

# Torneo Rosarino de Programación 2026 - Inicial

## *Cuaderno de Problemas*

6 de Junio de 2026

### **Información General**

Este cuaderno contiene 10 problemas. Verifique que su cuaderno está completo.

#### **A) Sobre la entrada**

1. La entrada de su programa debe ser leída de entrada standard.
2. La entrada está compuesta de un único caso de prueba, descrito mediante una o más líneas dependiendo del problema.
3. Cuando una línea de entrada contiene varios valores, estos están separados por un único espacio en blanco; la entrada no contiene ningún otro espacio en blanco.
4. Cada línea, incluyendo la última, contiene exactamente un caracter de final-de-línea.
5. El final de la entrada coincide con el final del archivo.

#### **B) Sobre la salida**

1. La salida de su programa debe ser escrita en salida standard.
2. Cuando una línea de salida contiene varios valores, estos deben ser separados por un único espacio en blanco; la salida no debe contener ningún otro espacio en blanco.
3. Cada línea, incluyendo la última, debe contener exactamente un caracter de final-de-línea.

## Problema A - Obra Pública - Fácil

En el país de Grafonia hay  $N$  ciudades numeradas del 1 al  $N$  y  $M$  caminos bidireccionales que las conectan. Cada camino tiene un estado de conservación:

- 0 → en buen estado, se puede usar libremente.
- 1 → en mal estado, hay que repararlo antes de poder usarlo.



Vos sos el ministro de obras públicas de Grafonia y querés viajar desde la ciudad 1 hasta la ciudad  $N$ . Tu objetivo es determinar la mínima cantidad de caminos en mal estado que se deben reparar para que el viaje sea posible, o informar si no existe ningún recorrido posible incluso reparando todos los caminos.

### Entrada

La primera línea contiene dos enteros  $N$  y  $M$  ( $2 \leq N \leq 10^5$ ,  $0 \leq M \leq 2 \cdot 10^5$ ), el número de ciudades y el número de caminos, respectivamente.

Las siguientes  $M$  líneas contienen tres enteros  $u$ ,  $v$  y  $e$  ( $1 \leq u, v \leq N$ ,  $e \in \{0, 1\}$ ), indicando que existe un camino bidireccional entre la ciudad  $u$  y la ciudad  $v$  con estado  $e$ .

### Salida

Imprimir un único entero que indique la mínima cantidad de caminos en mal estado que hay que reparar para ir de la ciudad 1 a la ciudad  $N$ , o imprimir  $-1$  si no hay manera de llegar.

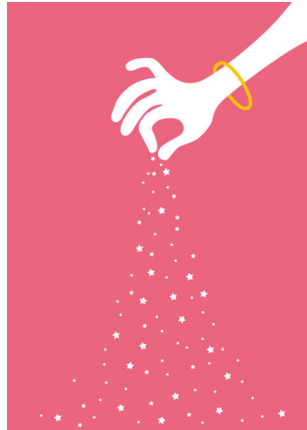
Ejemplo de entrada 1	Ejemplo de salida 1
5 7	0
1 2 0	
2 3 0	
1 3 1	
1 4 1	
4 3 0	
4 5 1	
3 5 0	

Ejemplo de entrada 2	Ejemplo de salida 2
5 7 1 2 0 2 3 0 1 3 1 1 4 1 4 3 0 4 5 1 3 5 1	1

## Problema B - Cocina con magia

Paula Paulina es una renombrada cocinera rosarina que alcanzó la fama gracias a sus recetas típicas y simples.

Sin embargo, últimamente empezó a atravesar una crisis: es tan famosa que sus ingredientes parecen no ser suficientes para toda la demanda que ella tiene. Paula acude a nosotros como programadores y nos dice algo entre dientes: “*Mirá, me quedé sin ingredientes y tuve que acudir a los poderes arcanos*”. Así es, Paula Paulina ya no cocina solo con ingredientes, sino también con magia.



Como quiere cocinar Carlitos con una cantidad absurda de ingredientes ( $10^5$ ), nos ha enunciado los siguientes puntos clave:

- Para hacer sus Carlitos necesita  $N$  ingredientes distintos.
- De cada ingrediente  $i$  necesita una cantidad  $A_i$  de gramos.
- Para cada ingrediente  $i$  Paulina tiene  $B_i$  gramos en su cocina.
- Además, tiene  $K$  gramos de un ingrediente mágico que puede convertir en cualquier otro ingrediente.

