

## A. 25 DE MAYO

Los Gonzales es una numerosa familia Argentina que les encanta celebrar todas las fechas Patrias con una gran comida familiar. Particularmente el 25 de Mayo a la abuela teresa le encanta preparar un gran Guiso para toda la familia, eso la pone muy feliz ya que piensa que tener la panza llena le hace tener el corazón contento.

Teresa es muy precavida por lo que realiza las compras para el gran festejo con muchos días de antelación, pero aun así nunca está segura si tiene todo. Vos que sos su nieto informático favorito te pidió encarecidamente que dada la receta original determines si están todos los ingredientes y si hay suficiente cantidad.

Nota: No prestar atención a las unidades de las cantidades. Se asegura que los nombres no contendrán espacio ni caracteres especiales.

### Entrada

Un entero **N** ( $1 \leq N \leq 25$ ) indicando la cantidad de ingredientes de la receta de Teresa, seguido de **N<sub>i</sub>** nombres de ingredientes y su respectiva cantidad necesaria por persona.

Un entero **M** indicando la cantidad de ingredientes que compró Teresa.

Un entero **K** indicando la cantidad de personas invitadas al Guiso (incluida Teresa).  
Seguido de **M<sub>i</sub>** nombres de ingredientes y la cantidad disponible expresada en punto flotante.

### Salida

Determinar si es posible realizar el guiso para las **K** personas imprimiendo "Un guiso muy rico!". En caso de que no, determinar qué ingredientes y en qué cantidad son necesarios aún.

Nota: Para poder expresar precisión decimal 1, en C++ se debe escribir el resultado como, `cout<<fixed<<setprecision(1)<<X`;

Para python `"print{:.1f}".format(X)"`

#Caso 1:

Entrada	Salida
5 Cebolla 0.5 Fideos 0.3 Lentejas 0.5 Papa 0.3 Tomate 0.5 5 4 Cebolla 3 Fideos 1.2 Lentejas 3	Un guiso muy rico!

Liga de programación competitiva



Papa 1.2 Tomate 2	
----------------------	--

#Caso 2:

Entrada	Salida
5 Batata 0.4 Cebolla 0.3 Oregano 0.01 Porotos 0.5 Tomate 0.3 5 4 Batata 1.2 Cebolla 1.2 Oregano 0.04 Porotos 1.5 Tomate 1.2	Batata 0.4 Porotos 0.5

#Caso 3

Entrada	Salida
3 Cebolla 0.5 Fideos 0.5 Tomate 0.5 2 2 Cebolla 1 Tomate 1	Fideos 1.0

## B. A RITMO!

Emi y Jon son dos grandes personas que se conocieron alrededor de un fuego practicando un arte tan ancestral como la vida misma, ellos tocan la guitarra. Descubrieron que hacen un muy buen dúo pero Jon quiere copiarle a Emi ya que lo considera un maestro.

Emi al ser una persona con tanta paciencia y dedicación le enseña a Jon sus mejores secuencias de Acordes que conoce.

Jon intenta darlo todo y se da cuenta de sus diferencias de destreza y conocimiento pero aun así lo intenta.

Jon intenta estar Ritmo de Emi. Se considera que Jon está a "¡A Ritmo!" si su porcentaje de coincidencia en las Secuencias de Emi superan o igualan el 90%. De otra forma "¡Debe Mejorar!"

### Entrada

Un entero **N** ( $1 \leq N \leq 100$ ) indicando el tamaño de la secuencia musical. Seguido de una cadena **S1** indicando las **N<sub>i</sub>** notas de Emi. Una cadena **S2** (**S1**, **S2**  $1 \leq S1, S2 \leq 100$ ) indicando las **N<sub>j</sub>** notas de Jon.

### Salida

Devolver un punto flotante **X** con precisión 2 indicando el porcentaje de aprobación de la secuencia, seguido de la leyenda "A Ritmo!" o "Debe Mejorar!".

#Caso 1:

Entrada	Salida
7 va mo lo re do la re va mo la ve la do re	42% Debe mejorar!

#Caso 2:

Entrada	Salida
10 do re mi fa so do re re mi mi re re mi fa so do re re mi mi	90% A ritmo!

## C. OIRAM

Oiram es un queridísimo profesor en la FICH (Facultad de Inventores de Chistes Horribles), cuándo es el turno de exámenes finales el desconfía mucho de sus alumnos, ya que considera que todos tienen la intención de copiarse. Por lo que ordena la disposición de bancos de forma particular.

