

Modulo 2. Arreglos

Statement is not available on English language

1. Acumulados al inverso

time limit per test: 1 second
memory limit per test: 256 megabytes

Dada una secuencia de N números enteros, necesitamos saber cuál es el acumulado (sumatoria) de los k últimos, con k desde 2 hasta N .

Input

La entrada comienza con una línea que contiene un valor entero positivo N tal que ($1 \leq N \leq 5000$). Luego sigue una línea con N valores enteros, cada uno no en el intervalo $[-100, 100]$, separados entre sí por un espacio en blanco.

Output

La salida debe contener $N - 1$ líneas con los acumulados correspondientes.

Example	
input	<div>Copy</div>
4 10 20 30 40	
output	<div>Copy</div>
70 90 100	

Statement is not available on English language

2. Juego trisolariano

time limit per test: 1 second
memory limit per test: 256 megabytes

El problema de los tres cuerpos es una novela de ciencia ficción del escritor chino Liu Cixin que hace parte de la trilogía titulada El recuerdo del pasado de la Tierra donde también están El bosque oscuro y El fin de la muerte. El título hace referencia al problema de astrodinámica que consiste en determinar, en cualquier instante, las posiciones y velocidades de tres cuerpos con diferente masa, sometidos a atracción gravitacional mutua y partiendo de unas posiciones y velocidades dadas.

Si ya leíste por lo menos el primer libro, sabrás que los trisolarianos son una raza extraterrestre con muchas particularidades. Una de ellas, presumiblemente, es que les gusta un juego con tres participantes. Por turnos, cada uno elije un valor entero entre uno y ocho (también, presumiblemente, tienen 8 dedos y un sistema numérico octal). Si un jugador elije un valor par, y la suma de los tres valores da par, ese jugador gana un punto. Igual ocurre si elije un valor impar y la suma da impar.

Dada una serie de N turnos, ¿les ayudarías a determinar el puntaje final? Al primer jugador se le denomina SO, al segundo LAR, y al tercero IS.

Input

La entrada comienza con una línea que contiene un valor entero positivo N tal que ($1 \leq N \leq 12000$) que corresponde a la cantidad de turnos. Luego siguen tres líneas, la primera para las N jugadas del jugador SO, la segunda para las N jugadas del jugador LAR, y la tercera para las N jugadas del jugador IS. En los tres casos los valores se encuentran separados entre sí por una coma y un espacio en blanco.

Output

La salida debe contener una única línea con el mensaje SO:X, LAR:Y, IS:Z con los puntajes correspondientes.

Example	
input	<div>Copy</div>
4 2, 3, 5, 4 3, 4, 5, 6 2, 4, 6, 8	
output	<div>Copy</div>
SO:2, LAR:2, IS:2	

Statement is not available on English language

3. Criptografía para dummies

time limit per test: 1 second
memory limit per test: 256 megabytes



Fuente: https://commons.wikimedia.org/wiki/File:Don_Adams_Barbara_Feldon_Get_Smart_1965.JPG

La serie televisiva de los 60's Get Smart, traducida en Latinoamérica como Super agente 86 era una parodia de las películas de espías protagonizada por Don Adams, como Maxwell Smart o el agente 86, y Barbara Feldon, como la agente 99.

En la serie había dos agencias de espionaje y contraespionaje: por un lado CONTROL donde estaban 86 y 99, que imitaba a la CIA estadounidense, y por otro KAOs, que imitaba la KGB soviética.

Obviamente el trabajo de ambas agencias era mantener sus secretos y develar los de su contraparte por lo que supongamos que una de ellas usaba el siguiente mecanismo para encriptar mensajes. Considerando únicamente las letras $a - z$ en minúsculas, más los dígitos 0 - 9, más el guion bajo para representar el espacio en blanco, un mensaje M pasa por dos procesos:

- M se transforma en M^* intercambiando cada carácter en posición impar por el carácter inmediatamente posterior
- M^* se transforma en M^{**} intercambiando cada carácter por su imagen especular. Esto es, el carácter en la primera posición se intercambia por el de la última, el de la segunda por el de la penúltima y así sucesivamente.

Por ejemplo para $M = "h o l a _ m u n d o"$

- M^* sería = "o h a l m _ n u o d"
- M^{**} sería = "d o u n _ l a h o"

Dada una serie de mensajes encriptados en este sistema, ¿harías un programa para descryptarlos?

Input

La entrada comienza con una línea que contiene un valor entero positivo C que corresponde a los casos de prueba, no más de 50. Luego siguen C líneas para cada caso, cada una con hasta 2000 caracteres separados entre sí por un espacio en blanco.

Output

Por cada caso de prueba la salida debe contener una línea con el mensaje correspondiente descryptado sin espacios en blanco.

Example	
input	<div>Copy</div>
2 i a a f g r t o i p c r o i n a m _ c e n e _ e n v i 6 _ _ 8 t e e n a g	
output	<div>Copy</div>
criptografia agente_86_viene_en_camino	

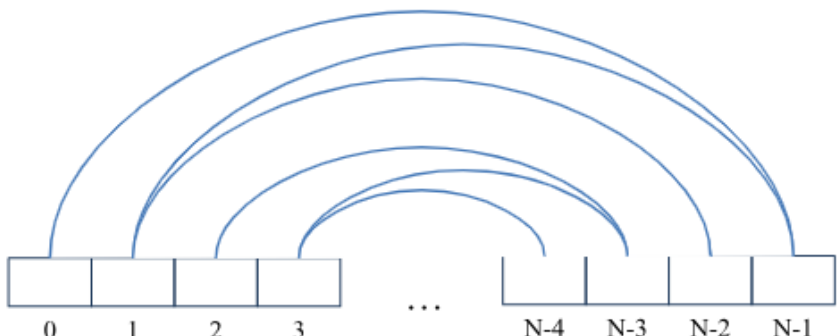
Statement is not available on English language

4. Recorrido arcoiris

time limit per test: 1 second
memory limit per test: 256 megabytes

Cuando tenemos un arreglo con N elementos y necesitamos recorrerlo por alguna razón, lo más natural es hacerlo de izquierda a derecha, ¿verdad? Esto es, desde el índice 0 hasta el $N - 1$ aumentando de a uno.

