



UPC eXtreme Programming

2023



UPC eXtreme Programming 2023

A. Desafío de Clase

2 seconds, 256 megabytes

Hoy en clase de Matemáticas el profesor de Kun dejó un desafío, el cual le brindará un puntaje extra en el examen parcial a quien logre resolverlo.

El desafío consiste en expresar un número  $n$  como producto de 2 números primos  $a$  y  $b$  diferentes ( $a < b$ ). Debido a que Kun ha tenido una nota baja en el parcial, desea que usted lo ayude a resolver el desafío y pueda de esta forma obtener el puntaje extra.

**Input**

La entrada consiste de un entero  $n$  ( $2 \leq n \leq 10^5$ )

**Output**

Imprimir los enteros  $a$  y  $b$  (en ese orden). En caso no haya solución imprimir -1.

| input  |
|--------|
| 6      |
| output |
| 2 3    |

| input  |
|--------|
| 8      |
| output |
| -1     |

B. Escapista

2 seconds, 256 megabytes

Álex y Bernard están atrapados en una prisión en un planeta extraño. Sin embargo, han descubierto que al guardia alienígena le gustan tanto los desafíos matemáticos como a ellos, así que lo convencieron de jugar un juego: Si ganan, saldrán libres. Caso contrario, ambos serán convertidos en nuggets para que los coman los Omicronianos (también llamados Popplers).

El juego se juega de la siguiente forma:

- En un tablero de  $n \times n$ , el guardia colocará una moneda en cada casilla ya sea con cara o sello, según él decida.
- Debajo de alguna de estas casillas (solo Bernard y el guardia sabrán donde) estará una llave.
- El guardia le dejará a Bernard voltear **exactamente una moneda, cualquier moneda**.
- Luego entregará a Álex el tablero tras el único cambio hecho por Bernard. Álex tendrá que adivinar debajo de cuál casilla estaba la llave.

Para ello, ambos amigos han convenido crear previamente al desafío una tabla de todos los posibles tableros y qué significarán para ellos: Cuando Alan vea cada tablero de caras o sellos, supondrá que la llave estaba debajo de la casilla marcada en sus notas. Así, Bernard le entregará las caras y sellos volteadas de alguna forma y Alan podrá adivinar dónde estaba la moneda.

**Input**

Un único entero  $n$ , el ancho y alto del tablero. Solo hay dos casos posibles:  $n = 2$ ,  $n = 4$ .

**Output**

Imprimir cada uno de los posibles tableros  $2^{n^2}$  de  $n \times n$  con c en cada casilla si la moneda está en cara o s si está en sello. Si Alan debe suponer que la llave está en la casilla, utiliza la mayúscula C o S.

El orden en el que imprimas los tableros es irrelevante. Si hay más de una solución, puedes imprimir cualquiera.

| input  |
|--------|
| 2      |
| output |
| Ss     |
| ss     |
| Cs     |
| ss     |
| sC     |
| ss     |
| cC     |
| ss     |
| ss     |
| Cs     |
| cs     |
| Cs     |
| sc     |
| cS     |
| cc     |
| cS     |
| ss     |
| sC     |
| cs     |
| sC     |
| sc     |
| Sc     |
| cc     |
| Sc     |
| sS     |
| cc     |
| cS     |
| cc     |
| Sc     |
| cc     |
| Cc     |
| cc     |

La estrategia en el caso de prueba siempre les permitirá salir de la cárcel. Por ejemplo, si la llave está en la casilla superior izquierda, sin importar cómo arregle el tablero el guardia, Bernard siempre tendrá alguna manera de voltear las monedas del tablero para que Alan vea alguno de los siguientes estados:

|   |   |
|---|---|
| S | s |
| s | s |

|   |   |
|---|---|
| C | s |
| s | s |

|   |   |
|---|---|
| S | c |
| c | c |

|   |   |
|---|---|
| C | c |
|---|---|



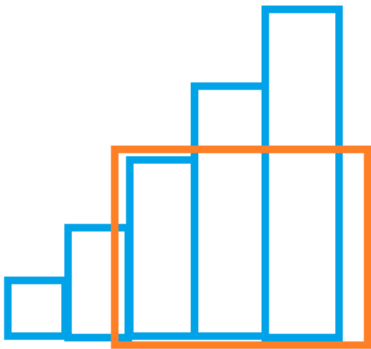
Por lo que, solo viendo las caras y sellos del tablero y comparándolas con la tabla que creó previamente, podrá adivinar dónde estaba la llave de la cárcel.

