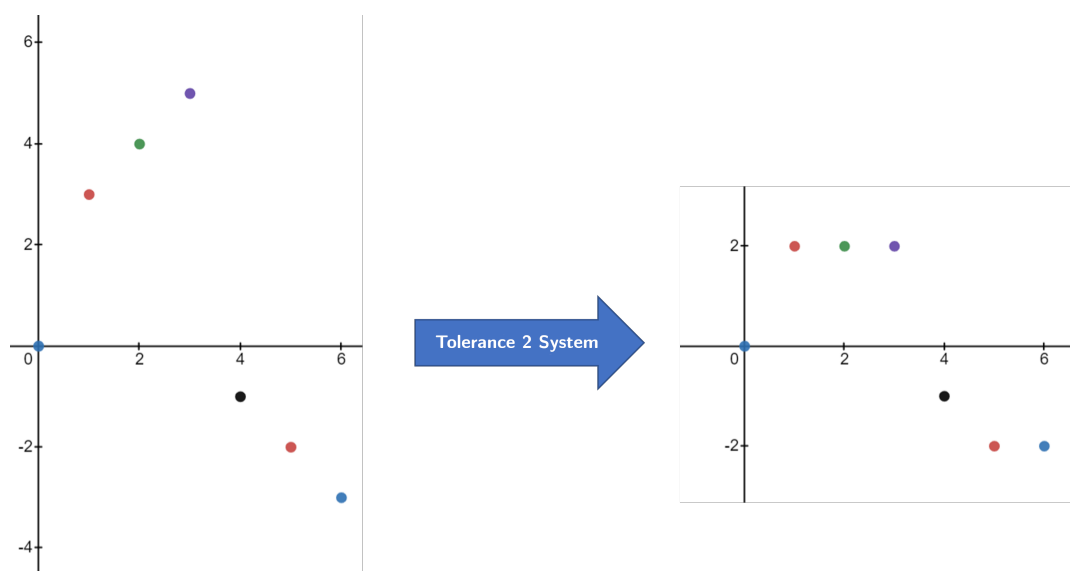


## Plane Turbulences (turbulences)

Brianair, una compagnia aerea di fama internazionale, sta conducendo uno studio sulle turbolenze. Stanno lavorando per trovare il sistema di stabilizzazione ottimale per i loro aerei. Questo sistema deve essere in funzione per tutta la durata del volo, tranne durante il decollo e l'atterraggio, cioè durante la parte del volo in cui l'aereo dovrebbe volare *in linea retta*.

Un sistema di stabilizzazione di *tolleranza*  $x$  assicura che l'aereo non si discosti dalla sua altitudine desiderata (quella che avrebbe se stesse volando in linea retta a quota costante) di più di una differenza assoluta di  $x$ . È possibile conoscere in anticipo l'altitudine dell'aereo ad ogni minuto del viaggio se non lo si equipaggia con un sistema di stabilizzazione. Vi verranno fornite tutte queste previsioni di deviazione di altitudine  $A_0, \dots, A_{N-1}$  per la durata del viaggio  $N$ , in ordine cronologico.

Il seguente esempio mostra come un sistema di stabilizzazione di tolleranza 2 porta un volo con previsioni di deviazione di altitudine  $A_0 = 0, A_1 = 3, A_2 = 4, A_3 = 5, A_4 = -1, A_5 = -2, A_6 = -3$  a un volo con deviazioni effettive  $B_0 = 0, B_1 = 2, B_2 = 2, B_3 = 2, B_4 = -1, B_5 = -2, B_6 = -2$ .



Altitudini prima e dopo l'applicazione di un sistema di stabilizzazione con tolleranza 2.

Brianair sa che i loro clienti amano i viaggi ad alta quota, quindi la soddisfazione del cliente (cioè il guadagno della compagnia aerea dall'implementazione del sistema) dopo aver volato su un aereo con un sistema di stabilizzazione di tolleranza  $x$  è pari a  $\sum_{i=0}^{N-1} B_i$ , dove  $B_i$  è l'altitudine stabilizzata al tempo  $i$ . In altre parole,  $B_i = \text{sign}(A_i) \cdot \min(|A_i|, x)$ .