Bueno, pues en un recorrido arcoiris también se pasa por todos los elementos una única vez, pero en un orden bastante peculiar. Como se muestra en la figura, se parte del índice 0, de allí se va para el $N - 1$, de allí para el 1, de allí se va para el $N - 2$, de allí para el 2, de allí se va para el $N - 3$, y así sucesivamente hasta llegar al "centro" del arreglo.



Input

La entrada comienza con una línea que contiene un valor entero positivo N ($2 \leq N \leq 5000$) que corresponde al tamaño del arreglo. Luego sigue una línea con N valores, todos caracteres $A - Z$, separados entre sí por una coma y un espacio en blanco.

Output

La salida debe contener una única línea con la concatenación de todos los elementos del arreglo siguiendo el recorrido descrito.

Example	
input	<div>Copy</div>
8 S, R, M, G, O, A, A, A	
output	<div>Copy</div>
SARAMAGO	

Statement is not available on English language

5. Arte callejero

time limit per test: 1 second
memory limit per test: 256 megabytes

Alonso y sus amigos se ganan la vida haciendo presentaciones a los transeúntes en los semáforos de la ciudad. A veces hacen malabares, a veces baile y a veces incluso un poco de magia. Sus espectáculos, aunque breves, gustan mucho por lo que suelen recoger una buena cantidad de dinero. Lo curioso es que a la hora de repartirlo son bastante excéntricos: lo dejan al azar. Más específicamente lo que hacen es que al terminar una jornada juntan todos los billetes y monedas en una enorme pila sin ningún orden en particular (por cuestiones físicas los billetes deben ir en la base de la pila pero, en el fondo, esto no tiene importancia). Luego cada integrante va sacando de a una única moneda o billete en la cima de la pila hasta que esta quede vacía.

Claramente todos reciben aproximadamente la misma cantidad de monedas o billetes, pero no necesariamente el mismo monto total. Lo que ellos dicen y que estadísticamente es correcto, es que, dada la aleatoriedad en la pila y a que cada día varían también de manera aleatoria el orden en que se acomodan los integrantes a la hora de repartir, quien un día obtiene menos en otro obtiene más, entonces a la larga todos obtienen prácticamente lo mismo.

Input

La entrada comienza con una línea que contiene un valor entero positivo C que corresponde a los casos de prueba, no más de 10. Cada caso de prueba comienza con una línea que contiene dos valores enteros positivos separados por un espacio en blanco: la cantidad M de integrantes y la cantidad N de monedas o billetes recolectados ($2 \leq M \leq 50$, $1 \leq N \leq 5000$). Posterior a esa sigue otra línea con las denominaciones de las N monedas o billetes separadas entre sí por un espacio en blanco. Esas denominaciones no corresponden necesariamente a un sistema monetario específico, solo se tratan de valores enteros positivos inferiores a 10000.

Output

Por cada caso de prueba la salida debe contener una línea con la diferencia entre lo que recibe en total quien recibe más y lo que recibe quien recibe menos.

Example	
input	<div>Copy</div>
2 2 4 10 20 40 30 3 5 60 40 30 20 10	
output	<div>Copy</div>
0 50	

Statement is not available on English language

6. Serpientes y escaleras

time limit per test: 1 second
memory limit per test: 256 megabytes



Fuente: <https://www.publicdomainpictures.net/es/view-image.php?image=298032&picture=juego-de-serpientes-y-escaleras>

En este popular juego de mesa los jugadores comienzan en la primera casilla del tablero y se turnan para lanzar un dado que les indicará la cantidad de casillas que deben avanzar en sentido ascendente. Si al finalizar un movimiento un jugador cae en una casilla donde comienza una escalera, sube por ella hasta la casilla donde ésta termina. Si, por el contrario, cae en una en donde comienza la cabeza de una serpiente, desciende por ésta hasta la casilla donde finaliza su cola.

En la versión de programadores, el tablero se reemplaza por un arreglo de N números enteros. Iniciando en el primer índice del arreglo, el jugador debe "saltar" tantos índices hacia adelante o hacia atrás dependiendo como sea el valor del arreglo en ese índice (si es positivo hacia adelante o si es negativo hacia atrás). Al llegar a ese índice se repite el proceso sucesivamente hasta que ocurra una de dos cosas: o que se cierre un ciclo (que en algún punto se salte a un índice por el que ya había pasado) o hasta que salte a un índice que esté por fuera del arreglo. Entendiendo este funcionamiento, lo que se requiere es saber cuántos saltos se requieren para que termine el juego.

Input

La entrada comienza con una línea que contiene un valor entero positivo C ($1 \leq C \leq 100$) que corresponde a la cantidad de casos. Cada caso comienza con una línea que contiene un valor entero positivo N ($1 \leq N \leq 5000$) que corresponde al tamaño del tablero, seguido de una línea con N valores enteros no nulos en el rango $[-100, 100]$ separados entre sí por un espacio en blanco.

Output

La salida debe contener C líneas, cada una con la cantidad de saltos en cada caso.

Example	
input	<div>Copy</div>
3 5 1 1 1 1 4 3 -1 4 -2 6 4 -3 5 -2 -1 6	
output	<div>Copy</div>
5 3 4	

