



Secuencia de Siguietes Múltiplos

A Ana le gusta jugar con números y hacer secuencias de ellos. Ahora está jugando a hacer secuencias de la siguiente forma: primero escoge un entero positivo m , que será el primer término de la secuencia, $a_1 = m$. Después, para todo $i \geq 2$, a_i es el menor múltiplo de i que es mayor o igual que a_{i-1} . Aquí hay algunos ejemplos de secuencias posibles para diferentes valores de m :

m	Secuencia
1	1,2,3,4,5,6,7, ...
2	2,2,3,4,5,6,7, ...
3	3,4,6,8,10,12, ...
4	4,4,6,8,10,12, ...
5	5,6,6,8,10,12, ...
7	7,8,9,12,15,18, ...
9	9,10,12,12,15,18, ...

Ana se ha fijado en que, para muchos valores de m , hay dos términos consecutivos de la secuencia que son iguales, es decir, existe un i tal que $a_i = a_{i+1}$. El caso $m = 5$ es un ejemplo, ya que $a_2 = a_3 = 6$. Pero hay otros números para los que esto no pasa. Por ejemplo, para $m = 1$ o $m = 3$ la secuencia tiene todos los términos distintos. Estos valores de m para los que la secuencia resultante no tiene dos números consecutivos iguales se llaman números *especiales*.

¿Puedes ayudar a Ana a encontrar el n -ésimo número especial?

Entrada y salida

La entrada consiste en un número entero positivo n .

Debes imprimir un entero, el n -ésimo número especial.

Ejemplos

Ejemplo 1

Entrada:

1

Salida:

1

Ejemplo 2

Entrada:

2

Salida:



3

Ejemplo 3

Entrada:

3

Salida:

7

Ejemplo 4

Entrada:

123456

Salida:

11970633999

Restricciones

$1 \leq n \leq 10^7$.

Subareas

1. (3 puntos) $n = 5$.
2. (8 puntos) $n \leq 20$.
3. (9 puntos) $n \leq 500$.
4. (65 puntos) $n \leq 10^5$.
5. (15 puntos) Sin restricciones adicionales.



Desordenado

Dada una lista $a = (a_1, \dots, a_n)$ de n enteros distintos, decimos que una reordenación (o permutación) $b = (b_1, \dots, b_n)$ de la lista a está *completamente desordenada* si toda sublista contigua no vacía de b no es igual a la sublista con los mismos índices de a pero ordenada. Es decir, si para todo $1 \leq i \leq j \leq n$, la secuencia de números $b_i, b_{i+1}, \dots, b_j$ no es igual a la secuencia de números $a_i, a_{i+1}, \dots, a_j$ pero ordenada.

Por ejemplo, dada la lista $(2, 5, 1, 4, 7)$, la permutación $(5, 7, 4, 2, 1)$ es completamente desordenada, pero las permutaciones $(7, 1, 4, 5, 2)$ y $(7, 5, 4, 1, 2)$ no lo son: en el primer caso, la sublista con $i = 2$ y $j = 4$ $(1, 4, 5)$ coincide con la sublista $(5, 1, 4)$ ordenada, en el segundo la sublista con $i = 2$ y $j = 2$ (5) coincide con la sublista (5) ordenada.

Dada una lista, se te pide que imprimas una permutación completamente desordenada.

Entrada y salida

La primera línea de la entrada contiene el número de casos T .

Por cada caso, la entrada tiene una línea con un entero n , seguida de una segunda línea con n enteros $a_1, \dots, a_n$.

Por cada caso, debes imprimir una línea con n enteros, una permutación completamente desordenada de a . Si hay varias posibles respuestas, puedes imprimir cualquiera de ellas.

Ejemplo

Entrada:

```
3
5
2 5 1 4 7
4
1 2 3 4
3
1 3 2
```

Salida:

```
7 1 4 5 2
3 4 2 1
3 2 1
```

Restricciones

$1 \leq T \leq 100$.

