

---

**Muralla china****P71312\_es**

Qin Shi Huang fue el primero de los emperadores de China. Entre muchas otras hazañas, es famoso por iniciar la construcción de la Gran Muralla China, y por aplicar la política de “Quemar los clásicos y enterrar los eruditos”, que consistía en suprimir la libertad de expresión, destruir todos los libros (excepto los de su propia escuela de pensamiento, y algunos manuales de ciencias prácticas, como la medicina, la agricultura o la adivinación) y matar a todos aquellos que intentaran salvar el conocimiento antiguo. En este problema deberás convencer a Qin Shi Huang que la informática puede ser, al menos, tan útil como la adivinación. Por ello, te pedimos que hagas un programa que descubra el mejor modo de posicionar soldados que protejan la Gran Muralla China del ataque de los mongoles.

Asumiremos que la Gran Muralla China es una línea recta dividida en  $N$  segmentos de longitudes  $x_1, x_2, \dots, x_N$ . Entre cada par de segmentos, y en los extremos del primer y del último segmento, hay una torre con espacio para alojar a un batallón. El problema consiste en decidir cómo situar  $k$  batallones en las  $N + 1$  torres, de modo que se minimice la distancia entre cualquier punto de la muralla y una de las  $k$  torres ocupadas.

Por ejemplo, consideremos la siguiente muralla, que se corresponde con el Ejemplo 1:

0 2	10	18	24	40	44	52
-----	----	----	----	----	----	----

Si situáramos  $k = 2$  batallones en los puntos  $x = 24$  y  $x = 40$ , el punto que eligirían los mongoles para asaltar la muralla (el punto de la muralla que está más lejos de uno de las torres ocupadas) sería sin duda  $x = 0$  (está a distancia 24):

0 2	10	18	24	40	44	52
-----	----	----	----	----	----	----

Es preferible, pues, situar los batallones en  $x = 18$  y  $x = 40$  (el punto  $x = 0$  está ahora a distancia 18 del primer batallón):

0 2	10	18	24	40	44	52
-----	----	----	----	----	----	----

O, mejor aún, situarlos en  $x = 10$  y  $x = 40$  (ahora, el punto más vulnerable es  $x = 25$ , que está a distancia 15 de ambos batallones):

0 2	10	18	24	40	44	52
-----	----	----	----	----	----	----

En cambio, si tuviéramos 3 batallones, los deberíamos situar en  $x = 10$ ,  $x = 24$  y  $x = 44$ , y obtendríamos una distancia de 10 (esta vez atacarían por  $x = 0$  o por  $x = 34$ ):

0 2	10	18	24	40	44	52
-----	----	----	----	----	----	----

## Entrada

Una entrada está formada por un número indeterminado de casos de prueba. Cada caso de prueba se da en dos líneas. La primera línea contiene los números  $n > 0$  y  $k > 0$ , separados por espacios. Se cumple que  $k \leq n$ . La segunda línea contiene  $n$  números pares  $x_1, \dots, x_n > 0$  con las longitudes de los segmentos.

## Salida

Escribe tantas líneas como casos de prueba. Para cada caso de pruebas, debes escribir cuál es la mínima distancia que es posible conseguir situando del mejor modo posible los  $k$  batallones. Dado que todas las longitudes son pares, es seguro que la solución será un número entero positivo.

## Puntuación

Hay 10 grupos de entradas distintas. Tu programa recibirá 10 puntos por cada grupo de entradas resuelto correctamente en no más de 1 segundo de CPU por entrada. Los casos de prueba de la entradas del grupo  $i$ -ésima tendrán valores  $n$  no superiores a 1, 3, 5, 10, 30, 100, 1000, 10000, 30000, 100000. En las entradas de los primeras 4 grupos, las longitudes de los segmentos de las murallas no serán superiores a 100; en los siguientes 4 grupos, las longitudes no serán superiores a 10000; en los últimos 2 grupos, las longitudes no serán superiores a  $10^{12}$ . Ninguna entrada contendrá más de 100 casos, ni más de  $2 \cdot 10^6$  caracteres.

### Ejemplo de entrada 1

```
7 2
2 8 8 6 16 4 8
7 3
2 8 8 6 16 4 8
```

### Ejemplo de salida 1

```
15
10
```

### Ejemplo de entrada 2

```
1 1
100
1 2
100
```

### Ejemplo de salida 2

```
100
50
```

### Ejemplo de entrada 3

```
2 1
46 90
2 2
46 90
2 3
46 90
3 2
1000000000 1000000002 1000000000
```

### Ejemplo de salida 3

```
90
46
45
1000000000
```

## Información del problema

Autoría: Omer Giménez

Generación: 2026-01-25T11:38:33.810Z

© Jutge.org, 2006–2026.

<https://jutge.org>