

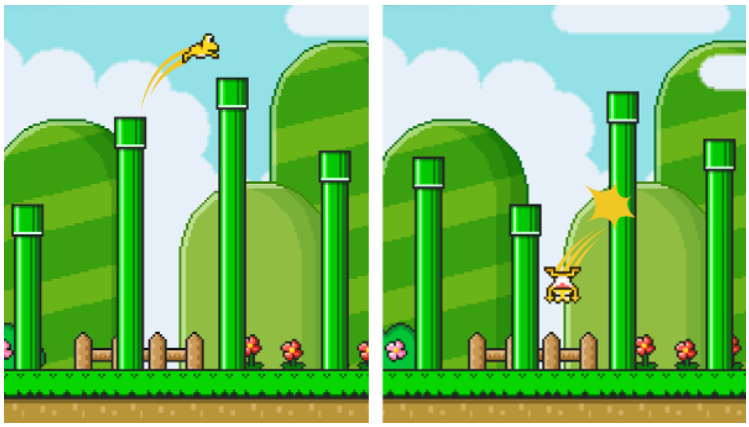
Nivel “AEDD”
Problemas

14 de Octubre de 2016

Problema 1: Salto de Rana

En cada etapa del juego Salto de Rana tenés que asegurarte que tu anfibio, a través de una secuencia de tubos de diferentes alturas, logre pasar al tubo que está más a la derecha. Sin embargo, la rana solo sobrevive si la diferencia de altura de los tubos consecutivos es como máximo de la altura de su salto. Si la altura del tubo siguiente es demasiado alta, la rana golpea el tubo y cae. Si la altura del tubo siguiente es demasiado baja, la rana no sobrevive a la caída. La rana comienza siempre en la parte superior del tubo ubicado más a la izquierda.

En este juego, la distancia entre los tubos es irrelevante, lo que significa que la rana siempre puede llegar al siguiente tubo con un salto.
Usted debe escribir un programa que, dadas las alturas de los tubos y la altura del salto de rana, mostrará si esa etapa del juego puede ser superada o no.



Entrada

La entrada viene en dos líneas. La primera tiene dos números enteros positivos **P** y **N**, la altura del salto de la rana y el número de tubos ($1 \leq P \leq 5$ y $2 \leq N \leq 100$). La segunda línea tiene **N** números enteros positivos que indican las alturas de los tubos de izquierda a derecha. No hay altura mayor que 10.

Salida

La salida consiste de una sola línea. Si la rana puede llegar a la tubería del extremo derecho, imprimir en pantalla "YOU WIN". Si falla, imprimir en pantalla "GAME OVER".

Ejemplos

Entrada	**Salida**
5 10 1 3 6 9 7 2 4 5 8 3	YOU WIN
1 2 2 2	YOU WIN
1 2 1 3	GAME OVER

Problema 2: Flowers Flourish From France

Fiona ha amado siempre la poesía, y recientemente descubrió una forma poética fascinante. Los tautogramas son un caso especial de aliteración, que es la ocurrencia de la misma letra en el principio de las palabras adyacentes. En particular, una oración es un tautograma si todas sus palabras comienzan con la misma letra.

Por ejemplo, las siguientes frases son tautogramas:

- Flowers Flourish from France
- Sam Simmonds speaks softly
- Peter pIckEd pePPers
- truly tautograms triumph

Fiona quiere deslumbrar a su novio con una poesía romántica llena de este tipo de frases. Por favor ayude a Fiona a comprobar si cada frase que escribió es un tautograma o no.

Entrada

Cada caso de prueba viene en una sola línea que contiene una frase que consiste de una secuencia de a lo sumo 50 palabras separadas por espacios. Una palabra es una secuencia de como máximo 20 letras mayúsculas y minúsculas contiguas del alfabeto inglés. Una palabra contiene al menos una letra y una frase contiene al menos una palabra.

El último caso de prueba viene seguido por una línea que contiene sólo un único carácter "*" (asterisco).

Salida

Para cada caso de prueba, la salida es una sola línea que contiene una mayúscula "Y" si la sentencia es un tautograma, o una mayúscula "N" si no lo es.

