos los miembros, con el mejor posicionamiento posible de la máquina.

Ejemplo de entrada	Ejemplo de salida
10 20 30	80
10 30 20	60
30 10 20	100

## E. Volleyball

A un entrenador de voleibol le gustaría llevar estadísticas sobre su equipo. En cada juego, realiza un seguimiento de la cantidad de servicios, bloqueos y ataques de cada jugador, así como cuántos de estos servicios, bloqueos y ataques fueron exitosos (puntos anotados). Se necesita un programa que muestre, con dos puntos decimales, cuál es el porcentaje total de servicios, bloqueos y ataques exitosos de todo el equipo.

### Entrada

La entrada comienza con el número de jugadores  $N$  ( $1 \leq N \leq 100$ ), seguido de los nombres de cada uno de estos jugadores. Debajo del nombre de cada jugador, se presentan dos líneas de enteros. En la primera línea, los números representan los servicios (**S**), los bloqueos (**B**) y los intentos de ataque (**A**) para el jugador específico ( $0 \leq S, B, A \leq 10000$ ). En la segunda línea hay el número de estos servicios (**S1**), bloqueos (**B1**) y ataques (**A1**) que tuvieron éxito ( $0 \leq S1 \leq S$ ;  $0 \leq B1 \leq B$ ;  $0 \leq A1 \leq A$ ).

### Salida

La salida debe contener el porcentaje, con dos decimales de precisión, del total de servicios, bloqueos y ataques exitosos logrados por todo el equipo, tal como se muestra en el ejemplo.

Ejemplo de entrada	Ejemplo de salida
<pre> 3 Renan 10 20 12 1 10 9 Jonas 8 7 1 2 7 0 Edson 3 3 3 1 2 3 </pre>	<pre> Pontos de Saque: 19.05 %. Pontos de Bloqueo: 63.33 %. Pontos de Ataque: 75.00 %. </pre>

## F. Batuke

Batuke es un perro con conducta repetitiva que tiene una rutina particular cuando sale a pasear por su barrio. El barrio también es particular; una matriz de  $N \times N$  que Batuke recorre con una rutina en espiral: Batuke comienza caminando: una celda a la derecha, una abajo, luego dos a la izquierda, dos arriba, luego 3 a la derecha, 3 abajo, luego 4...

Lucas (el dueño de Batuke), lo lleva en auto hasta la celda de comienzo, y Batuke siempre sigue su rutina hasta recorrer todas las celdas en el barrio.

Por ejemplo, si el barrio tiene tamaño  $N=4$ , las celdas están numeradas como sigue:

1	2	3	4	<p>y el recorrido de Batuke, partiendo de la casilla 1,1 (la casilla superior izquierda es la 0,0) es:</p> <p>6,7,11,10,9,5,1,2,3,4,8,12,16,15,14,13</p> <p>En este caso Batuke camina 24 celdas en total.</p>
5	6	7	8	
9	10	11	12	
13	14	15	16	

Pero Batuke no entiende de límites y siempre sigue su rutina (aún si tiene que caminar varias celdas más para recorrer todas las celdas de su barrio). Si la celda de comienzo es la 2,2; el recorrido en espiral (sólo se muestran las celdas del barrio) es: 11,12,16,15,14,10,6,7,8,13,9,5,1,2,3,4.

En este caso Batuke camina 24 celdas en total.

### Entrada

La entrada consiste de tres números enteros:  $N$  ( $2 < N \leq 10$ ),  $F$  y  $C$ .  $N$  es el número de filas (y columnas) del barrio (cuyas celdas están numeradas en  $1..N \times N$  según filas, de arriba a abajo y de izquierda a derecha).  $F$  y  $C$  indican fila y columna donde Batuke comienza su paseo.

### Salida

La salida consiste de una línea conteniendo una lista con las casillas del barrio ordenadas según el recorrido que hace Batuke, y separadas por un espacio en blanco. La línea siguiente contiene un número entero que muestra la cantidad de celdas que visitó Batuke (que puede ser diferente a la cantidad de celdas del barrio donde vive, ya que Batuke nunca cambia su conducta, y eso lo lleva en ocasiones a recorrer celdas de los barrios vecinos).



Ejemplo de entrada	Ejemplo de salida
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

## G. Rot13

Escriba un programa que reciba un mensaje secreto y devuelva este mensaje codificado. El mensaje es encriptado con un cifrado llamado Rot13. Rot13 funciona de la siguiente manera, por cada letra se le asigna un índice, basado en su ubicación en el alfabeto a = 1, b = 2, c = 3, ..., z = 26. Cada letra del mensaje deberá ser movida 13 lugares a la derecha. Si se alcanza el final del alfabeto, el conteo comienza en el primer carácter del alfabeto. Por ejemplo el carácter "a" (citado solo para ejemplificar) sería mapeado a "n", "y" a "l", "f" a "s" y así sucesivamente. El mismo proceso también se aplica para mayúsculas. Cualquier carácter no alfabético debe ser transferido a la salida sin cambios.

