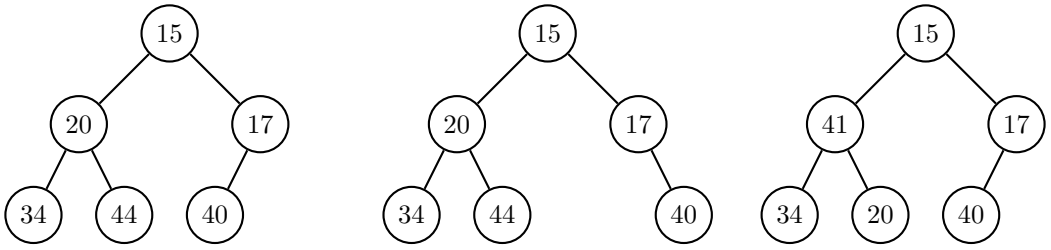
6. ¿Es un montículo?

Un árbol binario es *completo* cuando todos sus nodos internos tienen dos hijos no vacíos, y todas sus hojas están al mismo nivel; y es *semicompleto* si o bien es completo o tiene vacantes una serie de posiciones consecutivas del último nivel empezando por la derecha, de tal manera que al rellenar dichas posiciones con nuevas hojas se obtiene un árbol completo.

Un *montículo de mínimos* es un árbol binario semicompleto donde cada elemento es menor o igual que sus hijos (si los tiene).

De los siguientes árboles solamente el de la izquierda es un montículo.



Dado un árbol binario, el problema consiste en decidir si es o no un montículo de mínimos.

Entrada

La entrada comienza con el número de casos que vienen a continuación. Cada caso de prueba consiste en una línea con la descripción de un árbol binario: primero aparece su raíz (un entero no negativo), y a continuación la descripción del hijo izquierdo y después la del hijo derecho. El número –1 indica el árbol vacío. Los árboles nunca contendrán más de 20.000 nodos.

Salida

Para cada árbol se escribirá `SI` si es un montículo de mínimos y `NO` en caso contrario.

Entrada de ejemplo

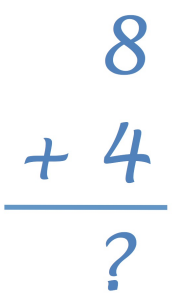
```
3
15 20 34 -1 -1 44 -1 -1 17 40 -1 -1 -1
15 20 34 -1 -1 44 -1 -1 17 -1 40 -1 -1
15 41 34 -1 -1 20 -1 -1 17 40 -1 -1 -1
```

Salida de ejemplo

```
SI
NO
NO
```

Autor: Alberto Verdejo.

7. 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 ayudarle?

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
3 4 5 6
0
```

Salida de ejemplo

```
9
19
36
```

Autor: Alberto Verdejo.

8. 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
0
```

Salida de ejemplo

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

Autores: Alberto Verdejo y Pedro Pablo Gómez Martín.

9. Pájaros en vuelo

Es muy conocido que algunas especies de aves vuelan creando una formación en V. La razón es simple: de esa forma las aves que van detrás aprovechan el rebufo de las que van delante. Esto implica que el pájaro que hace más esfuerzo es aquél que va primero ocupando el vértice de la V, por lo que es importante elegir bien quién es.

Algunas especies lo que hacen es colocarse en orden de edad de izquierda a derecha, de forma que el pájaro que queda en el centro es el que no es ni demasiado joven ni demasiado viejo. De esta forma evitan que el que abre camino esté poco desarrollado o demasiado cansado para tirar del grupo. Cuando a la bandada se van uniendo más pájaros, ocupan su lugar y, si es necesario, el pájaro que abre el camino cambia.

Por ejemplo, imaginemos que hay una bandada en vuelo con pájaros de edades 10, 20 y 30 meses. En ese caso, el que ocupará el primer lugar es el de 20 meses. Si ahora llegan dos pájaros nuevos, uno de 25 meses y otro de 35, ambos ocuparán su sitio y el primero pasará a ser el de 25 meses. Si posteriormente llega un abuelo de 40 meses con su nieto de 5, ocuparán los extremos y el primer puesto no variará.

Lo que queremos es simular la formación de una de estas bandadas de pájaros que comienza con un único ejemplar y a la que se van añadiendo nuevas parejas. Cada vez que se añade una pareja queremos determinar la edad del ave que ocupa la punta de la bandada.

Entrada

La entrada estará compuesta por una serie de casos de prueba, cada uno con la información de creación de una bandada de pájaros. Cada caso consta de dos líneas: la primera contiene la edad del primer pájaro que echa a volar seguido del número de parejas que se irán uniendo después (al menos una y hasta 100.000 parejas); en la línea siguiente aparecen las edades de cada pareja.

Se garantiza que las edades de cada uno de los pájaros que entran en la bandada son distintas. Para eso éstas vienen expresadas en segundos, y nunca será un número mayor a 100.000.000.

La entrada termina con una línea con dos ceros.

Salida

Para cada caso de prueba se escribirá una línea que tendrá tantos números como parejas se incorporan a la bandada, indicando la edad del pájaro que ocupa la primera posición tras la incorporación de cada pareja.

Entrada de ejemplo

```
30 3
10 20 35 25 5 40
0 0
```

Salida de ejemplo

```
20 25 25
```

Autor: Marco Antonio Gómez Martín.