$3 \leq n \leq 100$.

$1 \leq a_i \leq 10^9$.

Todos los a_i son distintos.



XXVI Olimpiada Informática Española
Final día 2
desordenado

Subtareas

1. (37 puntos) $n \leq 8$.
2. (36 puntos) Para todo $1 \leq i \leq n$, $a_i = i$.
3. (27 puntos) Sin restricciones adicionales.



Comiendo galletas

Juan tiene galletas de n tipos distintos. Del i -ésimo tipo tiene a_i galletas. Juan quiere comerse todas las galletas, pero debe seguir una restricción: cada día debe haber un tipo de galleta t tal que el número de galletas que come del tipo t ese día debe ser igual al número total de galletas que come de tipos distintos a t ese día.

Por ejemplo, si $n = 3$ y $(a_1, a_2, a_3) = (4, 5, 7)$, una posible forma de comerse las galletas en dos días sería la siguiente:

1. El primer día se come 4 galletas del primer tipo, una galleta del segundo y 3 del tercero. En este día, se ha comido tantas galletas del tipo $t = 1$ como galletas de otros tipos distintos ($4 = 1 + 3$).
2. El segundo día se come 4 galletas del segundo tipo y 4 galletas del tercer tipo. Se ha comido tantas galletas del tipo $t = 2$ (o $t = 3$) como galletas de otros tipos se ha comido ($4 = 0 + 4$).

¿Puede Juan comerse todas las galletas? En caso afirmativo, se pide que diseñes una planificación de cuántas galletas de cada tipo debe comer cada día para cumplir su objetivo. Juan prefiere comerse las galletas en menos días, así que tu planificación deberá ser de como máximo $4n$ días, y para obtener la puntuación máxima la planificación debe ser de como máximo $\lceil \frac{n}{2} \rceil$ días.

Entrada y salida

La primera línea de la entrada contiene el número de casos T .

Por cada caso, la entrada tiene una línea con un entero n , seguida de una segunda línea con n enteros $a_1, \dots, a_n$.

Por cada caso, debes imprimir una línea con la palabra **SI** si es posible que Juan se coma todas las galletas, o con la palabra **NO** en caso contrario. En caso de que la respuesta sea **SI**, adicionalmente debes imprimir una línea con un entero d , seguida de d líneas, cada una con n enteros. El i -ésimo de la línea j es el número de galletas del tipo i que come Juan en el día j .

Ejemplo

Entrada:

```
4
3
4 5 7
2
2 1
5
1 3 2 11 5
7
1 999999999 1000000000 1000000000 1000000000 1000000000 1000000000
```



Salida:

```
SI
2
4 1 3
0 4 4
NO
SI
1
1 3 2 11 5
SI
3
1 999999999 1000000000 0 0 0 0
0 0 0 1000000000 1000000000 0 0
0 0 0 0 1000000000 1000000000
```

Restricciones

$$1 \leq T \leq 50.$$

$$2 \leq n \leq 1000.$$

$$1 \leq a_i \leq 10^9.$$

La suma de n para todos los casos es como máximo 1000.

El d que imprimas debe satisfacer $1 \leq d \leq 4n$.

El número de galletas que se come Juan de cada tipo debe ser un entero no negativo, menor o igual al número de galletas que quedan de ese tipo en ese día.

Subtareas

1. (5 puntos) $n = 3$.
2. (10 puntos) $n \leq 6$.
3. (9 puntos) Todos los a_i son iguales.
4. (12 puntos) Para todo $1 \leq i \leq n$, $a_i = i$.
5. (14 puntos) La suma de todos los a_i es menor o igual que $8n$.
6. (50 puntos) Sin restricciones adicionales.