### Entrada

La entrada consiste en algunos casos de pruebas. Cada caso está formado por una línea que contiene un mensaje con al menos 1 hasta 50 caracteres que incluyen letras, números y los siguientes símbolos: !@#\$%^&\*()-\_+[]{}|;':./<>?"

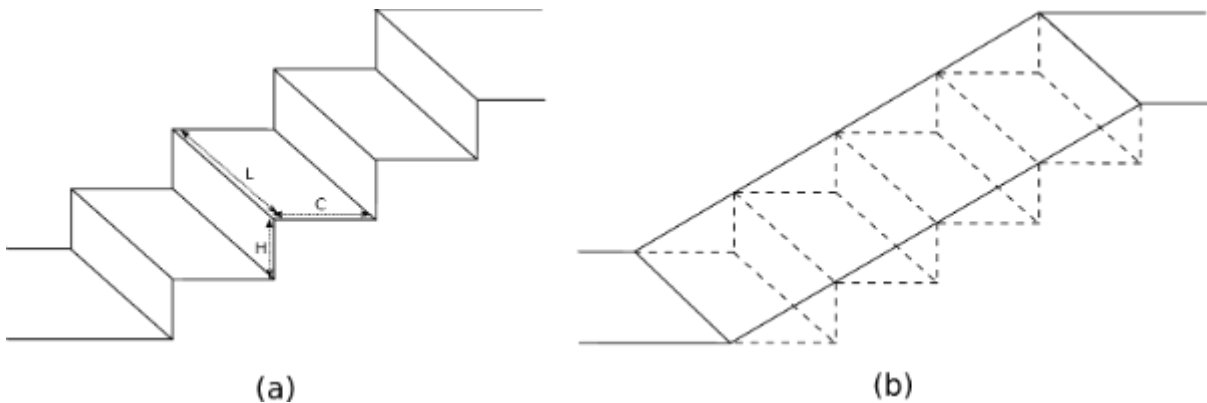
### Salida

Por cada línea de la entrada, imprimir un mensaje en la salida codificado en el método rot13.

Ejemplo de entrada	Ejemplo de salida
<pre>z a x Hello world! This is a test message. abc123&lt;&gt;,xyz?* \$</pre>	<pre>m n k Uryyb jbeyq! Guvf vf n grfg zrffntr. nop123&lt;&gt;,klm?* \$</pre>

## H. Escalera del FNDI

El edificio del Departamento Federal de Islas Naturales (FDNI) está siendo ampliamente remodelado para mejorar su accesibilidad. Hay una escalera con  $N$  escalones. Cada escalón tiene  $H$  centímetros de alto,  $C$  centímetros de largo y  $L$  centímetros de ancho. La figura (a) muestra una escalera con  $N=4$  escalones.



Para que el edificio sea más accesible, la dirección del FDNI decidió ubicar una rampa sobre la escalera. La rampa es dura y tiene forma rectangular. Será ubicada sobre la escalera de manera que todos los escalones queden cubiertos, tal como lo indica la figura (b).

Tu tarea es, dado la cantidad de escalones y sus medidas, determinar el área total de la superficie de la rampa.

### Entrada

La entrada contiene varios casos de prueba. La primer línea de cada caso de prueba contiene un entero  $N$  ( $1 \leq N \leq 1000$ ), que indica la cantidad de escalones en la escalera. La segunda línea contiene tres enteros  $H$ ,  $C$  and  $L$  ( $1 \leq H, C, L \leq 100$ ), que indican las medidas de cada escalón, en centímetros. La entrada termina con el fin de archivo (end-of-file o EOF).

### Salida

Para cada caso de prueba, imprimir una línea con el área total de la superficie de la rampa, en metros cuadrados. Redondear e imprimir este valor con exáctamente 4 decimales.

Ejemplo de entrada	Ejemplo de salida
4 3 4 30	0.0600

## I. Fibonot

La sucesión de Fibonacci es una de las sucesiones más famosas del mundo. Cada término de la sucesión de Fibonacci se forman a partir de la suma de los 2 términos anteriores, a excepción de los 2 primeros términos que son, por definición, iguales a 1. Por ejemplo, los primeros 9 términos de la sucesión son:

1, 1, 2, 3, 5, 8, 13, 21, 34 ...

Sin embargo, no nos interesa encontrar los términos de la sucesión de Fibonacci, ¡sino los términos de la sucesión de Fibonot!

La sucesión de Fibonot está formada por los números que no pertenecen a la sucesión de Fibonacci. Más específicamente, enteros positivos distintos de cero en orden ascendente.

Aquí están los primeros términos de la sucesión de Fibonot:

4, 6, 7, 9, 10, 11, 12, 14, 15 ...

Tu tarea es encontrar el K-ésimo término de la sucesión de Fibonot.

### Entrada

La entrada consiste en un único entero  $K$  ( $1 \leq K \leq 10^5$ ) que especifica el índice del término de la sucesión de Fibonot que se desea obtener.

### Salida

Un único entero con el K-ésimo término de la sucesión de Fibonot.

Ejemplo de entrada	Ejemplo de salida
1	4
3	7
6	11