Ejemplo

Entrada	Salida
Flowers Flourish from France Sam Simmonds speaks softly Peter pIckEd pePPers truly tautograms triumph this is NOT a tautogram *	Y Y Y Y N

Problema 3: Juego de Estudiantes

A Juilherme y Jogerio realmente les gusta jugar juegos matemáticos. Juilherme creó uno nuevo para jugar mientras siguen TecnoMate 2016 en línea.

El juego tiene las siguientes etapas:

1) Juilherme elige un número entero N y Jogerio elige un número entero M.

2) Juilherme y Jogerio deben encontrar dos números primos P1 y P2, los cuales deben ser el número entero más cercano posible si se compara con N y M, respectivamente. Además de eso, P1 debe ser igual o menor a N y P2 debe ser igual o menor a M.

3) El objetivo final del juego es encontrar la multiplicación de P1 y P2. Quien encuentra la respuesta primero será el ganador.

Ellos van a tratar de encontrar la respuesta de la manera más rápida posible, y en ocasiones pueden cometer algún error. Por lo tanto, necesitan un algoritmo que les proporcione la respuesta final del juego, que luego se compara con su respuesta.

Utilizando la información del juego, debes crear un nuevo programa que dado N y M, imprima la respuesta final del juego.

Entrada

La entrada será sólo una línea con N y M (2 <= N, M <= 1000)

Salida

La salida del programa debe ser sólo una línea con la respuesta final del juego.

Ejemplo

Entrada	Salida
10 15	91
50 100	4559

Problema 4: Sudoku

El Sudoku se hizo conocido en todo el mundo muy rápidamente, siendo el pasatiempo más popular en el planeta actualmente. Sin embargo algunas personas, completan la matriz incorrectamente, y no respetan las reglas. Tu tarea es escribir un programa que chequee si una matriz con sus valores es una solución al Sudoku ó no.

La matriz es una matriz de 9x9 enteros. Será considerada una solución al Sudoku, si cada fila y cada columna contiene todos los números entre 1 y 9. Además, si la matriz se particiona en 9 submatrices de 3x3 (como se muestra en la figura), cada una de ellas debe también contener todos los números entre 1 y 9. La siguiente matriz es un ejemplo de solución para el Sudoku.

$$\left(\begin{array}{ccc|ccc|ccc} 1 & 3 & 2 & 5 & 7 & 9 & 4 & 6 & 8 \\ 4 & 9 & 8 & 2 & 6 & 1 & 3 & 7 & 5 \\ 7 & 5 & 6 & 3 & 8 & 4 & 2 & 1 & 9 \\ \hline 6 & 4 & 3 & 1 & 5 & 8 & 7 & 9 & 2 \\ 5 & 2 & 1 & 7 & 9 & 3 & 8 & 4 & 6 \\ 9 & 8 & 7 & 4 & 2 & 6 & 5 & 3 & 1 \\ \hline 2 & 1 & 4 & 9 & 3 & 5 & 6 & 8 & 7 \\ 3 & 6 & 5 & 8 & 1 & 7 & 9 & 2 & 4 \\ 8 & 7 & 9 & 6 & 4 & 2 & 1 & 5 & 3 \end{array}\right)$$

Entrada

Se ingresan varias instancias. La primer< línea de la entrada contiene n > 0, el número de matrices en la entrada. Las líneas siguientes describen n matrices. Cada matriz está descripta por 9 líneas. Estas líneas contienen 9 enteros cada una.

Salida

Para cada instancia, tu programa debe imprimir una línea conteniendo “Instancia k”, donde k es el número de la instancia. En la segunda línea, tu programa debe imprimir “SIM” (portugués de Si) si la matriz dada es una solución para el Sudoku, ó “NAO” (portugués de No) en otro caso. Imprima siempre una línea en blanco luego de cada instancia.

