Nitro Scala Proficiency Test – Fun with spirals

The purpose of this test is to get some insights into your problem-solving skills and into your ability to write quality code. Unlike most tests, you can take as long as you like to complete it (at the same time we do not expect that it should take you more than 4 hours of effort to complete the test; if you are stuck, please contact talent organization). It is important that the result be demonstrably correct and robust. A single threaded solution is sufficient. Maximizing performance is not a factor in this test. It is expected that you use the Scala library to the maximum extent possible. You can also use 3rd party open-source libraries, if you think it is necessary and valuable.

When you are done, please zip-up the code and send it back to our talent organization. Ensure the repository has a README file in the root detailing any steps needed to build the code. How easy it is to build the program is relevant.

Test

Create a console program in Scala that will answer the following question:

Fill a two-dimensional NxN grid with NxN numbers (N is odd and the numbers are monotone increasing). Start in the middle/center and walk counter-clockwise to the outside. The middle square starts with 1.

```
17  16  15  14  13
18   5   4   3  12
19   6   1   2  11
20   7   8   9  10
21  22  23---> ...
```

Now given a location (one of the cell-values), calculate the Manhattan-Distance to the center.

For example:

From square/location 1 the distance is 0.
From location 12 the distance is 3 (down, left, left).
From location 23 the distance is only 2 (up twice).
From location 1024 the distance is 31.

How many steps are required to go from location 368078 to the center?

Judging

The submission will be judged by assessing, among other things, the following factors:

- **Correctness** – does it do what the spec requires?
- **Verification** – does the submission verify that it behaves as required?
- Readability – is it easily understandable?
- Clarity – does it make clear what it is trying to do?
- Simplicity – is the code relatively uncomplicated?
- Safety – is it resource and input safe?
- Expertise – does the code feel like it was written by an expert?
- Craftsmanship – how maintainable is the code by others?

Please be aware that failing to satisfactorily address correctness and verification will result in the submission being summarily rejected.

Feel free to use any open source third party libraries you like, but bear in mind that the person reviewing the code may not have the libraries in question available to them. Hence, either include them in your repo or include steps to get them automatically as part of you build script/project/solution/etc.

While you are expected to use online resources to research your submission, plagiarizing is considered bad form for a test. To obviate this, you will be asked at interview to explain your code, to evaluate its efficiency and to describe the modifications necessary when a change is made to the requirements.