Para la materia “Derivando hasta la locura 1” ordena que todos los estudiantes se ubiquen en filas con numeración impar. Y para la materia “Integrando sin descanso 2”, todos los estudiantes deben ubicarse en filas con numeración par. Usted para ayudar a Oiram y que su humor se mantenga estable, debe gestionar la disposición de alumnos. Deberá obtener en cuántos posibles lugares puede sentarse un alumno dependiendo de la materia a rendir.

El aula dispondrá de **N** filas las cuáles tendrán una longitud de **L** cada una.

### Entrada

Un carácter **C** con valor **D** o **I** el cuál indica la materia “Derivando hasta la locura 1” o “Integrando sin descanso 2” respectivamente. Seguido por un entero **N** ( $1 \leq N \leq 100$ ) indicando la cantidad de filas y un entero **L** ( $1 \leq L \leq 100$ ) con la longitud de cada fila.

### Salida

Un entero **X** indicando la cantidad de asientos disponibles para la materia dada.

#Caso 1:

Entrada	Salida
D 5 4	12

## D. BUILDING WALLS

Luego de que el titán colosal destruya la muralla de María, la tropa de exploración decidió construir una nueva muralla, tan resistente que ningún titán podrá destruirla.

Pero si el titán es muy alto, puede simplemente saltar sobre la muralla. Por esto, el ejército de exploración te contrató para que escribas un programa que, dada la altura de la muralla y las alturas de los titanes, calcule qué titanes podrán pasar por encima de la muralla.

Un titán puede saltar sobre una muralla solo si es más alto que la muralla.

Se asegura que el nombre del titán no contendrá espacios.

### Entrada

La primera línea contiene dos enteros **N** ( $1 \leq N \leq 100$ ) y **W** ( $1 \leq W \leq 1000$ ) representando respectivamente cuántos titanes conoce la Tropa de Exploración y la altura de la muralla que desean construir.

Cada una de las siguientes **N** líneas contiene un string **S** ( $1 \leq S \leq 100$ ) representando el nombre del titán, seguido de un entero **H** ( $1 \leq H \leq 1000$ ) que representa la altura del titán.

### Salida

Imprima por pantalla los nombres de los titanes que podrán pasar por arriba de la muralla, en el orden en el que aparecen en la entrada.

#Caso 1:

Entrada	Salida
3 50 colossal 60 acorazado 15 femea 14	colossal

## E. BUCEO PELIGROSO

El reciente terremoto en Nlogonia no afectó mucho las construcciones de la capital, la cual fue el epicentro del temblor. Pero los científicos encontraron que afectó de forma grave la pared del dique, la cuál ahora tiene un daño estructural significativo en su base, si no es reparado rápidamente puede causar el colapso del dique y en consecuencia la inundación de la ciudad.

La reparación debe ser hecha por buceadores, a una gran profundidad, bajo extremas y peligrosas condiciones. Pero como la supervivencia de la ciudad está en juego, hubo una gran cantidad de habitantes dispuestos a cumplir con esta misión.

Como es tradición en misiones peligrosas, cada buceador recibirá al principio de su misión una carta con un número de identificación. Al final de la misión los voluntarios devolverán la identificación, poniéndola en un depósito.

El dique está a salvo de nuevo, pero desafortunadamente parece ser que no todos los voluntarios volvieron de la misión. Fuiste encargado con la rigurosa tarea de, dadas las identificaciones colocadas en el depósito, determinar cuáles voluntarios perdieron sus vidas para salvar la ciudad.

### Entrada

Cada caso de prueba constará de dos líneas. La primer línea contiene dos enteros **N** y **R** ( $1 \leq R \leq N \leq 10^4$ ) indicando respectivamente el número de voluntarios que fueron a la misión y el número de voluntarios que volvieron de la misión (Se asegura que al menos volvió un voluntario). Seguido por **R** enteros indicando las identificaciones de los voluntarios que volvieron.

### Salida

Para cada caso de prueba tu programa debe imprimir una línea conteniendo el número de identificaciones de los voluntarios que no volvieron de la misión, en orden ascendente de sus identificaciones. En caso de que todos hayan vuelto, imprimir “\*”.

#Caso 1:

Entrada	Salida
5 3 3 1 5	2 4

#Caso 2:

Entrada	Salida
6 6 6 1 3 2 5 4	*

## F. PRIMOS GEMELOS

Un **número primo** es aquel que únicamente tiene dos divisores positivos: el 1 y él mismo.