C. Edificios

2 seconds, 256 megabytes

La empresa "Industria Constructora de Productos Complejos" (ICPC) acaba de construir  $n$  edificios contiguos, cada uno de 1 metro de ancho y de alturas no necesariamente iguales.

Dadas la alturas de los edificios (de izquierda a derecha) se le pide hallar el edificio rectangular de mayor área que se hubiera podido construir, de tal modo que todo el interior de dicho edificio debería estar cubierto por regiones de los actuales edificios (ver imagen).



Input

La primera línea contiene un entero  $n$  ( $1 \leq n \leq 10^5$ ), denotando el número de edificios y en la siguiente línea se le darán las alturas  $h_i$  ( $1 \leq h_i \leq 10^6$ ) de los  $n$  edificios (de izquierda a derecha).

Output

Imprimir el área máxima del edificio que se hubiera podido construir.

| input     |
|-----------|
| 5         |
| 1 2 3 4 5 |
| output    |
| 9         |

En el caso de ejemplo el edificio rectangular de mayor área que se hubiera podido construir es de 3 x 3, ver imagen arriba.

D. Empezando con Potencia

2 seconds, 256 megabytes

Kun se encuentra participando en su primer concurso de programación y se enfrenta con el desafío de decir si un número  $n$  es una potencia de 2, es decir que  $n = 2^a$ , donde  $a$  es un entero no negativo.

Input

La entrada consiste de un entero  $n$  ( $1 \leq n \leq 10^8$ )

Output

Imprimir "SI" en caso el número sea una potencia de 2, caso contrario imprimir "NO"

| input  |
|--------|
| 4      |
| output |
| SI     |

| input |
|-------|
| 12    |

| output |
|--------|
| NO     |

E. Ensamblador

1 second, 256 megabytes

Eres programador en jefe de Intel, y te han encargado programar un emulador de un CPU muy antiguo. Este CPU solo tenía las instrucciones

- MOVer un operando a un registro.
- MULTiplicar dos operandos y guardar el resultado en un registro.
- DIVidir dos operandos y guardar el resultado en un registro.
- ADDicionar dos operandos y guardar el resultado en un registro.
- CMParar el valor de dos operandos para decidir si saltar a línea del programa o continuar la ejecución con la siguiente línea.
- INTerrumpir el flujo del programa para imprimir un operando en pantalla (**luego el programa continúa normalmente con la siguiente instrucción**).
- Un registro puede ser *EAX, EBX, ECX, EDX*. Todos son enteros con signo de 32 bits y su valor inicial es 0 al iniciar el programa.
- Un valor inmediato es un número entero con signo de 32 bits.
- Un operando puede ser un registro o un valor inmediato.

Se garantiza que el programa terminará en menos de  $10^5$  instrucciones.

Input

Recibirás las instrucciones del programa en cada línea del input, que siempre son de la forma:

- MOV <REG> <OPER>: Copia OPER hacia el registro REG, ahora REG=OPER.
- ADD <REG> <OPER1> <OPER2>: REG = OPER1 + OPER2
- MUL <REG> <OPER1> <OPER2>: REG = OPER1 \* OPER2
- DIV <REG> <OPER1> <OPER2>:
  - Si OPER2!=0, REG = OPER1 / OPER2 (división entera).
  - Si OPER2=0, el programa imprime "ERROR" (sin comillas) y termina.
- CMP <OPER1> <OPER2> <i>: Si OPER1 < OPER2 salto a la LÍNEA  $i$ -ésima (la primera línea es la línea número 1), sino continúa el programa.
- INT <OPER>: Imprimir OPER y un salto de nueva línea

