

## A. Points on Line

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Little Petya likes points a lot. Recently his mom has presented him  $n$  points lying on the line  $OX$ . Now Petya is wondering in how many ways he can choose three distinct points so that the distance between the two farthest of them doesn't exceed  $d$ .

Note that the order of the points inside the group of three chosen points doesn't matter.

### Input

The first line contains two integers:  $n$  and  $d$  ( $1 \leq n \leq 10^5$ ;  $1 \leq d \leq 10^9$ ). The next line contains  $n$  integers  $x_1, x_2, \dots, x_n$ , their absolute value doesn't exceed  $10^9$  — the  $x$ -coordinates of the points that Petya has got.

It is guaranteed that the coordinates of the points in the input **strictly increase**.

### Output

Print a single integer — the number of groups of three points, where the distance between two farthest points doesn't exceed  $d$ .

Please do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

### Examples

<b>input</b>	<a href="#">Copy</a>
4 3 1 2 3 4	
<b>output</b>	<a href="#">Copy</a>
4	
<b>input</b>	<a href="#">Copy</a>
4 2 -3 -2 -1 0	
<b>output</b>	<a href="#">Copy</a>
2	
<b>input</b>	<a href="#">Copy</a>
5 19 1 10 20 30 50	
<b>output</b>	<a href="#">Copy</a>
1	

### Note

In the first sample any group of three points meets our conditions.

In the seconds sample only 2 groups of three points meet our conditions:  $\{-3, -2, -1\}$  and  $\{-2, -1, 0\}$ .

In the third sample only one group does:  $\{1, 10, 20\}$ .

### → Attention

Package for this problem was not updated by the problem writer or Codeforces administration after we've upgraded the judging servers. To adjust the time limit constraint, solution execution time will be multiplied by 2. For example, if your solution works for 400 ms on judging servers, then value 800 ms will be displayed and used to determine the verdict.

### Codeforces Round #153 (Div. 1)

**Finished**

### → Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

[Start virtual contest](#)

### → Problem tags

[binary search](#) [combinatorics](#)  
[two pointers](#) \*1300

No tag edit access

### → Contest materials

- [Announcement](#) ✕
- [Tutorial](#) ✕

[Codeforces](#) (c) Copyright 2010-2020 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: May/18/2020 15:29:15<sup>UTC-4</sup> (g2).  
Desktop version, switch to [mobile version](#).  
[Privacy Policy](#).

Supported by



ITMO UNIVERSITY