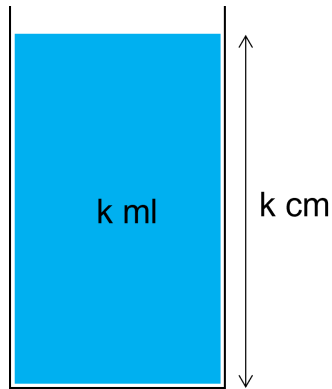
Adicionalmente, la puntuación que obtienes en una subtarea depende del valor de d . La puntuación de cada subtarea es multiplicada por un multiplicador $M(d, n)$: el valor que se toma de $M(d, n)$ es el menor entre todos los casos de la subtarea.

$$M(d, n) = \begin{cases} 0 & d > 4n \\ 0.4 + 0.4 \cdot \left(\frac{n}{2d}\right) & 4n \geq d > \lceil \frac{n}{2} \rceil + 5 \\ 0.8 + 0.04 \cdot \left(\lceil \frac{n}{2} \rceil + 5 - d\right) & \lceil \frac{n}{2} \rceil + 5 \geq d > \lceil \frac{n}{2} \rceil \\ 1.0 & \lceil \frac{n}{2} \rceil \geq d \end{cases}$$

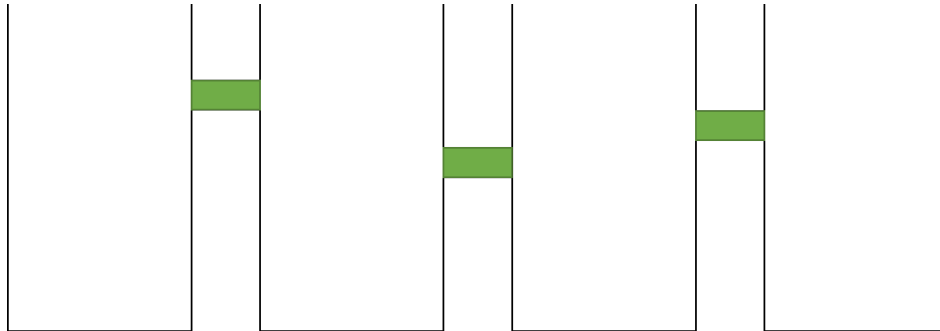


Vasos comunicantes

Alexandra ha estado aprendiendo esta semana sobre volúmenes en física y quiere hacer un experimento en casa. Tiene n vasos idénticos de plástico, que han sido diseñados para que la diferencia en centímetros entre el borde donde acaba el agua en el vaso y el fondo del vaso sea igual al número de mililitros de agua que este alberga.

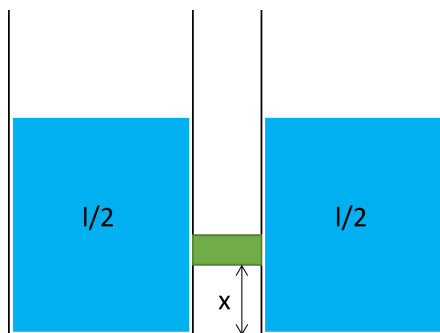


Alexandra ha puesto los vasos en una hilera y ha conectado el i -ésimo vaso con el $i + 1$ -ésimo para todo i , a través de un tubo horizontal. Para hacer el experimento más interesante, ha colocado los tubos a alturas sobre el fondo del vaso no necesariamente iguales:



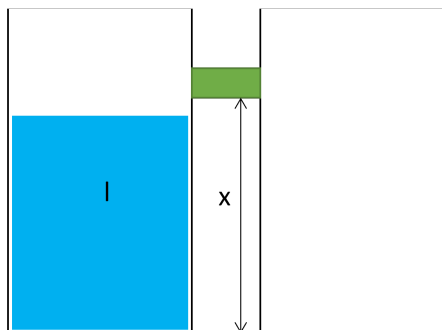


Si dos vasos contiguos están conectados por un tubo a una altura x , y echamos l mililitros de agua al primer vaso, si $x \leq \frac{l}{2}$, ambos vasos terminarán con $\frac{l}{2}$ mililitros.

