Challenge 16 - ÑAPA

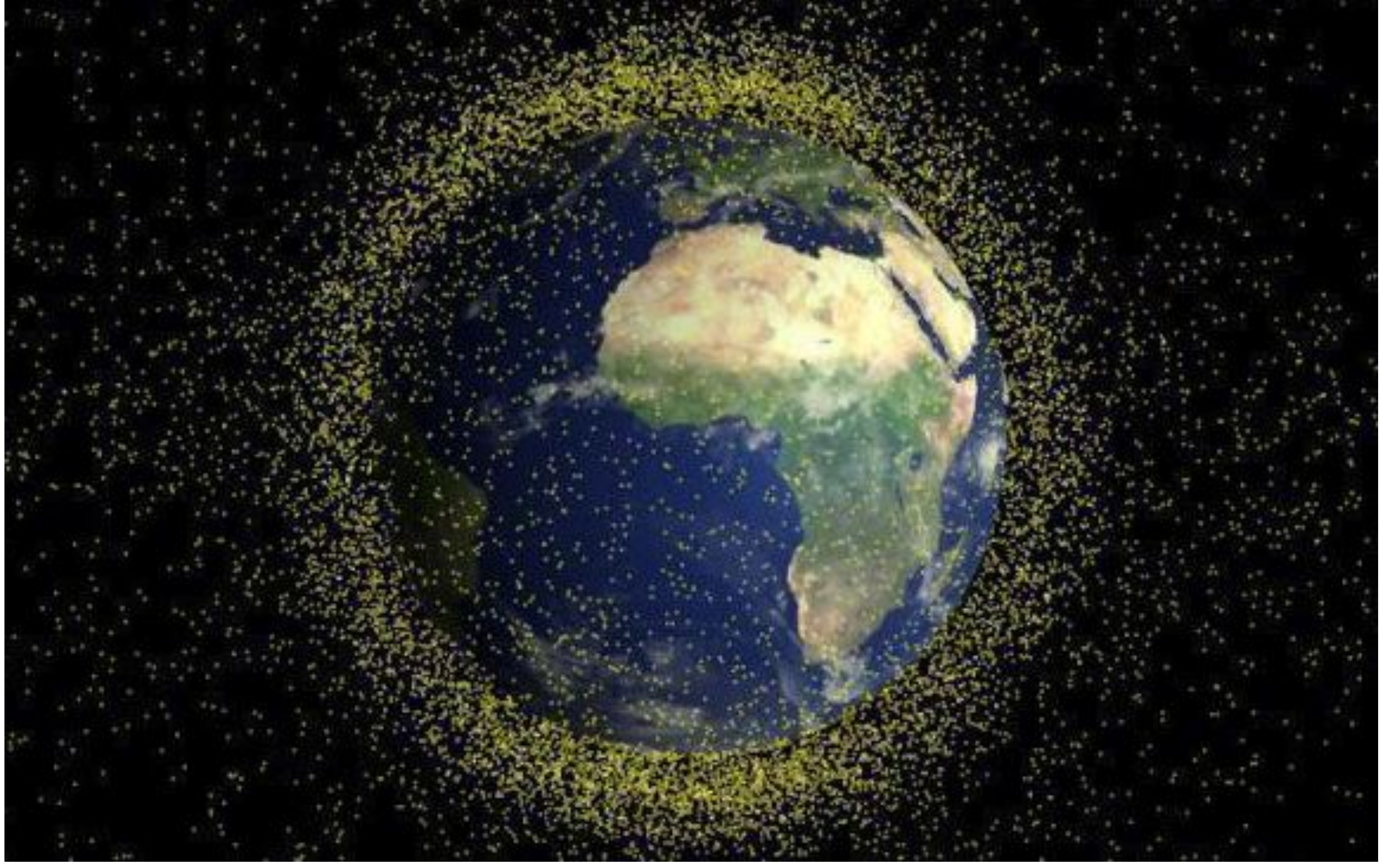
« Prev Next »

The Spanish government is in charge of developing the software to support the new ÑASA Atomic Particle Analyzer (ÑAPA) of the T250 (Javalambre Survey Telescope).