Para hacerlo más claro, veamos la siguiente entrada:

$i$	1	2
$A_i$	8	9
$B_i$	5	6

$$K = 6$$

En este caso, vemos que Paulina solo puede hacer un Carlito debido a que cuenta con 5 gramos del ingrediente  $B_1$ , pudiendo complementarlo con 3 gramos del ingrediente mágico para llegar a 8 gramos, y luego, cuenta con 6 gramos del ingrediente  $B_2$  pudiendo complementarlo con otros 3 gramos del ingrediente mágico, para llegar a 9 gramos.

Dado un  $N$ , un  $K$ , un arreglo  $A_1, A_2, \dots, A_N$  y otro arreglo  $B_1, B_2, \dots, B_N$ , determine a cuántas personas puede saciar Paulina, si cada una se conforma con un Carlito.

## Entrada

La primera línea contiene dos enteros  $N$  y  $K$  ( $1 \leq N \leq 10^5$ ,  $1 \leq K \leq 10^9$ ), la cantidad de ingredientes distintos a usar y la cantidad de gramos del ingrediente mágico, respectivamente.

La segunda línea contiene  $N$  enteros  $A_1, A_2, \dots, A_N$  ( $1 \leq A_i \leq 10^9$  para cada  $i = 1 \dots n$ ), donde el  $i$ -ésimo número equivale a la cantidad de gramos que se necesitan del  $i$ -ésimo ingrediente para hacer un Carlito.

La última línea contiene  $N$  enteros  $B_1, B_2, \dots, B_N$  ( $1 \leq B_i \leq 10^9$  para cada  $i = 1 \dots n$ ), donde el  $i$ -ésimo número equivale a la cantidad de gramos del  $i$ -ésimo ingrediente que hay disponibles.

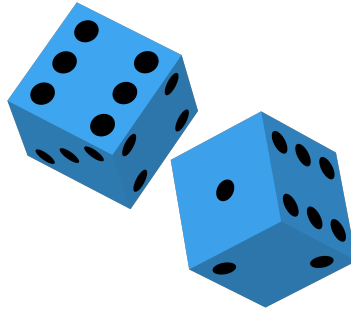
## Salida

Imprima una sola línea con un entero indicando la cantidad máxima de personas a las que Paulina puede satisfacer.

Ejemplo de entrada 1	Ejemplo de salida 1
2 6 8 9 5 6	1

## Problema C - Tirando Dados

Alice y Bob están jugando una partida con dados. Alice tiene  $M$  dados, donde el  $i$ -ésimo dado tiene  $A_i$  caras. Por su parte, Bob tiene  $N$  dados, donde el  $i$ -ésimo dado tiene  $B_i$  caras. Todos los dados son justos y sus caras están numeradas desde el 1 hasta la cantidad de caras correspondientes.



En cada partida, ambos jugadores tiran todos sus dados simultáneamente y suman los resultados obtenidos. El jugador que consiga la mayor suma total se consagra como el ganador.

Sabiendo que Alice tiene una probabilidad  $P_A$  de ganar y Bob tiene una probabilidad  $P_B$  de ganar, tu tarea es determinar cuál de los dos tiene la ventaja estadística. ¿Quién de los dos tiene mayor probabilidad de ganar, o acaso tienen exactamente la misma probabilidad de hacerlo?

### Entrada

La primera línea contiene dos enteros  $M$  y  $N$  ( $1 \leq M, N \leq 10^5$ ), la cantidad de dados que tienen Alice y Bob, respectivamente.

La segunda línea contiene  $M$  enteros  $A_1, A_2, \dots, A_M$  ( $1 \leq A_i \leq 10^9$ ), donde el  $i$ -ésimo número indica la cantidad de caras del  $i$ -ésimo dado de Alice.

La tercera línea contiene  $N$  enteros  $B_1, B_2, \dots, B_N$  ( $1 \leq B_i \leq 10^9$ ), donde el  $i$ -ésimo número indica la cantidad de caras del  $i$ -ésimo dado de Bob.

### Salida

Imprimir una única línea indicando el resultado del juego:

- ALICE si Alice tiene mayor probabilidad de ganar ( $P_A > P_B$ ).
- BOB si Bob tiene mayor probabilidad de ganar ( $P_A < P_B$ ).
- EMPATE si ambos tienen exactamente la misma probabilidad de ganar ( $P_A = P_B$ ).

Ejemplo de entrada 1	Ejemplo de salida 1
8 1 4 4 4 4 4 4 4 4 6	ALICE

Ejemplo de entrada 2	Ejemplo de salida 2
2 2 6 4 4 6	EMPATE

## Problema D - Pablito minimiza clavitos

Todos en Rosario saben cuál es el pasatiempo favorito de Pablito: clavar clavitos. Pero a Pablito no le gusta hacerlo sin criterio alguno. Él es un perfeccionista y, como tal, siempre busca usar la mínima cantidad de clavos posible para completar su labor. Para el desafío de hoy, Pablito decidió arrojar  $N$  tablas de madera de forma aleatoria sobre una misma línea recta con la condición de no poder moverlas en lo absoluto luego de tirarlas.