Entrada	**Salida**
2 1 3 2 5 7 9 4 6 8 4 9 8 2 6 1 3 7 5 7 5 6 3 8 4 2 1 9 6 4 3 1 5 8 7 9 2 5 2 1 7 9 3 8 4 6 9 8 7 4 2 6 5 3 1 2 1 4 9 3 5 6 8 7 3 6 5 8 1 7 9 2 4 8 7 9 6 4 2 1 5 3 1 3 2 5 7 9 4 6 8 4 9 8 2 6 1 3 7 5	Instancia 1 SIM Instancia 2 NAO

7 5 6 3 8 4 2 1 9 6 4 3 1 5 8 7 9 2 5 2 1 7 9 3 8 4 6 9 8 7 4 2 6 5 3 1 2 1 4 9 3 5 6 8 7 3 6 5 8 1 7 9 2 4 8 7 9 6 4 2 1 3 5	

Problema 5: Colección de Pomekons



Desde que está oficialmente lanzada pomekon en Brasil, Gabriel está intentando realizar su gran sueño: Ser un maestro pomekon. Su meta es conquistar los 151 pomekon disponibles. Él logró la captura de muchos monstruos, pero en tu ciudad aparecen muchos pomekon repetidos, haciendo que se capture varias veces el mismo pomekon.

Como siempre vé tu mochila llena de libros de programación, Gabriel te pide que hagas un programa que le informe cuántos pomekons faltan para completar la colección.

Entrada

La primera línea del caso de prueba consiste en un entero **N** ($1 \leq \mathbf{N} \leq 10^3$), representando la cantidad de pomekons que Dabriel ya ha capturado.

Las siguientes **N** líneas consisten en cadenas de caracteres **S** ($1 \leq \mathbf{|S|} \leq 10^3$) representando el nombre de cada pomekon. El nombre de cada pomekon consiste solamente de letras mayúsculas y minúsculas.

Salida

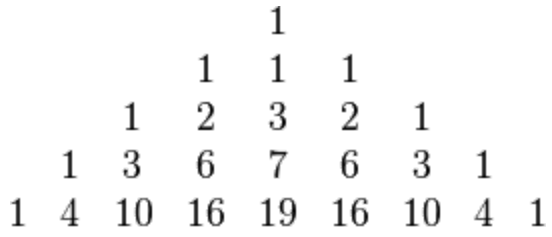
Debe imprimir: "Falta(m) **X** pomekon(s).", Donde **X** es la cantidad de pomekons no capturados.

Ejemplos:

Entrada	**Salida**
7 Charmander Caterpie Pidgeot Rattata Zubat Zubat Zubat	Falta(m) 146 pomekon(s).
8 Zubat Zubat Zubat Zubat Zubat Zubat Zubat Zubat	Falta(m) 150 pomekon(s).

Problema 6: Triángulo Trinomial

El triángulo trinomial es un triángulo numérico de coeficientes trinomiales. Este triángulo puede ser obtenido a partir de una fila que contiene un único "1", la siguiente fila contiene tres "1" y cada elemento de las filas siguientes se calcula sumando los elementos que están arriba a la izquierda, inmediatamente arriba y arriba a la derecha:



La primera fila del triángulo trinomial se numera con cero, la segunda fila es la número 1 y así sucesivamente.

Tu tarea es, dado un número de fila *R*, escribir un programa que muestre la suma de sus elementos. Por ejemplo, la suma de los elementos de la fila 2 es 9 = 1 + 2 + 3 + 2 + 1.

Entrada

La entrada es el número de fila *R* ($0 \leq R \leq 20$).

Salida

La salida es la suma de todos los elementos de la fila *R*. No olvide el caracter de fin de línea luego de exhibir la suma.

Ejemplos:

Entrada	Salida
0	1
1	3
2	9

Problema 7: Campo de Lombrices

Las lombrices de tierra son muy importantes tanto para la agricultura como para la materia prima en la producción de alimentos para animales. La Organización para la Bioingeniería de las lombrices de tierra (OBM) es una organización no gubernamental que promueve el aumento de la producción, el uso y la exportación de las lombrices.
Una de las actividades promovidas por la OBM es el mantenimiento de una granja experimental para la investigación de nuevas tecnologías para crear lombrices. En la granja, la superficie dedicada a la investigación es de forma rectangular, y se encuentra dividida en celdas cuadradas del mismo tamaño. En cada celda se crea sólo un tipo de lombriz. Las células se utilizan para probar los efectos sobre la producción de lombrices de las variaciones de especies de lombrices, tipo de suelo, el fertilizante, la humedad, etc. Los investigadores de la OBM mantienen una vigilancia constante del desarrollo de lombrices en cada celda, y tienen una estimación muy precisa de la productividad de cada una de ellas.

