

Lo que cuesta sumar

Johnny Calculín sabe sumar números mentalmente a gran velocidad, pero el esfuerzo que le supone realizar una suma depende del valor de los sumandos. En concreto, sumar a más b le supone un esfuerzo igual a $a + b$ (independientemente de los valores concretos de a y b). Sencillo.

Cuando tiene que sumar más de dos números la cosa se complica, ya que el orden en que va realizando las sumas afectan al esfuerzo total empleado. Por ejemplo, si tiene que sumar 1, 2 y 3, puede sumar $1 + 2$ con un esfuerzo igual a 3, y después sumar $3 + 3$ con un esfuerzo igual a 6, lo que supone un esfuerzo total igual a 9. En cambio, si suma primero $2 + 3$ con un esfuerzo igual a 5 y después $5 + 1$ con un esfuerzo igual a 6, el esfuerzo total asciende a 11. Obviamente el resultado de la suma es siempre 6, por las propiedades conmutativa y asociativa de la suma.

Johnny se está preparando para un concurso de sumas y quiere averiguar cómo debería ir sumando los números para necesitar el mínimo esfuerzo. ¿Puedes ayudarlo?

$$\begin{array}{r} 8 \\ + 4 \\ \hline ? \end{array}$$

Entrada

La entrada está compuesta por diversos casos de prueba, ocupando cada uno de ellos dos líneas: la primera línea contiene un entero N (entre 1 y 100.000) que representa el número de sumandos, y la segunda contiene esos N sumandos, números enteros entre 1 y 1.000.000.

La entrada termina con un caso donde N es 0 que no debe procesarse.

Salida

Para cada caso de prueba se debe escribir una línea con el esfuerzo mínimo necesario para sumar los números de la entrada.

Entrada de ejemplo

```
3
1 2 3
4
3 1 4 2
4
30 40 50 60
1
5
0
```

Salida de ejemplo

```
9
19
360
0
```

Autor: Alberto Verdejo.

21

Unidad Curiosa de Monitorización

La *Unidad Curiosa de Monitorización* (UCM) se encarga de leer los datos proporcionados por una serie de sensores y enviar con cierta periodicidad los datos obtenidos y procesados a los usuarios que se han registrado previamente.

La UCM admite que los usuarios se registren proporcionando un *Identificador*, un número entre 10^3 y 10^9 que identifica de forma única al usuario, y un *Periodo*, el intervalo de tiempo que transcurrirá entre dos envíos consecutivos de información a ese usuario (entre 1 y 5.000). Es decir, cuando hayan pasado *Periodo* segundos desde que el usuario se registró, este recibirá la información de la UCM por primera vez; y después recibirá la información cada *Periodo* segundos.

Acaban de registrarse varios usuarios, todos con identificadores distintos. ¿Podrías decir a quiénes irán dirigidos los K primeros envíos de información? Si dos o más usuarios tienen que recibir la información al mismo tiempo, los envíos se realizan en orden creciente de sus identificadores de usuario.



Entrada

La entrada consta de una serie de casos de prueba. Para cada uno, primero aparece el número de usuarios que se acaban de registrar (no más de 25.000). A continuación, aparecen los datos de las órdenes de registro de cada uno de ellos (el identificador de usuario y la periodicidad con la que desea recibir información). Y por último aparece el número K de envíos cuyo destinatario se desea conocer ($1 \leq K \leq 100.000$). La entrada termina cuando el número de usuarios sea 0.

Salida

Para cada caso se escribirán los identificadores de los usuarios que recibirán los K primeros envíos, uno por línea.

Después de cada caso se escribirá ----.

Entrada de ejemplo

```
2
1234 200
9000 300
5
0
```

Salida de ejemplo

```
1234
9000
1234
1234
9000
----
```

Autor: Alberto Verdejo.

Ordenando a los pacientes en urgencias

En el Hospital ACR (*Aquí Curamos Rápido*) se han puesto a mejorar las Urgencias para que los enfermos que llegan con dolencias más graves sean atendidos antes que los demás. Para eso, han comprado una UCM (*Unidad de Catalogación Médica*) que es capaz de valorar instantáneamente el estado de un paciente con un número entero entre 0 y 1.000.000, donde 0 indica que su dolencia es menor (quizá incluso inexistente y sea un mero hipocondríaco) y 1.000.000 indica que el enfermo está casi caminando hacia la luz.



Por desgracia, la afluencia de enfermos es tal que incluso así es muy complicado saber rápidamente quién debería ser el próximo en ser atendido. No hacen más que entrar pacientes nuevos a la vez que los más graves son atendidos, y no es fácil mantenerlos ordenados. Cuando un médico queda libre, debe ser atendido el enfermo a la espera con una valoración más grave. Si hay dos pacientes evaluados con la misma gravedad, deberá ser atendido el que más tiempo lleve esperando.

Para ayudar en la tarea de decidir quién es el próximo, desde ACR se ha hecho un llamamiento para buscar ayuda entre los mejores programadores. ¿Eres tú uno de ellos?