Luego de eso se dispone a unir las usando sus nuevos Clavos Extra Largos™, los cuales le permiten atravesar y clavar simultáneamente cualquier número de maderas que se superpongan en un mismo punto.

Viendo las tablas desde arriba, cada una puede representarse geoméricamente como un intervalo de la forma  $[L, R]$ . Tu tarea es determinar e imprimir la mínima cantidad de clavos (o puntos) que habría que poner de tal forma que todos los intervalos contengan al menos un clavo.

### Entrada

La primera línea contiene un único entero  $N$  ( $1 \leq N \leq 2 \cdot 10^5$ ) que representa la cantidad de tablas de madera.

Las siguientes  $N$  líneas contienen dos enteros  $L_i$  y  $R_i$  ( $0 \leq L_i \leq R_i \leq 10^9$ ) cada una, que indican los extremos izquierdo y derecho (inicio y fin) de la  $i$ -ésima tabla.

### Salida

Imprima una sola línea con un único entero  $K$ , que indica la mínima cantidad de clavos necesarios para asegurar todas las tablas de madera.

<b>Ejemplo de entrada 1</b>	<b>Ejemplo de salida 1</b>
2 2 3 3 4	1

<b>Ejemplo de entrada 2</b>	<b>Ejemplo de salida 2</b>
5 1 2 3 4 5 6 7 8 9 10	5

## Problema E - Missing Xor

Mariano está aprendiendo una nueva operación binaria: el XOR bit a bit ( $\oplus$ ). Esta operación compara las representaciones binarias de dos valores y devuelve 1 solo en las posiciones donde los bits son distintos, y 0 donde son iguales. En lenguajes como C++ o Python, se realiza usando el operador  $\wedge$ .

Para practicar, Mariano escribió  $N$  enteros  $A_1, A_2, \dots, A_N$  en su cuaderno. Ahora quiere agregar exactamente un número  $K$  a su cuaderno de forma que el XOR de todos los números, incluyendo  $K$ , sea exactamente  $X$ . ¿Existe tal número  $K$ ?

Formalmente, Mariano busca un  $K$  que cumpla:

$$K \oplus A_1 \oplus A_2 \oplus \dots \oplus A_N = X$$

### Entrada

La primera línea contiene un único entero  $N$  ( $1 \leq N \leq 2 \cdot 10^5$ ), la cantidad de enteros que Mariano escribió en su cuaderno.

La segunda línea contiene  $N$  enteros  $A_1, A_2, \dots, A_N$  ( $1 \leq A_i \leq 10^9$ ), los números del cuaderno de Mariano.

La tercera línea contiene un único entero  $X$  ( $1 \leq X \leq 10^9$ ), el valor que Mariano quiere obtener.

### Salida

Si no existe ningún número  $K$  que cumpla la condición, se debe imprimir una única línea con la palabra NO.

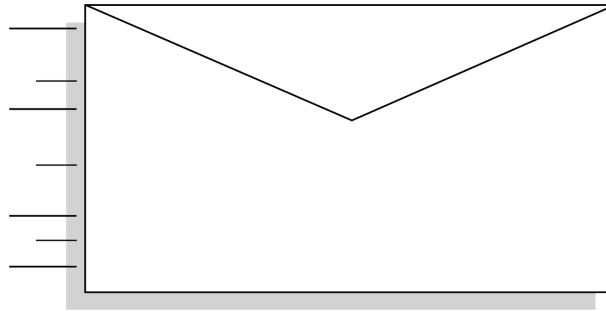
En el caso de que sí exista, se debe imprimir SI en la primera línea y, en la siguiente línea, el valor de dicho  $K$ .

Ejemplo de entrada 1	Ejemplo de salida 1
3 1 2 2 4	SI 5

Ejemplo de entrada 2	Ejemplo de salida 2
2 1 1 100	SI 100

## Problema F - Spam

Un día abriendo su correo electrónico Bob se encuentra con el siguiente mensaje:  
“Hola, el rey de Nlogonia ha muerto y en su testamento te ha dejado como único heredero  
Por favor danos acceso a tu cuenta bancaria para poder depositarte la herencia.”  
Bob, emocionado por recibir el dinero, procede a buscar el papel donde tiene anotada la  
contraseña de su cuenta bancaria. Dicho papel contiene un número  $N$  ( $1 \leq N \leq 100$ ) y una  
serie de operaciones matemáticas, sin embargo, Bob no es muy bueno en las matemáticas  
así que te pide que lo ayudes a hacer un programa para descifrar la contraseña



### Entrada

Una única línea con un entero  $N$  ( $1 \leq N \leq 100$ )

### Salida

Se debe imprimir una única línea con el resultado de las siguientes operaciones “Multiplica  $N$  por 666, divide el resultado por 74, luego suma 67 y finalmente multiplica por 757”.