81	28	240	10
40	10	100	240
20	180	110	35

Un investigador de la OBM inventó y construyó una máquina cosechadora de lombrices y decidió probarla en esta granja. La máquina tiene la anchura de una celda, y trabaja desde el centro del suelo recogiendo todos las lombrices en la misma, separándolas, limpiándolas y envasádolas para su comercialización. Es decir, esta máquina elimina uno de los pasos de mano de obra intensiva más costosos del proceso de producción de lombrices. Sin embargo, como la máquina aún está todavía en desarrollo tiene una limitación: sólo puede moverse en línea recta.

Teniendo en cuenta esta restricción, se decidió probar la máquina a fin de obtener el mayor número posible de lombrices en una sola pasada (recta, de lado a lado, dentro de las celdas del campo bajo análisis). Es decir, la máquina deberá recoger todos las lombrices de la "columna" o "fila" que posea la suma de la productividad esperada más alta posible.
Escriba un programa que al darle el mapa de campo de lombrices que describe la productividad estimada en cada una de sus celdas, calcule el número total esperado de lombrices que se recogerá por la máquina durante la prueba (siguiendo el criterio establecido con anterioridad).

Entrada

La primera línea de la entrada contiene dos valores enteros N y M que representan la cantidad de filas (1 ≤ N ≤ 100) y columnas (1 ≤ M ≤ 100), respectivamente, de la granja de células a analizar. A continuación, se ingresan las N filas que contienen M valores enteros que representan las productividades esperadas en cada una de las celdas de la granja. Las entradas deben ser leídas desde el dispositivo de entrada estándar.

Salida

La salida debe estar compuesta de una sola línea que contiene un valor entero que indica el número total previsto de lombrices a ser cosechada por la máquina durante la prueba. Esta salida debe debe ser escrita en el dispositivo de salida estándar.

Ejemplo:

Entrada	**Salida**
3 4 81 28 240 9 40 10 100 240 20 180 110 35	450
4 1 100 110 0 100	310

Problema 8: Prant y su indecisión

Prant es un chico muy indeciso: la toma de decisiones nunca fue su fuerte. Pero Prant tiene un nuevo perro, y debe elegir su nombre. Tiene un nombre en mente, pero al ser un chico muy indeciso, Prant, suele cambiar el nombre del animal haciendo una operación de cambio de letras: él elige dos letras A y B, y todas las letras A se tornarán letras B, y las B se tornarán letras A, también A y B pueden ser la misma letra, porque Prant se pone muy nervioso al elegirlas.

Para tomar una decisión Prant elige aleatoriamente algunas letras y las define como sus letras favoritas (después de todo, Prant es un muy indeciso, ¿qué mejor que elegir esas letras aleatoriamente?) de manera que el mejor nombre para el animal, es el que más letras favoritas tiene. El problema es que Prant se pone muy ansioso al hacer las operaciones, y a veces hace cambios muy ilógicos en el nombre.

Por ejemplo, si las letras favoritas de Prant son {a, e, i, o, u}, el nombre inicial del perro es “abccdab”, el cual tiene dos letras favoritas en él y Prant hace las siguientes operaciones de cambio de letras:

- cambiar(c, e), entonces "abccdab" -> "abeedab", ahora el nombre del perro es "abeedab" con 4 letras favoritas.
- cambiar(b, i), entonces "abeedab" -> "aieedai", ahora el nombre del perro es "aieedai" con 6 letras favoritas.
- cambiar(a, f), entonces "aieedai" -> "fieedfi", ahora el nombre del perro es "fieedfi" con 4 letras favoritas.
- cambiar(d, h), entonces "fieedfi" -> "fieehfi", ahora el nombre del perro es "fieehfi" con 4 letras favoritas.