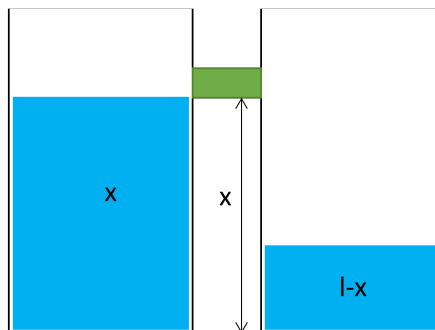


En cambio, si $x > \frac{l}{2}$, el primer vaso acabará con $\min(x, l)$ mililitros y el segundo con l menos esta cantidad.

Caso 1: $\min(x, l) = l$



Caso 2: $\min(x, l) = x$



Estos patrones se repetirán cuando haya más de dos vasos. Es decir, cuando empezamos a echar agua al primer vaso, inicialmente esta se queda toda en el primer vaso hasta llegar a la altura del primer tubo, momento en el que toda el agua que se sigue echando pasa a ir llenando el segundo vaso, manteniendo la altura del primero, hasta que el agua del segundo vaso llega a la altura del primer o del segundo tubo.

Si llega antes a la del primero, a partir de entonces el agua echada pasa a repartirse de forma igual entre los dos primeros vasos hasta que la del segundo llega a la altura del segundo tubo, momento en el que el agua que se echa pasa a estar toda rellenando el tercer vaso hasta que la altura de este llega a la del segundo o tercer tubo y se continúa la secuencia.

Sin embargo, si el agua del segundo vaso llega antes a la altura del segundo tubo, el agua que se vaya echando irá toda al tercer vaso hasta que esta llegue a la altura del segundo o tercer tubo y de nuevo continuaríamos la secuencia.

Las principales dos reglas a tener en cuenta es que si tenemos m vasos aislados y L mililitros de agua y las alturas de los tubos son todas iguales o inferiores a $\frac{L}{m}$ centímetros, habrá $\frac{L}{m}$ mililitros de agua en cada uno de los vasos y que, una vez que empieza a caer agua al i -ésimo vaso, hasta que la altura del agua en ese vaso no llegue a la del tubo anterior, la altura del agua en el vaso anterior en calma no aumentará.



Te diremos la lista de alturas A , donde A_i es igual a la altura entre el i -ésimo vaso y el $i + 1$ -ésimo (donde el primer vaso se numera con un 1). Sabiendo que ha echado x mililitros de agua en el primer vaso (asumimos que nunca se saldrá agua del vaso por arriba, ya que Alexandra la echa con sumo cuidado y nunca habrá más agua de la que puede soportar el sistema), ¿cuánta agua habrá en el j -ésimo cuando esta deje de correr? Deberás imprimir esta altura truncada para que sea un entero.

Una vez leída la lista de alturas, recibirás q consultas. Sin embargo, estas consultas pueden venirte en dos formatos. Antes de empezar a leerlas, recibirás un entero F , que será igual a 1 o 2. Si $F = 1$, recibirás q consultas con formato (x, j) , donde x y j representan lo explicado anteriormente. Sin embargo, si $F = 2$, recibirás q consultas con formato $(x + C, j)$, donde en la primera consulta $C = 0$ pero en las siguientes C será igual al resultado de la consulta anterior. Si $F = 2$ deberás responder las consultas con el valor original de x , no con $x + C$.

Entrada y salida

La entrada consiste en:

- Una línea con un entero n
- Una línea con las $n - 1$ alturas que conforman A
- Una línea con dos enteros: q y F
- q líneas con dos enteros: x y j (o $x + C$ y j si $F = 2$).

Debes escribir una línea por cada consulta con la cantidad de agua que habrá en el vaso.

