

## A. 25 DE MAYO

Los Gonzales son 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 aún 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.

### Entrada

Un entero **N** indicando la cantidad de ingredientes por porción de la receta de Teresa por persona, seguido de **N<sub>i</sub>** nombres de ingredientes y su respectiva cantidad. Un entero **M** indicando la cantidad de ingredientes que compró Teresa y **K** indicando la cantidad de personas invitadas al Guiso (incluida Teresa). Seguido de **M<sub>i</sub>** nombres de ingredientes y la cantidad disponible.

### Salida

Determinar si es posible realizar el guiso para las **K** personas escribiendo "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 Papa 1.2 Tomate 2	Un guiso muy rico!

#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. 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** indicando la cantidad de filas y un entero **L** con la longitud de cada fila.

## Salida

La cantidad de asientos disponibles para la materia dada.

#Caso 1:

Entrada	Salida
D 5 4	12

## C. PALINDROMO

Un palíndromo (del griego πάλιν δρόμος, palin dromos, 'volver a ir atrás') es una palabra o frase que se lee igual en un sentido que en el otro (por ejemplo; Ana, Anna, Otto).

[Wikipedia]

El juego favorito de Lucas mientras cocina (ya que le aburre mucho cocinar), es pensar en palabras o combinaciones de letras las cuáles podrían ser palíndromas.

Ya que Lucas piensa en una gran cantidad de palabras a diario él tiene la costumbre de anotarlas en una agenda para luego analizar si éstas pueden cumplir o no la condición de ser palíndromas. Y en caso de serlas le gusta contar la cantidad de **permutaciones** posibles las cuáles las hagan palíndromas.

### Entrada

Una cadena de caracteres de longitud **N**. Se asegura que solo estará compuestas de letras del alfabeto inglés y no tendrá espacios ni símbolos de caracteres especiales.

### Salida

El número de total de combinaciones que hagan palíndroma la palabra o "No es posible!" en caso de que no se pueda formar ninguna combinación.

#Caso 1:

Entrada	Salida
ana	1

#Caso 2:

Entrada	Salida
ajo	No es posible!

#Caso 3:

Entrada	Salida
abcabc	6

Nota: Las combinaciones posibles que se hacen palíndromas son: [abccba, acbbca, baccab, bcaacb, cabbac, cbaabc]

#Caso 4:

Entrada	Salida
aabb	2

Nota: Las combinaciones posibles que se hacen palíndromas son; [abba, baab]

#Caso 5:

Entrada	Salida
abcbcaaa	12

## D. THE RANK

John Smith sabe que su hijo, Thomas Smith, está entre los mejores estudiantes de la clase e incluso de la escuela. Después de que los estudiantes hayan hecho los exámenes de Inglés, Literatura, Matemática e Historia los resultados fueron publicados.

Hay **N** estudiantes, donde cada uno tiene un **único** id (de 1 a N). El id de Thomas es el **1**. Cada estudiante tiene cuatro puntajes correspondiendo a los exámenes de Inglés, Literatura, Matemática e Historia respectivamente.

Los estudiantes son dados en orden creciente según sus ids.

Luego, en el ranking, los estudiantes serán ordenados de forma decreciente por la suma de sus puntajes. O sea, un estudiante con la mayor suma tendrá el primer lugar en el ranking. Si dos o más estudiantes tienen la misma suma de puntajes, entonces serán ordenados de forma creciente según sus ids.

Ayuda a John a encontrar a su hijo.

### Entrada

La primera línea contiene un entero **N** ( $1 \leq N \leq 1000$ ) indicando la cantidad de estudiantes.

Cada una de las siguientes **N** líneas contendrán 4 enteros, **a<sub>i</sub>**, **b<sub>i</sub>**, **c<sub>i</sub>**, **d<sub>i</sub>** ( $0 \leq a_i, b_i, c_i, d_i \leq 100$ ) correspondiendo a los grados del i-esimo estudiante en Inglés, Literatura, Matemática e Historia.

### Salida

Imprimir la posición de Thomas Smith. Una vez que ya se haya evaluado el ranking. El id de Thomas es 1.