El mejor nombre generado por Prant fue “aieedai”, porque fue el nombre con la mayor cantidad de letras favoritas.

Su tarea es: dado el nombre inicial del perro, las letras favoritas de Prant y las operaciones que fueron efectuadas, descubra cuál es el mejor nombre que se generó para su mascota. El nombre original puede también ser una respuesta válida.

Entrada

La primera línea consiste de tres enteros: *k, m, n* ($1 \leq k \leq 26$, $1 \leq m, n \leq 100000$). En las siguientes dos líneas habrá dos cadenas de caracteres (strings), compuestas sólo de letras en minúsculas, de tamaño *k* y *m* respectivamente. La primera cadena consta de letras no repetidas, que se corresponden a las letras favoritas de Prant. La siguiente es el nombre original del perro (el primer nombre que pensó Prant).

Las siguientes *n* líneas están compuestas de dos caracteres A y B, que representan la cantidad de letras involucradas en cada operación realizada por Prant

Salida

La salida debe contener dos líneas.
La primera deberá de ser un entero V que presenta el número de letras favoritas en el nombre del perro. El segundo debe contener el nombre (si hay múltiples respuestas, muestre la primera generada por Prant, que contiene V cantidad de letras favoritas).

Ejemplo

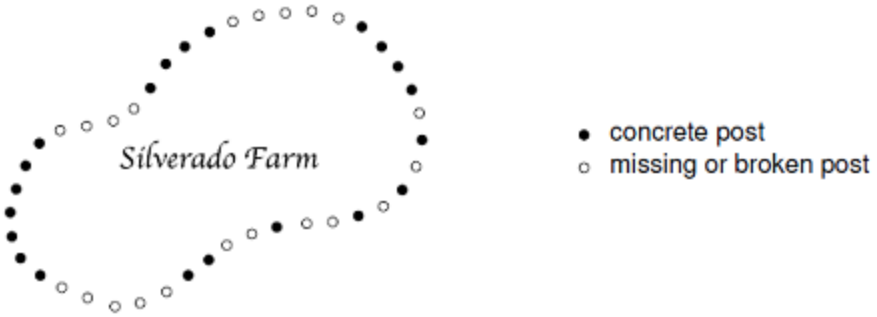
Entrada	Salida
4 5 4 abcd aeiou c e b i a f d h	3 acbou
5 7 4 aeiou abccdab c e b i a f d h	6 aieedai

Problema 9: Tornado

¿Este clima loco es el resultado de la continua interferencia de la humanidad en el medio ambiente o es simplemente un ciclo normal de los cambios climáticos a través del tiempo? Nadie lo sabe a ciencia cierta, pero el hecho es que los fenómenos naturales como los tornados y los huracanes están golpeando nuestro mundo con más fuerza y frecuencia que en décadas pasadas.

Un tornado acaba de llegar a la granja Silverado, productora de leche y ganado bovino, e hizo estragos. El techo del granero se rasgó, varios árboles fueron arrancados, el camión de la granja fue tumbado... Pero lo peor es que el tornado destruyó varias secciones de la valla que rodeaba la propiedad. La valla estaba muy bien construida, con postes de hormigón cada dos metros y alambres de púas que rodeaban todo el perímetro de la granja (el perímetro, en metros, es un número par, por lo que la valla es perfectamente regular).

Ahora, varios postes están rotos o perdidos, y existen algunos blancos en la valla. Para evitar que el ganado se salga de la propiedad, la valla debe ser restaurada lo más rápido posible. La reconstrucción de la valla a su forma original, con postes de hormigón, demandará mucho tiempo. Mientras tanto, los propietarios de la granja decidieron cerrar las brechas con una valla temporal, hecha con postes de madera. Los postes de madera se colocan exactamente en los mismos lugares en los que faltan los postes o están rotos. Pero, con el fin de realizar la reconstrucción temporal más rápida y menos costosa, los propietarios decidieron utilizar el menor número de postes posible: un poste de madera se utiliza para reemplazar un poste de concreto que falta o está roto, solo si la longitud del alambre de púas necesario para unir dos postes (de madera u hormigón) es superior a cuatro metros.