Output

Imprimir lo que el emulador debería haber impreso en pantalla por cada INT

| input  |
|--|
| MOV EAX 3<br>MOV EBX 4<br>ADD ECX EAX EBX<br>INT ECX |
| output   |
| 7  |

| input   |
|---|
| MOV ECX 144<br>MOV EAX -1<br>ADD EAX EAX 1<br>MUL EBX EAX EAX<br>CMP EBX ECX 3<br>INT EAX |
| output  |
| 12  |

|                 |
|-----------------|
| input           |
| DIV EAX EAX EBX |
| output          |
| ERROR           |

|  |
|--|
| input  |
| MOV EAX 3<br>INT EAX<br>MOV EAX 4<br>INT EAX |
| output                                       |
| 3<br>4                                       |

|  |
|--|
| input  |
| MOV EAX 5<br>MOV EBX 2<br>DIV EAX EAX EBX<br>INT EAX |
| output   |
| 2  |

F. Carcelero

1 second, 256 megabytes

Este problema es idéntico a Escapista, pero ahora eres el guardia

Álex y Bernard están atrapados en una prisión en un planeta extraño. Sin embargo, han descubierto que al guardia alienígena le gustan tanto los desafíos matemáticos como a ellos, así que lo convencieron de jugar un juego: Si ganan, saldrán libres. Caso contrario, ambos serán convertidos en nuggets para que los coman los Omicronianos (también llamados Popplers).

El juego se juega de la siguiente forma:

- En un tablero de  $n \times n$ , el guardia colocará una moneda en cada casilla ya sea con cara o sello, según él decida.
- Debajo de alguna de estas casillas (solo Bernard y el guardia sabrán donde) estará una llave.
- El guardia le dejará a Bernard voltear **exactamente una moneda, cualquier moneda.**
- Luego entregará a Álex el tablero tras el único cambio hecho por Bernard. Álex tendrá que adivinar debajo de cuál casilla estaba la llave.

Para ello, ambos amigos han convenido crear previamente al desafío una tabla de todos los posibles tableros y qué significarán para ellos: Cuando Alan vea cada tablero de caras o sellos, supondrá que la llave estaba debajo de la casilla marcada en sus notas. Así, Bernard le entregará las caras y sellos volteadas de alguna forma y Alan podrá adivinar dónde estaba la moneda.

Determina si, dada la tabla que Álex y Bernard acordaron, los dejarás vivir o los convertirás en deliciosos nuggets

Input

Un único entero  $n$ , el ancho y alto del tablero.

Luego, cada uno de los posibles tableros  $2^{n^2}$  de  $n \times n$  con c en cada casilla si la moneda está en cara o s si está en sello. Si Alan debe suponer que la llave está en la casilla, utiliza la mayúscula C o S

Output

Imprimir

Vivos

si lograrán sobrevivir sin importar el reto que les pongas o

Nuggets

si como cacerero puedes asegurar una estrategia con la que no sobrevivirán.

|   |
|---|
| input   |
| 2<br>Ss<br>ss<br><br>cS<br>ss<br><br>sC<br>Ss<br><br>cC<br>sS<br><br>Ss<br>cS<br><br>cS<br>cS<br><br>sC<br>Cs<br><br>cC<br>cS<br><br>Ss<br>sC<br><br>cS<br>cC<br><br>sC<br>Cc<br><br>cC<br>cC<br><br>output |
| Nuggets   |

La estrategia en el caso de prueba falla: Por ejemplo, si el carcelero pone todas las monedas en Cara, no hay manera de que Bernard le comunique a Álex que la llave estaba en la casilla superior izquierda.

Esto es debido a que, volteando una sola moneda, no podría llegar a ninguno de los estados que le comunicarían a Álex que la llave está en la casilla superior izquierda:

|   |   |
|---|---|
| S | s |
| s | s |
| S | s |
| c | s |
| S | s |
| s | c |
| S | s |



G. Queso

1 second, 256 megabytes

Fuiste al mercado a comprar un bloque de 1000kg de queso que medía 1m<sup>3</sup> de tamaño. Pero te has dado cuenta que en realidad te han dado menos de lo que pediste. El problema es que no tienes una balanza, pero por alguna razón si tienes las coordenadas de cada burbuja de aire perfectamente esférica que había dentro del queso. **Las burbujas se pueden chocar entre sí, pero están contenidas totalmente dentro del queso.**

Quieres calcular cuál es el peso del queso que realmente te vendieron para luego ir a reclamar. Podemos asumir que el queso era perfectamente uniforme.