Se llaman **primos gemelos** a los pares de números primos cuya diferencia es exactamente 2.  
Por ejemplo, 41 y 43 forman un par de primos gemelos, ya que ambos son primos y cumplen que  $43 - 41 = 2$ .

Dado un entero **N** se necesita saber los dos primos gemelos más grandes menores a **N**.

### Entrada

Un entero **N** ( $6 \leq N \leq 1000$ )

### Salida

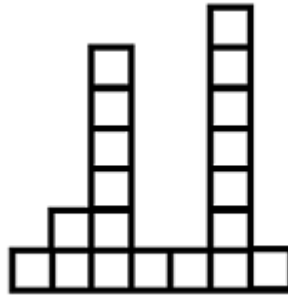
Dos enteros **X** e **Y** indicando los primos gemelos según las condiciones anteriores.

#Caso 1:

Entrada	Salida
10	5 7

## G. VALLAS

Hay una reja frente a la casa de Raúl. La cerca consta de  $n$  tablones del mismo ancho, uno tras otro, de izquierda a derecha. La altura del  $i$ -ésimo tablón es de  $h_i$  metros; cada tablón puede tener alturas distintas.



Valla para  $n = 7$  y  $h = [1, 2, 6, 1, 1, 7, 1]$

Raúl compró un piano elegante y está pensando en cómo meterlo en casa. Para llevar a cabo su plan, necesita sacar exactamente  $k$  tablones consecutivos de la cerca. Cuánto más altos son los tablones, más difícil es arrancarlos, así que Raúl quiere encontrar tales  $k$  tablones consecutivos cuya suma de sus alturas sea la mínima posible.

Te pidió ayuda para escribir un programa que encuentre los índices de los  $k$  tablones consecutivos con la altura total mínima. La cerca no es cíclica.

### Entrada

La primera línea de la entrada contiene los enteros  $N$  y  $K$  ( $1 \leq N \leq 1.5 \cdot 10^5$ ,  $1 \leq K \leq N$ ) indicando el número de tablones de la valla y el ancho del hueco para el piano. La segunda línea contiene la secuencia de enteros,  $h_1, h_2, \dots, h_n$  ( $1 \leq h_i \leq 100$ ) donde  $h_i$  es la altura del  $i$ -ésimo tablón en la valla.

### Salida

Imprima un entero  $X$  tal que la suma de las alturas de los tablones  $i, i+1, \dots, i+k-1$  sea la mínima posible. Si hay varios  $X$ , imprima cualquiera de ellos.

#Caso 1:

Entrada	Salida
7 3 1 2 6 1 1 7 1	3



# H. SLACKLINE!

El **slackline** es un deporte en el que se coloca una **cuerda** tensa entre dos árboles con el objetivo de caminar sobre ella sin caerse.

Carlos es un fanático de este deporte, y además es muy perfeccionista: cada día decide la longitud exacta que quiere para su slackline. Si en el parque cercano a su casa existen dos árboles cuya distancia coincide exactamente con su deseo, utilizará esos. En caso de que no exista una distancia exactamente igual a la que desea, Carlos se conformará con la **distancia menor más grande posible** (es decir, una distancia más pequeña tal que sea la más grande posible).

El parque puede representarse como una cuadrícula de **MxN** posiciones. Cada posición puede contener o no un árbol. La distancia entre posiciones adyacentes (verticales u horizontales) es **D**. La cuerda de Carlos se puede colocar únicamente entre **dos árboles distintos**, sin atravesar otro árbol por el medio.

## Entrada

Un entero **L** ( $1 \leq L \leq 100$ ) indicando la longitud deseada del Slackline.

Un entero **D** ( $1 \leq D \leq 100$ ) indicando la distancia entre posiciones adyacentes de la cuadrícula.

Dos enteros **M** y **N** ( $1 \leq M, N \leq 100$ ) indicando las dimensiones de la cuadrícula.

Una matriz **M x N** de enteros, donde; un 1 representa un árbol y 0 indica que no hay árbol.

## Salida

Un único entero indicando la longitud del slackline que Carlos utilizará, según su deseo y las condiciones dadas anteriormente

#Caso 1:

Entrada	Salida
13 3 3 3 1 0 0 0 0 0 0 0 1	12

#Caso 2:

Entrada	Salida
20 5 3 3 1 0 0 0 0 0 0 1 1	18

#Caso 3:

Entrada	Salida
21 5 3 3 1 0 1 0 1 0 0 0 1	15

Nota: Considerar que la distancia entre posiciones es D arrancando desde D. Por ejemplo. Si  $D=3$  y se tiene 1 0 1 entonces la distancia entre árboles será de 9, no de 6.