Ejemplo de entrada 1	Ejemplo de salida 1
5	84784

## Problema G - Conjetura de Silverbach

Muchos conocen la famosa conjetura de Goldbach, la cual establece que “todo número par mayor o igual a 4 se puede expresar como la suma de dos números primos”. Sin embargo, muy pocos conocen la conjetura de su hermano, Silverbach, que propone que “todo número entero mayor o igual a 12 se puede expresar como la suma de dos números compuestos”. Al contrario que la de Goldbach, la conjetura de Silverbach sí se puede demostrar.

Dado un entero  $N$  mayor o igual a 12, exprésalo como la suma de dos números compuestos.

\*Nota: Un número compuesto es un número entero positivo que tiene al menos un divisor distinto de 1 y de sí mismo (es decir, no es primo y es mayor que 1).

### Entrada

La entrada consiste en una única línea que contiene un entero  $N$  ( $12 \leq N \leq 10^6$ ).

### Salida

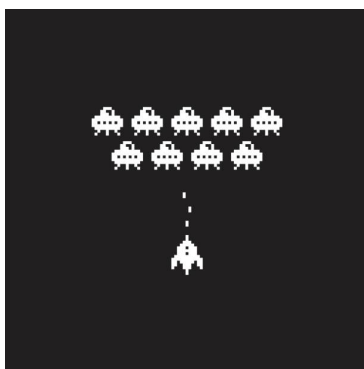
Imprimir en una única línea dos números compuestos  $X$  e  $Y$  ( $1 < X, Y < N$ ) tales que  $X + Y = N$ . Si existen múltiples soluciones válidas, imprimir cualquiera de ellas.

Ejemplo de entrada 1	Ejemplo de salida 1
12	6 6

## Problema H - Space Geometry

Estás programando el motor gráfico para un juego de naves retro estilo *Shoot 'em up* diseñado para consolas de 32 bits.

En este juego, cuando los enemigos son destruidos, dejan una estela de daño expansivo con forma perfectamente circular. Un problema conocido en el desarrollo de videojuegos 2D es calcular exactamente el área de la pantalla que está cubierta por estas explosiones, especialmente cuando dos explosiones ocurren muy cerca y sus áreas se superponen.



Para optimizar el *hitbox* del jugador, el motor del juego te envía las coordenadas en pantalla de dos explosiones simultáneas y sus respectivos radios de daño. Tu objetivo es escribir el algoritmo que calcule el área total cubierta por la unión de ambas explosiones.

### Entrada

La primera línea contiene tres enteros  $X_A$ ,  $Y_A$  y  $R_A$  ( $-100 \leq X_A, Y_A \leq 100$ ,  $1 \leq R_A \leq 100$ ), representando la coordenada y el radio de la primera explosión.

La segunda línea contiene tres enteros  $X_B$ ,  $Y_B$  y  $R_B$  ( $-100 \leq X_B, Y_B \leq 100$ ,  $1 \leq R_B \leq 100$ ), representando la coordenada y el radio de la segunda explosión.

### Salida

Imprimir el área exacta que abarcan ambas explosiones combinadas. El motor gráfico aceptará tu respuesta como válida si el error absoluto o relativo es como máximo  $10^{-6}$ .

**Ayuda para el formato de salida:** Para imprimir un número de punto flotante  $r$  con precisión podés usar:

En Python: `print(f"{r:.10f}")`

En C++: `std::cout << std::fixed << std::setprecision(10) << r << "\n";`

Ejemplo de entrada 1	Ejemplo de salida 1
1 1 1 2 2 1	5.7123889804

Ejemplo de entrada 2	Ejemplo de salida 2
1 1 1 3 3 1	6.2831853072

# Problema I - Kilos de desperdicio

Seba sabe exactamente la cantidad de alimento que van a necesitar sus gatos cada día. Sin embargo, se enfrenta a un gran problema: sus gatos son tan exquisitos que se niegan a comer cualquier alimento que no haya sido comprado ese mismo día, obligando a Seba a ir a la tienda diariamente.

