

PEARLS

Andrew has a string of N pearls (numbered 1 to N from left to right) that he plans to sell. The i th pearl has size d_i .

The *value* of a string of pearls is equal to the difference between the size of the largest and smallest pearl in the string. A string with zero or one pearls has a value of 0. Cunning as always, Andrew thinks he could cut the string of pearls into one or more substrings to increase the total value. However, each cut will cost him C dollars.

Once Andrew is finished cutting, he will sell all his substrings. His *profit* is the total value of his substrings, minus the cost of any cuts made. What is the maximum profit possible?

Subtasks and Constraints

For all subtasks:

- $1 \leq N \leq 1\,000\,000$.
- $0 \leq C \leq 1\,000\,000\,000$.
- $0 \leq d_i \leq 1\,000\,000\,000$, for all i .

Additional constraints for each subtask are given below.

Subtask	Points	Additional constraints
1	25	$N \leq 100$
2	15	$N \leq 2000$
3	15	$N \leq 100000$
4	30	$C = 0$
5	15	No additional constraints.

Input

- The first line of input contains the two integers N and C .
- The second line contains N integers d_1, d_2, \dots, d_N .

Output

Output a single integer, the maximum profit possible.

Note: Your solution may involve integers which are large. Consider using 64-bit integers ('long long' in C++) in your solution.

Note: Due to the large input size, if you are using `std::cin` in C++ it is recommended that you include the following two lines at the top of your main function (if you are using `scanf`, you do not need to include them):

```
std::ios_base::sync_with_stdio(false);
std::cin.tie(NULL);
```

Sample Input 1

10 25
70 30 40 60 50 50 100 200 40 50

Sample Output 1

210

Sample Input 2

8 0
30 40 10 10 50 0 10 100

Sample Output 2

170

Explanation

In Sample Input 1, Andrew can make two cuts, creating three substrings:

- [70 30], with a value of 40,
- [40 60 50 50 100], with a value of 60, and
- [200 40 50], with a value of 160.

This gives Andrew a profit of $40 + 60 + 160 - (2 \times 25) = 210$.

In Sample Input 2, Andrew can make two cuts, creating three substrings:

- [30 40 10], with a value of 30,
- [10 50], with a value of 40, and
- [0 10 100], with a value of 100.

This gives Andrew a profit of $30 + 40 + 100 - (2 \times 0) = 170$.