Problem C. air_show

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

small 5 points *2 minute timeout*	The contest is finished.
large 20 points *10 minute timeout*	The contest is finished.

Problem

Air Show

We are planning an awesome air show. The most impressive act features two airplanes doing acrobatic moves together in the air. The flight plan for each plane is already finalized. These plans may cause the planes to get very close to each other, which is dangerous; you have been hired to assess the extent of this risk.

A flight plan for one plane is a sequence of **N** timed segments. Within each segment a plane flies straight at a constant speed. However, a plane may change direction and speed dramatically when changing segments. Formally, the flight plan consists of an ordered list of **N** + 1 3-dimensional points P_0, P_1, ..., $P_{\mathbf{N}}$ and an ordered list of **N** transition times T_0, T_1, ..., $T_{\mathbf{N}-1}$. As its i-th move, for each i in {0, 1, ..., **N** - 1}, the plane following this plan flies from P_i to P_{i+1} at a constant speed in exactly T_i seconds. Since the planes are working together as a single act, the sum of the times for each segment must be equal for both planes.

A transition instant is a time in between two consecutive moves (even when two consecutive moves happen to require no change in speed or direction). That is, the **N** - 1 transition instants for a plane with the flight plan above happen exactly at times T_0, $T_0 + T_1$, $T_0 + T_1 + T_2$, ..., $T_0 + T_1 + ... + T_{N-2}$. The starting and finishing times are not considered transition instants.

Transition instants are the most dangerous times for pilots. Given a minimum safe distance **D**, for each plane *p*, we ask you to count the number of transition instants when *p* is at a distance strictly less than **D** from the other plane. You may consider each plane to be a single point. If both planes occupy the same point at the same instant, no collision occurs; the planes just pass through each other and continue.

The arithmetic for this problem for our official solution requires integers with more than 64 bits. `__int128` is available in our C++ installation and `BigInteger` is available for Java.

Input

The input library is called "air_show"; see the sample inputs below for examples in your language. It defines four methods:

- **GetSafeDistance()**:
  - Takes no argument.
  - Returns a 64-bit integer: the minimum safe distance **D**.
  - Expect each call to take 0.5 microseconds.
- **GetNumSegments()**:
  - Takes no argument.
  - Returns a 64-bit integer: the number of segments **N** of each flight plan.
  - Expect each call to take 0.5 microseconds.
- **GetTime(a, i)**:
  - Takes two 64-bit integers in the ranges 0 ≤ a < 2, 0 ≤ i < GetNumSegments().
  - Returns a 64-bit integer: the time used for move i of plane a (shown as T_i above).
  - Expect each call to take 0.5 microseconds.
- **GetPosition(a, i)**:
  - Takes two 64-bit integers in the ranges 0 ≤ a < 2, 0 ≤ i ≤ GetNumSegments().
  - Returns a 64-bit integer: an encoding of point i of the flight plan of plane a (shown as P_i above). Point (x, y, z), with each coordinate ranging between 0 and 2^{20}-1, is encoded as the integer $x \times 2^{40} + y \times 2^{20} + z$.
  - Expect each call to take 2.5 microseconds.

Output

Output a single line with two integers r_0 and r_1 separated by a single space, where r_i is the number of dangerous moments for plane i in which it is strictly closer than GetSafeDistance() to the other plane.

Limits

Number of nodes: 100. **(Notice that the number of nodes is the same for both the Small and Large datasets.)**
Time limit: 14 seconds.
Memory limit per node: 512 MB.
Maximum number of messages a single node can send: 1000.
Maximum total size of messages a single node can send: 128 KB. **(Notice that this is less than usual.)**
$1 \le$ GetSafeDistance() $< 2^{20}$.
$2 \le$ GetNumSegments() $\le 10^8$.

Small dataset

$0 \le$ GetPosition(a, i) $< 2^{40}$, for all a and i. (The x-coordinate of all points is 0, while coordinates y and z range between 0 and $2^{20} - 1$.)
GetTime(a, i) = 1, for all a and i.

Large dataset

$0 \le$ GetPosition(a, i) $< 2^{60}$, for all a and i. (All coordinates range between 0 and $2^{20} - 1$.)
$1 \le$ GetTime(a, i) $< 10^9$, for all a and i.
GetTime(0, 0) + GetTime(0, 1) + ... + GetTime(0, GetNumSegments() - 1) = GetTime(1, 0) + GetTime(1, 1) + ... + GetTime(1, GetNumSegments() - 1). (The sum of all the values of GetTime(a, _) for each a is the same.)
GetTime(0, 0) + GetTime(0, 1) + ... + GetTime(0, GetNumSegments() - 1) $\le 10^{12}$. (The total time of a flight plan does not exceed 10^{12}.)

Sample

Input	Output

`See input files below.`	`For sample input 1:`
	`1 1`
	`For sample input 2:`
	`0 1`
	`For sample input 3:`
	`3 1`

Note that the last 2 sample cases would not appear in the Small dataset.
Sample input libraries:
Sample input for test 1: air_show.h [CPP] air_show.java [Java]
Sample input for test 2: air_show.h [CPP] air_show.java [Java]
Sample input for test 3: air_show.h [CPP] air_show.java [Java]