Teniendo en cuenta la descripción de cuales postes faltan o están rotos, se debe codificar un programa que determine el menor número de postes de madera que permitan cerrar todos los huecos de la valla, de acuerdo a la decisión de los propietarios.

Entrada

La entrada contiene varios casos de prueba. La primera línea contiene un entero **N** indicando el número de postes de concretos que había originalmente en una valla ($5 \leq \mathbf{N} \leq 5000$). La segunda línea contiene **N** enteros $\mathbf{X}_i$ indicando el estado de cada poste de concreto luego del tornado ($0 \leq \mathbf{X}_i \leq 1$ para $1 \leq i \leq \mathbf{N}$). Si $\mathbf{X}_i = 1$ el poste *i* está en buenas condiciones, si

X_i = 0 el poste está roto o no está más. Tenga en cuenta que el poste **N** está al lado del poste 1. El final de la entrada se indica con **N** = 0.

Salida

Para cada caso de prueba de la entrada, su programa debe devolver una línea de salida que contiene un número entero que indica la cantidad de postes de madera que son necesarios para restaurar la valla, de acuerdo a la decisión de los propietarios.

Ejemplo

Entrada	Salida
10 1 0 0 1 0 0 1 0 1 1 11 1 0 0 1 0 0 0 1 1 0 1 12 0 0 0 0 0 1 1 0 0 0 1 1 0	2 2 3

Problema 10: Fila del supermercado

Hoy es la inauguración de un gran supermercado en la ciudad, y todos están emocionados con la promesa de precios bajos.

El supermercado tiene **N** cajeros, identificados por números de 1 a **N**, donde cada cajero tiene un tiempo de atención específico $\mathbf{v_i}$ para procesar un producto de cada cliente. Por lo tanto, si un cliente tiene $\mathbf{c_j}$ artículos en su canasto, un cajero tardará $\mathbf{v_i}*\mathbf{c_j}$ segundos para procesar todos los productos del cliente.

Cuando un cliente entra en la fila, espera hasta que un cajero se libere. Si más de un cajero están libres, el cliente será atendido por el cajero con el número de identificación más bajo. Esta caja sólo se liberará cuando el cajero termine de procesar todos los productos de los clientes.

Hay **M** clientes para atender, cada uno con un determinado número de artículos en su canasto. Teniendo en cuenta la información sobre los cajeros y los clientes, el gerente solicita ayuda para saber cuánto tiempo se tardará para que todos los clientes sean atendidos.

Entrada

La primer línea de entrada son dos enteros **N** y **M**, indicando el número de cajeros y clientes, respectivamente. ($1 \leq \mathbf{N} \leq \mathbf{M} \leq 10^4$).

Luego, se ingresarán **N** enteros $\mathbf{v_i}$, indicando cuánto demora el **i**-ésimo cajero para procesar un producto. ($1 \leq \mathbf{v_i} \leq 100$, para cada $1 \leq \mathbf{i} \leq \mathbf{N}$).

Por último, se ingresan **M** enteros $\mathbf{c_j}$, indicando cuántos productos tiene el **j**-ésimo cliente. ($1 \leq \mathbf{c_j} \leq 100$, para cada $1 \leq \mathbf{j} \leq \mathbf{M}$).

Salida

Imprimir una línea que contenga un entero, indicando cuánto se tardará en atender a todos los clientes.

