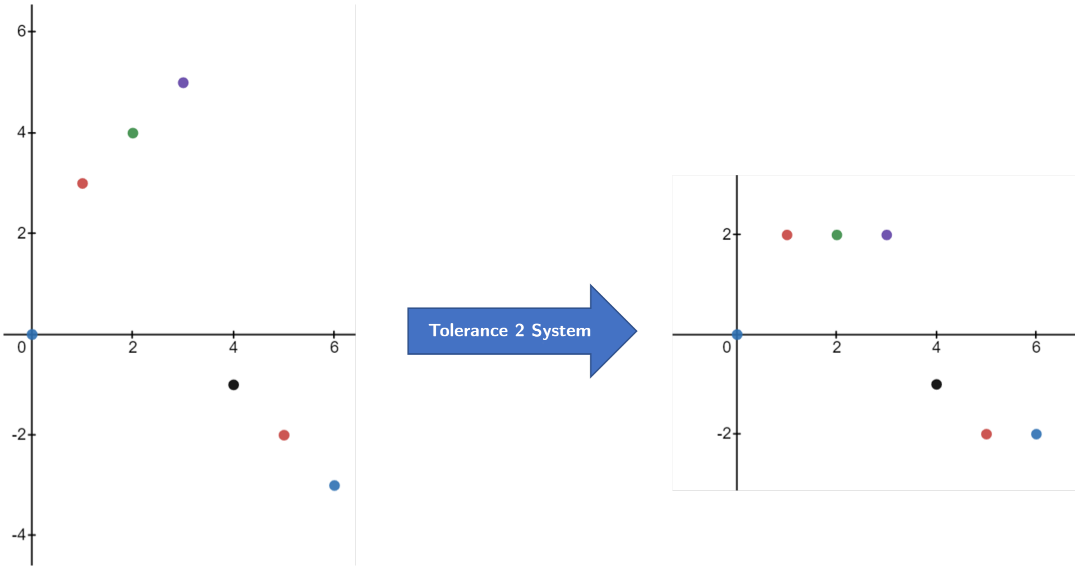
Plane Turbulences (`turbulences`)

A Brianair, uma companhia aérea de renome, está a realizar um estudo sobre turbulências. Estão a trabalhar para encontrar o sistema de estabilização ideal para os seus aviões. Este sistema deve estar em funcionamento durante todo o voo, exceto na aterragem e na descolagem, ou seja, na parte do voo em que o avião deve voar *em linha reta*.

Um sistema estabilizador de *tolerância* x assegurará que o avião não se desvia da altitude desejada (a que teria se estivesse a voar em linha reta a uma altitude constante) por uma diferença absoluta superior a x. É possível conhecer antecipadamente a altitude do avião em cada minuto da viagem, se não o equiparmos com um sistema de estabilização. São-lhe dadas todas estas previsões de desvios de altura $A_0, \ldots, A_{N-1}$ para a duração da viagem N, por ordem cronológica.

O exemplo seguinte mostra como um sistema estabilizador de tolerância 2 efectua um voo com previsões de desvio $A_0 = 0, A_1 = 3, A_2 = 4, A_3 = 5, A_4 = -1, A_5 = -2, A_6 = -3$ para um voo com desvios reais $B_0 = 0, B_1 = 2, B_2 = 2, B_3 = 2, B_4 = -1, B_5 = -2, B_6 = -2$.



Altitudes antes e depois da aplicação de um sistema de estabilização com tolerância 2.

A Brianair sabe que os clientes adoram viagens de avião a grande altitude, pelo que a satisfação do cliente (ou seja, o ganho da companhia aérea com a implementação do sistema) depois de voar num avião com um sistema estabilizador de tolerância x é igual a $\sum_{i=0}^{N-1} B_i$, em que B_i é a altitude estabilizada no momento i, ou seja, $B_i = \text{sign}(A_i) \cdot \min(|A_i|, x)$.

Contudo, o custo de subornar os reguladores para permitir um sistema com tolerância x é igual a Kx, em que K é uma constante não negativa. Por conseguinte, a companhia aérea pretende maximizar a sua receita com o voo, ou seja, $\left(\sum_{i=0}^{N-1} B_i\right) - Kx$.

Dados K e $A_0, \ldots, A_{N-1}$, serias capaz de encontrar a receita máxima que pode ser obtida definindo a tolerância óptima $x \geq 0$?

Implementação

Deves submeter um único ficheiro de código `.cpp`.

☞ Entre os ficheiros do problema encontrarás um template `turbulences.cpp` com um exemplo de implementação.

Tens de implementar a seguinte função:

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

- O inteiro N representa a duração do voo.
- O inteiro K representa o coeficiente de custo.
- O array A, indexado de 0 até $N-1$, contém os valores $A_0, A_1, \ldots, A_{N-1}$, onde A_i é a altitude prevista para o tempo i.
- A função deve devolver a máxima receita que pode ser obtida.

O avaliador irá chamar a função `revenue` e irá escrever o valor devolvido no ficheiro de output.

Avaliador Padrão

O diretório do problema contém uma versão simplificada do avaliador oficial, que podes usar para testar o teu problema localmente. O avaliador exemplo lê os dados de input de `stdin`, chama as funções que deves implementar, e finalmente escreve o output para `stdout`.

O input é feito de 2 linhas, contendo:

- Linha 1: os inteiros N e K.
- Linha 2: os inteiros A_i, separados por espaços.

O output é feito de uma única linha, contendo o valor devolvido pela função `revenue`.

Restrições

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

Pontuação

O teu program será testado num conjunto de casos de teste agrupados por subtarefa. Para obteres a pontuação associada a uma subtarefa, tens de resolver corretamente todos os casos de teste dessa subtarefa.

- **Subtarefa 1** [**0 pontos**]: Casos de exemplo.
- **Subtarefa 2** [**15 pontos**]: $N = 1$.
- **Subtarefa 3** [**30 pontos**]: $N \le 10^2$, $K \le 10^2$, $-10^2 \le A_i \le 10^2$ para cada $i = 0, \ldots, N-1$.
- **Subtarefa 4** [**17 pontos**]: Todos os A_i são iguais.
- **Subtarefa 5** [**18 pontos**]: Todos os A_i são não negativos.
- **Subtarefa 6** [**20 pontos**]: Nenhuma restrição adicional.

Exemplos

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

Explicação

No **primeiro caso de exemplo**, a situação é a descrita na figura do enunciado. A receita ótima é obtida com $x = 5$.

No **segundo caso de exemplo**, a receita ótima pode ser obtida com $x = 5$. Deste modo, a receita total será $(5 + 5 + -2 + 5 + -5) - 1 \cdot 5 = 3$.

No **terceiro caso de exemplo**, a receita ótima pode ser obtida usando um qualquer $x \geq 10^{12}$.