Tuttavia, il costo di corrompere i regolatori per consentire un sistema con tolleranza  $x$  è pari a  $Kx$ , dove  $K$  è una costante non negativa. La compagnia aerea quindi vuole massimizzare il suo guadagno dal volo, cioè  $\left(\sum_{i=0}^{N-1} B_i\right) - Kx$ .

Dati  $K$  e  $A_0, \dots, A_{N-1}$ , saresti in grado di trovare il massimo guadagno che può essere ottenuto impostando la tolleranza ottimale  $x \geq 0$ ?

## Implementazione

Dovrai sottoporre un unico file sorgente, con estensione `.cpp`.

📖 Tra gli allegati a questo task troverai un template `turbulences.cpp` con un'implementazione di esempio.

Devi implementare la seguente funzione:

```
C++ | long long revenue(int N, int K, vector<long long> A);
```

- L'intero  $N$  rappresenta la durata del volo.
- L'intero  $K$  rappresenta il coefficiente di costo.
- L'array  $A$ , indicizzato da  $0$  a  $N - 1$ , contiene i valori  $A_0, A_1, \dots, A_{N-1}$ , dove  $A_i$  è l'altitudine prevista al tempo  $i$ .
- La funzione deve restituire il massimo guadagno che può essere ottenuto.

Il grader chiamerà la funzione `revenue` e stamperà il valore di ritorno sul file di output.

## Grader di prova

La cartella del problema contiene una versione semplificata del grader usato durante la correzione, che potete usare per testare le vostre soluzioni in locale. Il grader semplificato legge i dati di input da `stdin`, chiama la funzione che dovete implementare e scrive il valore di ritorno su `stdout`.

L'input è composto da  $N + 1$  righe, contenenti:

- Riga 1: gli interi  $N$  e  $K$ .
- Riga 2: gli interi  $A_i$ , separati da spazio.

L'output è composto da una sola riga, contenente il valore restituito dalla funzione `revenue`.

## Assunzioni

- $1 \leq N \leq 2 \times 10^5$ .
- $0 \leq K \leq 2 \times 10^5$ .
- $-10^{12} \leq A_i \leq 10^{12}$ .

## Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio associato a un subtask, devi risolvere correttamente tutti i test case contenuti in esso.

- **Subtask 1 [ 0 punti]:** Casi d'esempio.
- **Subtask 2 [15 punti]:**  $N = 1$ .
- **Subtask 3 [30 punti]:**  $N \leq 10^2$ ,  $K \leq 10^2$ ,  $-10^2 \leq A_i \leq 10^2$  for each  $i = 0, \dots, N - 1$ .
- **Subtask 4 [17 punti]:** Tutti i  $A_i$  sono uguali.
- **Subtask 5 [18 punti]:** Tutti i  $A_i$  sono non negativi.
- **Subtask 6 [20 punti]:** Nessuna limitazione aggiuntiva.

### Esempi di input/output

stdin	stdout
7 1 0 3 4 5 -1 -2 -3	1
5 1 7 8 -2 5 -10	3
5 0 1000000000000 1000000000000 1000000000000 1000000000000 1000000000000	5000000000000

### Spiegazione

Nel **primo caso di esempio**, la situazione è come descritta nella figura sopra. Il guadagno ottimale si ottiene con  $x = 5$ .

Nel **secondo caso di esempio**, il guadagno ottimale si ottiene con  $x = 5$ . Quindi, il guadagno totale è  $(5 + 5 + -2 + 5 + -5) - 1 \cdot 5 = 3$ .

Nel **terzo caso di esempio**, il guadagno ottimale si ottiene con una qualsiasi  $x \geq 10^{12}$ .