

## SUMAS ADYACENTES

### DESCRIPCIÓN

Dada una sucesión  $A$  con  $n$  números ordenada de manera no descendente, es posible encontrar otra sucesión  $B$  con  $n-1$  números sumando los números adyacentes de  $A$ . Por ejemplo, la siguiente sucesión tiene 6 números:

$$A=(3, 5, 5, 6, 7, 9)$$

Y es posible formar otra sucesión  $B$  con 5 números sumando los números adyacentes de  $A$ :

$$B=(8, 10, 11, 13, 16)$$

Para cada sucesión  $A$  existe SÓLO UNA sucesión  $B$ , sin embargo, para cada sucesión  $B$  puede haber varias sucesiones que *podrían ser*  $A$ . De manera más precisa diremos que una sucesión  $A$  genera a una sucesión  $B$  si:

$$A=(a_1, a_2, \dots, a_n),$$

$$a_1 \leq a_2 \leq \dots \leq a_n,$$

$$\text{y } B=(a_1+a_2, a_2+a_3, \dots, a_{n-1}+a_n)$$

### PROBLEMA

Dada una sucesión  $B$ , de entre todas las sucesiones  $A$  que pudieron generarla, encuentra aquella cuyo primer número sea el mayor posible.

### ENTRADA

Tu programa deberá leer de la entrada estándar los siguientes datos:

- En la primera línea el número entero  $n$
- En la segunda línea  $n-1$  números enteros separados por espacios representando la sucesión  $B$

### SALIDA

Tu programa deberá escribir en salida estándar lo siguiente:

- Si hay una solución con el primer número mayor, deberás imprimir una única línea con  $n$  números enteros separados por un espacio representando la sucesión  $A$ .
- Si no existe una sucesión  $A$  que cumpla con las características simplemente imprime -1

### CONSIDERACIONES

$$2 \leq n \leq 5,000,000$$

$$1 \leq B_i \leq 1,073,741,824 \quad (\text{este número es igual a } 2^{30})$$

### EJEMPLOS

ENTRADA	SALIDA
6 8 10 11 13 16	3 5 5 6 7 9
4 8 10 9	-1

### REQUERIMIENTOS DE EJECUCION

Para obtener puntos en este problema, además de entregar la solución correcta tu programa deberá ejecutarse en un tiempo menor a 1 segundo.



## SERPIENTES Y ESCALERAS

### DESCRIPCIÓN

Seguramente has jugado serpientes y escaleras. Si no lo has hecho, se juega en un tablero con casillas que van del 1 al  $N$ . En cada turno, se tiran los dados y se avanza un número de casillas igual a la cantidad que sumen los dados. Al terminar el movimiento se puede caer en una casilla que sea el extremo de una serpiente o una escalera, en dicho caso sucederá alguna de las siguientes opciones:

- Si terminas tu tiro en el extremo inferior de una escalera, automáticamente avanzarás a la casilla en donde termina la escalera.
- Si terminas tu tiro en la cola de una serpiente, automáticamente retrocederás a la casilla en donde está la cabeza de la serpiente.
- Si terminas tu tiro en el extremo superior de una escalera, o en la cabeza de una serpiente, no pasa nada.

Cuando se llega la casilla  $N$  del tablero, aún cuando no sea de manera exacta, el juego termina.

### PROBLEMA

Dado un tablero de serpientes y escaleras y un conjunto de tiros de dados, determina cuantas formas distintas hay de terminar el juego si puedes seleccionar el orden en el que caen los tiros. Dos juegos se consideran distintos si sus secuencias de valores de tiros son distintas.

### ENTRADA

Tu programa deberá leer de la entrada estándar los siguientes datos:

- En la primera línea un número  $N$  que determina el largo del tablero.
- En la segunda línea un número  $S$ , la cantidad de saltos (serpientes o escaleras) que hay en el tablero.
- En las siguientes  $S$  líneas dos números separados por espacio que indican la casilla inicio y la final del salto.
- En la siguiente línea un número  $D$  que indica la cantidad de tiros de dado que se te van a dar.
- En la última línea hay  $D$  números separados por espacio con los distintos tiros de dado  $d_i$  que puedes utilizar.

**NOTA:** En ningún caso, un salto terminará en el inicio de otro salto, es decir, ninguna casilla te llevará a dos saltos consecutivos, ni ninguna casilla será el inicio de más de un salto.

### SALIDA

Tu programa deberá escribir a la salida estándar un único número que indique la cantidad de formas distintas que hay de terminar el juego.

### CONSIDERACIONES

$20 \leq N \leq 250$

$1 \leq S \leq 100$

$1 \leq D \leq 10$

$2 \leq d_i \leq 12$  El valor de cada tiro de dado.

### EJEMPLO DE ENTRADA

ENTRADA	SALIDA
40	8
2	
4 25	
30 22	
5	
10 7 10 4 5	

### EXPLICACION

Las posibles formas de terminar son: {4,7,10} {4,10,5} {4,10,7} {4,10,10} {4,5,10,10} {4,7,5,10} {4,5,7,10,10} {4,5,10,7,10}. ES MUY IMPORTANTE NOTAR QUE AUNQUE HAY DOS TIROS DE DADO "10" SÓLO HAY UNA SOLUCION "4,7,10". LO MISMO PASA CON OTRAS SOLUCIONES QUE CONTIENEN EL 10.

### REQUERIMIENTOS DE EJECUCION

Para obtener los puntos de este problema tu programa deberá ejecutarse en un tiempo menor a un segundo.