

2.(3 puntos) En Falcon Heights (Minnesota) la empresa FalHeights quiere ser la primera en usar exclusivamente drones para la entrega de paquetes a sus clientes. Para ello, los dueños han comprado una pequeña flota de flamantes drones, cuya principal limitación es la batería que se puede usar. Por ello, planificar los repartos se ha convertido en un desafío.

Los ingenieros de FalHeights conocen de cuántos drones disponen para hacer los repartos, así como de la lista de pesos (en kg) de los paquetes a entregas. Lo que buscan es la capacidad mínima de batería necesaria para hacer todos los repartos teniendo en cuenta lo siguiente:

- Por política de empresa, los repartos deben realizarse en estricto orden de compra (no se pueden reorganizar las entregas).
- El gasto de batería de un dron es proporcional al peso total de los paquetes transportados. En concreto, entregar 1 kg de carga supone necesitar 2 hA (hectoamperios) de batería.
- Sale más barato comprar todas las baterías de los drones de la misma capacidad.

Por ejemplo, si disponemos de dos drones y hay que entregar 3 paquetes de pesos 4 2 3, la batería mínima será de 10 hA (un dron repartirá 4 kg y otro 5 kg).

Se pide:

1. (2.5 puntos) Diseña un algoritmo recursivo eficiente que permita resolver el problema.
2. (0.5 puntos) Escribe la recurrencia que corresponde al coste de la función recursiva e indica a qué orden de complejidad asintótica pertenece dicho coste.

Entrada

La entrada consta de varios casos de prueba. Cada uno de ellos ocupa dos líneas. La primera línea indica la cantidad N de paquetes ($1 \leq N \leq 10^8$) y el valor d ($1 \leq d \leq N$) de drones que dispone la empresa. La segunda línea contiene los N pesos (en kg) de los paquetes, todos ellos valores enteros positivos. La entrada termina con 0 0.

Salida

Para cada caso de prueba debe escribirse una línea indicando la capacidad mínima necesaria de las baterías (en hA) para poder realizar todas las entregas con la cantidad de drones disponibles.

Entrada de ejemplo

```
3 2
4 3 2
3 3
4 3 2
3 1
4 3 2
5 2
3 5 2 4 1
0 0
```

Salida de ejemplo

```
10
8
18
16
```