# I. CIFRA DE CÉSAR

Julio César usaba un sistema de criptografía, ahora conocido como Cifrado César (Caesar Cipher), que desplazaba cada letra 2 lugares más a lo largo del alfabeto (por ejemplo, 'A' cambia a 'C', 'R' cambia a 'T', etc). Al final del alfabeto reiteramos desde el principio, es decir, 'Y' cambia a 'A'. Podemos, por supuesto, tratar de desplazar por cualquier número.

Nota: Considerar que 'A' es el número 65 en código ASCII. Se asegura que la entrada no tendrá espacios ni caracteres especiales.

## Entrada

Cada caso de prueba está compuesto por dos líneas. La primera contiene una cadena de caracteres que se encuentra codificada. Esta cadena contendrá entre 1 y 50 caracteres, inclusive. Cada carácter es una letra mayúscula ('A' - 'Z'), que pertenece a la codificación de la cadena del Cifrado César. La segunda línea contiene el número de desplazamiento a la derecha, este valor está entre 0 y 25, inclusive.

## Salida

Dado un texto codificado y un número de letras desplazadas, decodificar a la oración original, de acuerdo con la explicación anterior y el siguiente ejemplo.

#Caso 1:

Entrada	Salida
VQREQFGT 2	TOPCODER

#Caso 2:

Entrada	Salida
JMGL 4	FICH

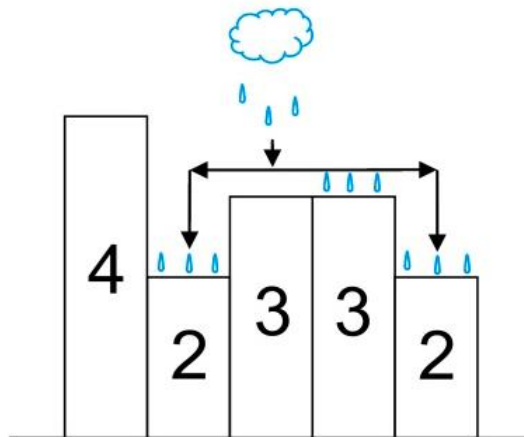
#Caso 3:

Entrada	Salida
WOCCS 10	MESSI

## J. MANOLO Y EL CAMPO

El pequeño Manolo a veces viaja a lo de su abuela que vive en el campo. Su abuela tiene un gran jardín, el cuál puede representarse como un rectángulo de tamaño de  $1 \times N$ , cuando es visto desde arriba. Este rectángulo está dividido en  $N$  secciones cuadradas iguales. El jardín es muy peculiar, ya que cada sección cuadrada tiene una altura fija y, gracias al nuevo sistema de riego, podemos crear lluvia artificial sobre cada sección.

Crear lluvia artificial es una operación bastante costosa. Por eso nos limitamos a crearla solo sobre una sección. En ese caso, el agua de cada sección regada fluirá a las secciones vecinas si su altura no supera la de la sección. Es decir, por ejemplo, el jardín se puede representar con un rectángulo de  $1 \times 5$ , donde las alturas de las secciones son iguales a 4, 2, 3, 3, 2. Entonces, si creamos una lluvia artificial sobre cualquiera de las secciones con altura de 3, el agua fluirá sobre todas las secciones, excepto la que tiene la altura de 4. El ejemplo está dado en la imagen:



Como a Manolo le apasiona la programación, decidió buscar una sección donde, si creamos lluvia artificial sobre ella, se maximizará el número de secciones regadas.

### Entrada

La primera línea contiene un entero positivo  $N$  ( $1 \leq N \leq 1000$ ). La segunda línea contiene  $N$  enteros positivos los cuales representan la altura de las secciones.

### Salida

Imprime un entero, indicando el número máximo de secciones regadas si creamos una lluvia artificial sobre exactamente una sección.

#Caso 1:

Entrada	Salida
1	1

Liga de programación competitiva

**MNR**

2	
---	--

#Caso 2:

Entrada	Salida
5 1 2 1 2 1	3

#Caso 3:

Entrada	Salida
8 1 2 1 1 1 3 3 4	6