Para complicar más las cosas, en la tienda solo venden el alimento en  $P$  tamaños distintos de bolsas, donde la  $i$ -ésima bolsa trae exactamente  $r_i$  kilos. Como no siempre puede comprar la cantidad exacta que necesita, hay días en los que Seba se ve obligado a tirar a la basura los kilos que sobran.



A Seba le gustaría diseñar un programa que, dadas  $Q$  consultas (una por cada día) con la cantidad  $N$  de kilos necesarios, calcule cuál es la mínima cantidad de alimento que será desperdiciada al comprar al menos  $N$  kilos.

Por ejemplo, si necesita comprar 25 kilos y solo hay bolsas de 2 y 15 kilos, la mejor opción es comprar una bolsa de 15 y seis bolsas de 2 (o cualquier otra combinación que sume 27 kilos), lo cual daría un desperdicio mínimo de solo 2 kilos.

Lamentablemente, Seba se va de viaje y no tiene tiempo de programarlo, ¡así que te pidió ayuda para resolverlo!

## Entrada

La primera línea contiene dos enteros  $Q$  ( $1 \leq Q \leq 10^5$ ) y  $P$  ( $1 \leq P \leq 50$ ), la cantidad de días a consultar y la cantidad de tamaños de bolsas disponibles, respectivamente.

La segunda línea contiene  $P$  enteros distintos  $r_1, r_2, \dots, r_P$  ( $1 \leq r_i \leq 100$ ), donde el  $i$ -ésimo número indica que existe una bolsa de  $r_i$  kilos de alimento.

Las siguientes  $Q$  líneas contienen un entero  $N$  ( $1 \leq N \leq 5 \cdot 10^4$ ) cada una, indicando la cantidad mínima de kilos de alimento que se deben comprar ese día.

## Salida

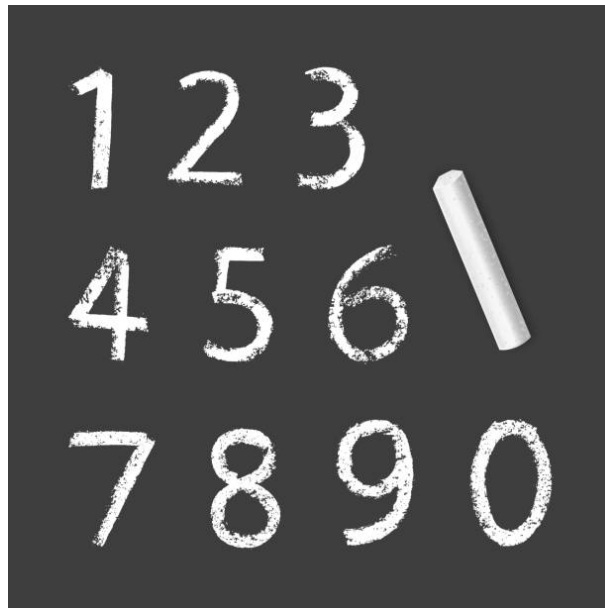
Para cada una de las  $Q$  consultas, imprimir una única línea con un entero que represente la mínima cantidad de kilos de alimento que será desperdiciada.

Ejemplo de entrada 1	Ejemplo de salida 1
2 3	0
2 15 7	1
10	
5	

Ejemplo de entrada 2	Ejemplo de salida 2
3 2	9
10 17	1
1	0
16	
27	

## Problema J - Real Divido

En una pizarra están escritos los números enteros del 1 al  $N$ . Ana y Beto son muy competitivos y deciden utilizar estos números para jugar un juego por turnos. En cada turno, el jugador activo debe elegir un número  $k$  que aún siga escrito en la pizarra y borrar tanto ese número  $k$  como todos sus divisores que permanezcan escritos. Por ejemplo, si un jugador elige el 6 y todos sus divisores están disponibles, borrará de la pizarra el 6, el 3, el 2 y el 1.



El jugador que en su turno no tenga números disponibles para elegir (porque la pizarra está vacía) pierde la partida.

Ana siempre es la primera en jugar. Si ambos jugadores juegan de forma completamente óptima, tu tarea es determinar quién será el ganador.

### Entrada

La primera línea contiene un único entero  $T$  ( $1 \leq T \leq 10^5$ ), la cantidad de casos de prueba.

Las siguientes  $T$  líneas contienen un único entero  $N$  ( $1 \leq N \leq 10^9$ ) cada una, que representa el número máximo escrito inicialmente en la pizarra para ese caso de prueba.

### Salida

Para cada caso de prueba, imprimir una única línea con el nombre del ganador: ANA si gana el primer jugador, o BETO si gana el segundo jugador.

Ejemplo de entrada 1	Ejemplo de salida 1
2	ANA
1	ANA
2	