#Caso 1:

Entrada	Salida
5 100 98 100 100 100 100 100 100 100 100 99 99 90 99 90 100 100 98 60 99	2

Nota: Los estudiantes tuvieron un puntaje total de, 398, 400, 398, 379 y 357. Entre los 5 estudiantes, Thomas y el tercer estudiante tuvieron el puntaje más alto, pero como Thomas tiene un id más chico entonces su posición es 2.

#Caso 2:

Entrada	Salida
6	1

Liga de programación competitiva



100 80 90 99	
60 60 60 60	
90 60 100 60	
60 100 60 80	
100 100 0 100	
0 0 0 0	

## E. FRONTERAS

Un aficionado a la geografía desea tiene una lista de países y sus ciudades. A su vez, tiene otra lista que indica, para cada ciudad, sus ciudades vecinas. Este entusiasta te pide que le hagas un programa que le diga, dado un país, qué países son limítrofes.

### Entrada

Un entero **N** ( $1 \leq N \leq 100$ ) indicando el número de países, seguido de, **N** entradas **C P** dónde **C** corresponde a una ciudad y **P** a qué país pertenece esa ciudad **C**.

Un entero **M** indicando el número de vecindades seguido de **M** entradas **A B** que son los pares de ciudades vecinas.

Una cadena de caracteres **X** indicando el país que está siendo buscado. La entrada asegura que los nombres de ciudades no tendrán espacios en blanco.

### Salida

Nombres de los países vecinos a **X**, separados por una línea en blanco y ordenados en forma alfabética ascendente. En caso de no tener, imprimir "NO".

#Caso 1:

Entrada	Salida
6 BuenosAires Argentina Cordoba Argentina Santiago Chile Valparaiso Chile Asuncion Paraguay CiudadDelEste Paraguay	Chile Paraguay
5 BuenosAires Santiago Cordoba Santiago Cordoba Asuncion Asuncion CiudadDelEste Valparaiso Santiago Argentina	

## F. MANU AND CAFES

A manu le gusta muchísimo ir a cafeterías. Durante su vida el visitó cafeterías  $N$  veces. Desafortunadamente, Manu empezó a sentir que cada visita no era muy distinta una a la otra. Para arreglar eso Manu realizó una pequeña investigación.

Manu le asignó un id individual a todas las cafeterías. Después las escribió en el orden en que las fue visitando. Ahora Manu quiere encontrar cuál fue la cafetería que visitó menos veces para poder asistir a esa nuevamente. Cada cafetería se identificará mediante un id que va desde  $0$  hasta  $M-1$ . Ayuda a Manu a encontrar esa cafetería.

### Entrada

Un entero  $M$  ( $1 \leq M \leq 100$ ) indicando la cantidad de cafeterías distintas que hay. Un entero  $N$  ( $1 \leq N \leq 100$ ) indicando la cantidad de visitas a cafeterías. Seguido de un arreglo de tamaño  $N$  donde  $a_1, a_2, \dots, a_n$  indica cada visita a cada cafetería.

### Salida

Imprimir la cafetería menos visitada por Manu.

#Caso 1:

Entrada	Salida
3 5 0 2 1 0 1	2

#Caso 2:

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



## G. HACIENDO PUZZLES

B	O	C	A
A	C	B	O
C	X	O	B
O	B	A	C

Javiera y Federico invierten mucho tiempo en línea completando diferentes puzzles.

En este caso, están frente a un puzzle en el que deben completar un tablero de  $N \times N$  casillas utilizando  $N$  letras diferentes, de tal forma que cada casilla tenga escrita exactamente una letra, y cumpliendo que no haya letras repetidas en una misma fila ni en una misma columna.

Luego de terminar el puzzle, enviaron su solución y el juez de corrección automática les respondió que está mal, pero además les dijo que el error está exactamente en una casilla, que si logran corregirla entonces tendrían una solución correcta.

Para ayudarlos, deberás indicar la fila y la columna correspondiente a la casilla errónea y decir qué letra debería ir en esa casilla para obtener una solución correcta.

### Entrada

La entrada contiene múltiples casos de prueba.

La primera línea de cada caso de prueba contiene un entero  $N$  ( $3 \leq N \leq 26$ ), indicando que el tablero posee  $N \times N$  casillas. Luego siguen  $N$  líneas con  $N$  letras mayúsculas del alfabeto inglés describiendo el tablero con la solución errónea de Javiera y Federico.