Ejemplos

Ejemplo 1

Entrada:

```
3
1 2
6 1
1 1
1 2
2 2
3 2
10 2
15 3
```

Salida:

```
1
0
1
1
3
5
```



Ejemplo 2

Entrada:

```
10
1 3 4 2 3 4 5 3 1
20 2
0 1
6 3
6 4
1 1
2 2
2 2
4 2
5 3
5 3
20 7
7 3
9 4
15 6
19 8
30 10
100 10
60 10
15 10
3 9
0 5
```

Salida:

```
0
0
0
1
0
1
1
0
0
0
1
0
0
0
0
10
5
0
0
0
```



Restricciones

- F será igual a 1 o 2
- $2 \leq n \leq 10^5$
- $1 \leq j \leq n$
- $0 \leq x \leq 10^9$
- $1 \leq q \leq 10^5$
- $1 \leq A_i \leq 10000$ para todo i

Subtareas

1. (4 puntos) $2 \leq n \leq 10$ y $1 \leq q \leq 20$.
2. (7 puntos) $A_i = A_s$ para todo i y todo s .
3. (10 puntos) $A_i \leq A_{i+1}$ para todo i .
4. (22 puntos) $A_i \leq 100$ para todo i .
5. (37 puntos) $F = 1$.
6. (20 puntos) Sin restricciones adicionales.



Reescritura

A Félix le gusta mucho jugar con *strings*. Hoy ha descubierto una forma de transformar *strings* y te pide ayuda para saber si puede transformar una *string* dada en otra.

Las transformaciones posibles vienen dadas por un conjunto de reglas con las siguientes propiedades:

- Las reglas dadas son de la forma $c \rightarrow ab$, donde a, b, c son caracteres.
- Cada regla $c \rightarrow ab$, se puede aplicar para sustituir una ocurrencia de c por ab . Por ejemplo si se tiene la regla $a \rightarrow bc$ podemos transformar la cadena $abca$ en $bc bca$ aplicando la regla sobre el primer carácter.
- Se pueden aplicar sucesivamente estas transformaciones. Por ejemplo con $a \rightarrow bc$, $b \rightarrow mm$ a partir de la *string* $abca$ podemos conseguir $bc bca$, $mmcbca$, $ammca$, $bcbcbc \dots$
- Se nos asegura que para cada carácter c hay como máximo una regla de la forma $c \rightarrow ab$.
- Se nos asegura que las reglas dadas *no forman ciclos*. Es decir, para cualquier carácter c , aplicando las reglas sucesivamente sobre la *string* que sólo contiene el carácter c , no podemos conseguir llegar a ninguna *string* distinta que contenga el carácter c .

Entrada y salida

La primera línea de la entrada contiene el número de casos T .

Por cada caso hay una primera línea con n donde n es el número de reglas. Siguen n líneas donde en la i -ésima hay una regla dada como $a \rightarrow bc$ donde a, b, c son caracteres. Sigue una línea con una cadena s y una cadena t .

Por cada caso debe imprimirse una línea con “SI” o “NO” (sin comillas) en caso que se pueda convertir s a t aplicando las reglas dadas o no se pueda respectivamente.

Ejemplo

Entrada:

```
2
1
a->bc
aa
bca
3
a->bc
b->cc
c->mn
abc
acmnx
```

Salida:

```
SI
NO
```



Restricciones

$$1 \leq T \leq 100$$

$$1 \leq n \leq 59$$

La suma de las longitudes de todas las s y t para todos los casos es como mucho $3 \cdot 10^5$.

Todos los caracteres que aparecen son letras minúsculas (a-z), mayúsculas (A-Z) o números (0-9).

Subtareas

1. (17 puntos) Sólo aparecen **a** y **b** como caracteres.
2. (18 puntos) $n = 1$.
3. (31 puntos) Suma de longitudes de s y t en cada uno de los casos es como mucho 60.
4. (34 puntos) Sin restricciones adicionales.