Input

Un entero positivo  $n \leq 2$ , la cantidad de burbujas en el queso. Luego,  $n$  líneas con 4 flotantes cada una: las coordenadas  $x, y, z$  y el radio  $r$  del queso, en ese orden. Todas las medidas están en metros. **Todas las burbujas están contenidas en el cubo desde la coordenada (0,0,0) hasta la coordenada (1,1,1).** Cada número será entregado con 6 dígitos de precisión.

Output

Un solo entero: el peso aproximado del queso en kilogramos. Tu solución será aceptada si tiene un error absoluto o relativo de 1% o menos.

| input                                    |
|--|
| 1<br>0.600000 0.400000 0.700000 0.300000 |
| output                                   |
| 886.902664                               |

| input   |
|---|
| 2<br>0.600000 0.400000 0.700000 0.300000<br>0.700000 0.700000 0.500000 0.200000 |
| output  |
| 854.812543  |

H. Boleto a México

2 seconds, 256 megabytes

Kun y Mako, expertos en algoritmia, están interesados en viajar a México, pero juntando sus ahorros solo disponen de dinero para comprar un boleto.

Para tomar la decisión de quién viajará, recurren al Grupo de Programación Competitiva (GPC). El GPC les propone un juego, de tal modo de quien gane dicho juego ganará el boleto a México.

El juego consiste de  $n$  estantes muy altos, cada uno con  $m_i$  cajones no muy anchos. Además el primer cajón del  $i$ -ésimo estante contiene  $x_i$  monedas, el segundo cajón tiene  $x_i + 1$  monedas, el tercer cajón contiene  $x_i + 2$  monedas, ..... y el último cajón contiene  $x_i + m_i - 1$  monedas.

EL GPC les indica que deben jugar en turnos alternados. En cada turno el jugador debe seleccionar un estante, luego debe seleccionar uno de los cajones de dicho estante y finalmente retirar una cantidad de monedas (mayor a 0) de dicho cajón. El jugador que no pueda retirar monedas pierde el juego.

El GPC decide que Mako jugará primero, luego se le pregunta a usted quien ganará el boleto a México si ambos juegan óptimamente (de la mejor manera posible)

Input

Problems - Codeforces

La primera línea contiene un número  $n$  ( $1 \leq n \leq 10^5$ ), denotando la cantidad de estantes. Luego siguen  $n$  líneas cada una con 2 enteros  $x_i$  ( $1 \leq x_i \leq 10^{15}$ ) y  $m_i$  ( $1 \leq m_i \leq 10^{15}$ ), detallados anteriormente.

Output

Imprimir "Mako" si Mako es el ganador, caso contrario imprimir "Kun".

| input           |
|-----------------|
| 2<br>3 1<br>2 2 |
| output          |
| Mako            |

| input           |
|-----------------|
| 2<br>1 1<br>1 1 |
| output          |
| Kun             |

I. Número eXtreme

2 seconds, 256 megabytes

Mako esta interesado en los números enteros que tienen todos sus dígitos diferentes, particularmente esta interesado en el "número eXtreme". Este número a parte de lo antes mencionado tiene la propiedad de no tener el dígito  $d$  y que al ser expresado en base 2 termina en  $d$  ceros.

Ahora se le pide a usted hallar el "número eXtreme" que se encuentre en el rango de  $L$  y  $R$  (inclusive) y que tenga el máximo valor posible.

Input

La entrada consiste de los enteros  $d$  ( $0 \leq d \leq 9$ ),  $L$  y  $R$  ( $10^8 \leq L \leq R \leq 10^9$ ), explicados anteriormente.

Output

Imprimir el "número eXtreme" en el rango solicitado y con el mayor valor posible. En caso de no existir solución imprimir -1.

| input                 |
|-----------------------|
| 4 100000000 100000003 |
| output                |
| -1                    |

| input                 |
|-----------------------|
| 4 300000000 500000000 |
| output                |
| 398761520             |