### Salida

## Liga de programación competitiva



Por cada caso de prueba imprimir dos entero **F**, **C**, y una letra **X**, donde **F** y **C** corresponden a la fila y columna de la casillas donde se cometió el error, y **X** corresponde a la letra que deberían utilizar en dicha casilla para obtener una solución correcta. Se garantiza que dicha casilla existe y es única.

#Caso 1:

Entrada	Salida
6 OEYCDK EYOKCD KDCEOY CKHOYE YOEDKC DCKYEO	4 3 D

#Caso 2:

Entrada	Salida
3 IWL GIW WLI	2 1 L

## H. ROUTER

Marcos es anfitrión de una competencia de programación y tiene un servidor donde se evalúan los ejercicios que suben los participantes. El router de Marcos es bastante malo, por lo que tiene que saber bien dónde posicionarlo, y el día de la competencia rápidamente debe moverlo al lugar más cercano a todos los participantes, para que la señal llegue a todos.

Como él no controla dónde se van a sentar los equipos, y a éstos poco les importan los dolores de cabeza que sufre la comisión organizadora, te pide que hagas un programa donde pueda cargar la posición de cada equipo y reporte en el momento la mejor posición para el router y el radio de alcance mínimo que debe tener para llegar a todos los equipos.

### Entrada

Un entero **N** ( $1 \leq N \leq 100$ ) indicando el número de equipos. **N** entradas con las posiciones **X** e **Y** de cada equipo.

### Salida

Tres reales **X**, **Y** y **R** indicando las coordenadas del router y el alcance de la red.

Nota: Para poder expresar precisión decimal 2, en C++ se debe escribir el resultado como, `cout<<fixed<<setprecision(2)<<X`;

Para python `"print{:.2f}".format(X)"`

#Caso 1:

Entrada	Salida
7	3.5
4 2	2
6 1	2.69
5 1	
1 3	
6 3	
4 3	
2 1	

# I. INFLACIÓN

En un colegio se paga una cuota mensual para cubrir los costos de funcionamiento.

Debido a la inflación, resulta difícil predecir cómo evolucionarán dichos costos con el tiempo.

Se dispone del historial de los últimos **N** meses de actividad, donde cada mes está representado por un valor, este puede ser negativo debido a las condiciones económicas desfavorables, o positivas en caso contrario.

Para tener una idea sobre el histórico de la economía, El contador del colegio necesita analizar este historial para obtener:

1. El período consecutivo de **M meses** con la mayor rentabilidad total.
2. El mes más rentable dentro de ese período.
3. La rentabilidad total obtenida en dicho intervalo.

## Entrada

Un entero **N** ( $1 \leq N \leq 100000$ ) indicando la cantidad de meses a evaluar. Un entero **M** ( $1 \leq M \leq N$ ) indicando el tamaño de los períodos a analizar. A continuación siguen **N** elementos  $n_i$  ( $-32768 \leq n_i \leq 32767$ ) que corresponden al balance de cada mes.

## Salida

En la primera línea se pide el rango del período. Seguido por la rentabilidad del período, y por último el mes más rentable dentro de ese período. Se pide cada resultado en una línea distinta.

#Caso 1:

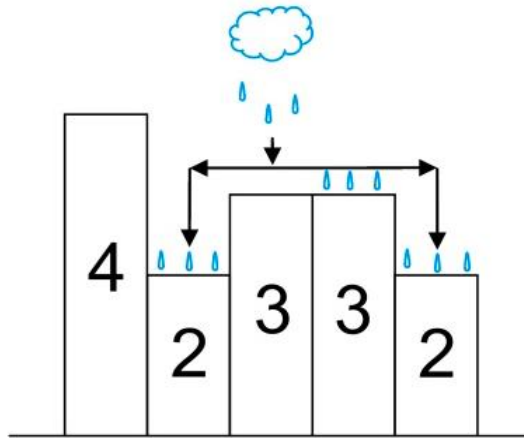
Entrada	Salida
10 3	5 7
11 -3 -5 2 20 -10 30 4 -100 22	40
	3

# 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 x 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
2	

#Caso 2:

Entrada	Salida
5	3
1 2 1 2 1	

#Caso 3:

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