Entrada

La entrada está formada por diversos casos de prueba. Cada caso comienza con una línea indicando el número n de eventos que ocurrirán (como mucho 200.000), y a continuación aparecen n líneas cada una con un evento. Un evento puede ser la llegada de un paciente nuevo, o la atención por parte de un médico que ha quedado libre del paciente más urgente. Los ingresos de pacientes nuevos se indican de la forma **I nombre gravedad**, donde **nombre** es una cadena de como mucho 20 caracteres (sin espacios) y **gravedad** es un número entre 0 y 1.000.000 con su estado (0 leve, 1.000.000 muy grave). Los eventos en los que se atiende al siguiente paciente se indican con el carácter **A**. Se garantiza que no habrá nunca eventos de tipo **A** si no quedan pacientes esperando.

La entrada termina cuando el número de eventos es 0.

Salida

Para cada evento de tipo **A** de cada caso de prueba se escribirá el nombre del paciente que es atendido en ese momento. Se atiende primero al paciente más grave y, en caso de igualdad entre dos o más pacientes, se elegirá entre ellos al que más tiempo lleve esperando.

Al finalizar el tratamiento de cada caso se escribirá una línea más con cuatro guiones (----).

Entrada de ejemplo

```
9
I Alberto 4000
I Pepe 3000
A
I Rosa 2000
I Laura 5000
A
I Sara 3000
A
A
O
----
```

Salida de ejemplo

```
Alberto
Laura
Pepe
Sara
----
```

Autores: Alberto Verdejo y Pedro Pablo Gómez Martín.

23

Multitarea

Los calendarios o agendas electrónicas controlan nuestra vida diaria. Para personas como yo, que no se nos da bien la multitarea, es importante tener como mucho una tarea planificada para cualquier minuto de nuestra vida.

En mi calendario hay dos tipos de tareas: tareas únicas y tareas periódicas. Las tareas únicas tienen un instante de comienzo, c y otro de finalización, f , lo que significa que la tarea me tendrá ocupado durante el intervalo de tiempo $[c, f)$. Las tareas periódicas tienen además de los instantes de comienzo y finalización de su primera aparición, un periodo de repetición. Se supone que estas tareas se repiten para siempre sin fin. Para simplificar las cosas, todos los tiempos se expresan en minutos desde un tiempo inicial 0. Por ejemplo, una tarea periódica con tiempo de inicio 5, tiempo de finalización 8 y periodo de repetición 100 ocurriría en los intervalos de tiempo $[5..8)$, $[105..108)$, $[205..208)$, etc...

Tu trabajo consiste en averiguar si mi calendario está libre de conflictos durante un periodo de mi vida. Se considera que dos tareas están en conflicto si y solo si sus intervalos de tiempo se superponen, por ejemplo $[2..5)$ y $[4..6)$ se superponen, pero no $[2..8)$ con $[9..10)$, o $[2..8)$ con $[8..10)$.



Entrada

La entrada está formada por una serie de casos de prueba, ocupando cada uno de ellos varias líneas.

En la primera línea aparecen tres números: N , que representa el número de tareas únicas; M , que representa el número de tareas periódicas; y T , que representa el número de minutos en los que quiero averiguar si hay o no conflictos. A continuación, aparecen N líneas, cada una con los instantes de comienzo y finalización de una tarea única. A estas les siguen M líneas más, cada una con tres números: el instante de comienzo y finalización de la primera aparición de una tarea repetitiva, y el periodo de repetición.

Siempre hay al menos 1 tarea y nunca más de 10.000. Todos los tiempos son números entre 0 y 10^8 . Se garantiza que todas las tareas tardan al menos un minuto y que los intervalos de una misma tarea periódica no solapan entre sí.

Salida

Para cada caso de prueba el programa escribirá SI si hay conflictos entre algunas tareas en el periodo de tiempo $[0..T)$, y NO en caso contrario.

Entrada de ejemplo

```
2 0 10
2 5
4 6
0 2 100
1 4 8
5 7 8
2 1 10
8 20
1 5
6 7 10
```

Salida de ejemplo

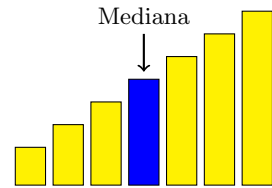
```
SI
NO
NO
```

Autor: Alberto Verdejo.

Cálculo de la mediana

En estadística, la *mediana* representa el valor de la variable de posición central en un conjunto de datos ordenados.

Sean x_1, x_2, \dots, x_n los datos de una muestra ordenada crecientemente. Si n es impar, la mediana es el valor que ocupa la posición central, $x_{(n+1)/2}$. Si n es par, la mediana es la media aritmética de los dos valores centrales, es decir, $(x_{n/2} + x_{n/2+1})/2$.



Entrada

La entrada está formada por diversos casos de prueba. Para cada caso, primero aparece en una línea el número n de datos de la muestra (como mucho 200.000), y a continuación, en otra línea, aparecen esos n valores según se han ido tomando (números enteros entre 1 y 1.000.000).

La entrada termina cuando una muestra no tiene valores (n es 0).

Salida

Para cada caso de prueba se escribirá una línea con los *dobles* de los valores de la mediana después de leer cada uno de los datos de la muestra. Escribimos el doble de la mediana para evitar escribir decimales en algunos casos.

Entrada de ejemplo

```
6
1 2 3 4 5 6
6
8 2 14 4 22 11
0
```

Salida de ejemplo

```
2 3 4 5 6 7
16 10 16 12 16 19
```

Autor: Alberto Verdejo.