Data Collection

Time Limit: 1 Second
Memory Limit: 256 MB

Large Language Models (LLMs) are a type of autoregressive machine learning model that have achieved amazing performance on various natural language tasks, including question answering, code generation, and multi-hop conversation. A decoder-only LLM (i.e. GPT, Mistral, LLaMA, etc.) consists of several decoder layers and a task-dependent head, where each decoder layer consists of a self-attention module followed by a dense layer projecting the attention output to an embedding for each token.

Since LLMs usually contain billions of parameters, training an LLM to achieve the state-of-the-art performance requires multiple training steps. The LLM will be first trained on a large text corpus collected from diverse sources that cover most fields to predict the next token in the sequence. The pre-training will help the LLM to obtain a basic understanding of the natural language, which allows it to adapt to downstream tasks easily. Afterwards, a common fine-tuning technique known as reinforcement learning from human feedback (RLHF) is often adopted to align the LLM to human preferences. By leveraging RL techniques, the LLM learns to generate outputs that are more satisfying and relevant to the input from the perspective of human. Moreover, this fine-tuning step can also prevent LLM from generating toxic and sensitive contents.

You are working as a research assistant in an NLP lab at UIUC. Your group has proposed an improvement of LLM architecture that is likely to achieve the new state-of-the-art performance across various benchmarks with careful training. As a member of the group, you are asked to construct a new dataset that will be used to train your new LLM. You have scraped data from n sources, where the data from the i-th source has been processed into a dataset with a_i samples. Now you need to combine the samples from these n sources to create a large text corpus, and this step is expected to be quite time-consuming because there are a lot of samples to process. Therefore, you want to calculate the overall time needed to combine all samples together.

To process the samples, you have written a script that is able to combine two datasets into a single dataset, and for two datasets with a_i samples and a_j samples, the time needed to combine these two datasets is $a_i + a_j + c$ seconds, where c is a constant.

Input

The first line of input contains two integers n and c ($1 \leq n \leq 10^6, 0 \leq c \leq 10^9$), as described in the problem statement.

The second line contains n integer $a_1, \ldots, a_n (1 \leq a_i \leq 10^9)$ - the size of each dataset.

Output

Output a single integer indicating the minimum amount of time (in seconds) needed to combine all datasets into one single dataset using your script.

Sample Inputs

```
5 10
1 1 2 2 5
```

Sample Outputs

```
63
```

Note

One possible solution is:

1. Merge a_1 and a_2 into a_{12} of size 2, taking $1 + 1 + 10 = 12$ seconds.
2. Merge a_3 and a_4 into a_{34} of size 4, taking $2 + 2 + 10 = 14$ seconds.
3. Merge a_{12} and a_{34} into a_{1234} of size 6, taking $4 + 2 + 10 = 16$ seconds.
4. Merge a_{1234} and a_5 into a_{12345} of size 11, taking $6 + 5 + 10 = 21$ seconds.

The total time needed is $12 + 14 + 16 + 21 = 63$ seconds. It can be proved that this is the best solution.