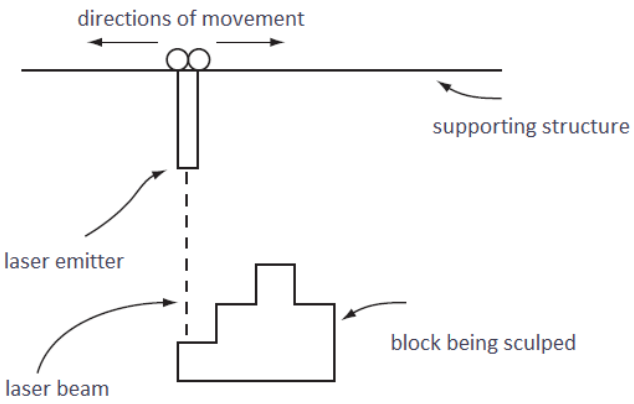
Ejemplo

Entrada	Salida
1 1 3 6	18
1 2 1 5 3	8
2 3 1 2 10 5 3	13

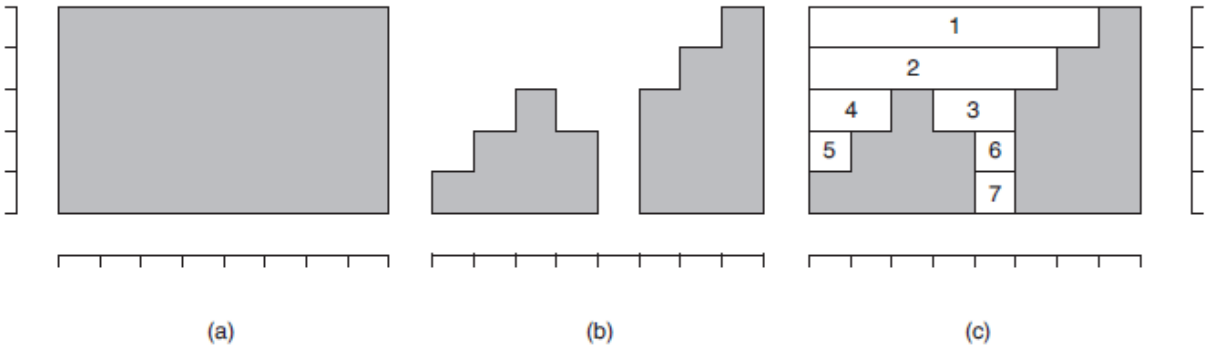
Problema 11: Escultura Laser

Desde su invención en 1958, el láser se ha utilizado en una gran variedad de aplicaciones, como equipos electrónicos, instrumentos quirúrgicos, armas y muchas más.

La imagen de abajo muestra un diagrama de un equipo para esculpir, con un láser, un bloque de un material sólido. En la figura podemos ver un emisor láser que se mueve horizontalmente hacia la derecha y la izquierda con una velocidad constante. Cuando el emisor está encendido y se mueve, una capa de anchura específica se retira del bloque, que se vaporiza por el láser.



La imagen de abajo ilustra el proceso de esculpir con un láser, que muestra un ejemplo de (a) un bloque, con una altura de 5 mm y una longitud de 8 mm, en el inicio del proceso, (b) el formato que queremos para el bloque esculpido, y (c) la secuencia de la absorción de la capa durante el proceso, teniendo en cuenta que en cada paso, se elimina una capa de 1 mm de ancho. En el primer paso, se retira la pieza numerada 1; en el segundo, la pieza 2; y así sigue. Durante el proceso de la escultura, el láser se enciende 7 veces, en total, una vez por cada trozo extraído del bloque.



Entrada

La entrada contiene varios casos de prueba. Cada caso de prueba se compone de dos líneas. La primera, contiene dos números enteros A y C, separados por un espacio en blanco, lo que indica, respectivamente, la altura ($1 \leq A \leq 10^4$) y la longitud ($1 \leq C \leq 10^4$) del bloque esculpido, en milímetros. La segunda línea contiene C enteros X_i, cada uno indica la altura final, en milímetros, del bloque comprendido entre las posiciones i e i+1 a través de la longitud ($0 \leq X_i \leq A$, para $0 \leq i \leq C - 1$). Tenga en cuenta que en cada paso, una capa de anchura de 1 mm se elimina en las partes del bloque en las que el láser está encendido.

El final de la entrada se indica por una línea que contiene sólo dos ceros, separados por un espacio en blanco.

Salida

Para cada caso de prueba, el programa debe imprimir una sola línea, que contiene un número entero, que indica el número de veces que el láser tiene que estar encendido para esculpir el bloque en el formato indicado

Ejemplo

Entrada	Salida
5 8 1 2 3 2 0 3 4 5 3 3 1 0 2 4 3 4 4 1 0 0	7 3 3