T250 Javalambre Survey Telescope at night

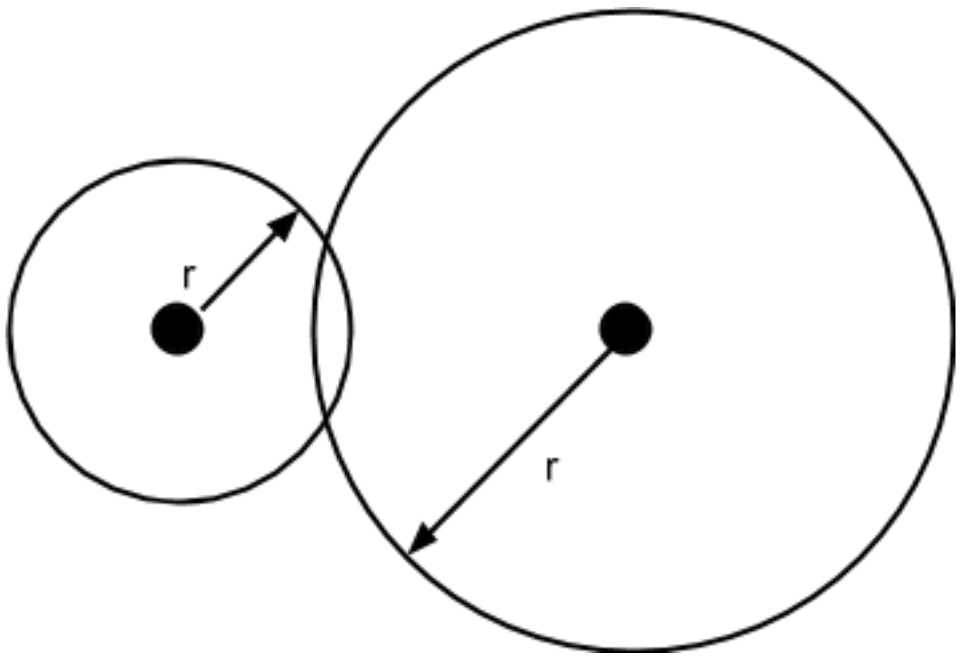
This telescope analyzes a small portion of the sky and reports N points mapped on a two-dimensional map with coordinates X and Y in which 0<=X and Y<100.000, for space dust, trash and small rocks that are on the same orbit.

The telescope also reports that points are circular with an estimated radius of R (1<=R<=500)



Dust and trash detected, full sky example, coarse grained detection

As the only capable software engineer in the team responsible for development, you must implement a program that, given the point report from the telescope, return the number of collisions.



A collision between a pair of points occurs when the distance is less than the sum of their radius.

The T250 team provides you with the first radar inspection (containing 3 million points) in the attached file.

For each line in the file, the first number is the X coordinate, the second is the Y coordinate, and the third is the R radius, separated by spaces and tabs.

Input

The input of the algorithm will be two positive integers separated by a comma. The first number is the nth point to read, and the second number will be the number of points to read.

Output

A positive integer with the number of collisions between pairs of points from the subset in the range specified in the input.

Example

For the following set of 12 points:

```
                                                54791               92148          43
                                                35138               75417          94
                                                87668               20721         454
                                                64455               33358         291
                                                40423               35057          15
                                                 2467               41977         784
                                                87438               28193         198
                                                20680               76562         278
                                                20930               75950         428
                                                56698               14029         492
                                                58959                3668         270
                                                60306               70806         268
```

With this input:

```
6,4
```

Will return this output:

```
1
```

This means that we will evaluate from the 6th point to the 9th point and there is one collision between the 8th and the 9th point.

Submit & test your code

To test and submit code we provide a set of tools to help you. Download contest tools if you haven't already done that. You will then be able to test your solution to this challenge with the challenge tokens.

```
Challenge tokens: CHALLENGE_16, CHALLENGE_SUBMIT_16
```

To test your program

```
./test_challenge CHALLENGE_16 path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

To test your program against the input provided in the submit phase

```
./test_challenge CHALLENGE_SUBMIT_16 path/program
```

During the submit phase, in some problems, we might give your program harder inputs. As with the test token, a nice output will tell you if your program got the right solution or not. You can try as many times as you need.

In the actual contest you first need to solve the test phase before submitting the code, you must provide the source code used to solve the challenge and you can only submit once (once your solution is submitted you won't be able to amend it to fix issues or make it faster).

If you have any doubts, please check the info section.

« Prev Next »