

# Citrine Informatics Technical Challenge

## Scientific Software Engineer

Version: 839a6542536cb5066752204c983e515a89222414

The following technical challenge is meant to be representative of the math and engineering required to solve the problems in scope for our group. There is no time limit, but the problem is designed to take approximately 10 hours so please do not spend much more time on it than that! In addition to your code, please include a brief write-up that explains your solution.

### software

---

One of the core capabilities of Citrination is the ability to efficiently sample high dimensional spaces with complex, non-linear constraints. In this challenge, we are asking you to efficiently generate candidates that systematically explore as much of the valid space as possible.

The "API" of the challenge is file based: you must deliver a script that can be run as:

```
./sampler <input_file> <output_file> <n_results>
```

along with installation instructions. The input file starts with a single line header that gives the dimensionality of the problem, which is defined on the unit hypercube. The next line is a single example feasible point. The remaining lines are a list of constraints as python expressions containing `+`, `-`, `*`, `/`, and `**` operators. They have been transformed such that they all take the form `g(x) >= 0.0`. We've included several example input files for reference. We've also included a python class that can parse the input file and produce a function that evaluates correctness. The output file should contain a list of vectors (space delimited within the vector; one vector per line). We will evaluate for `n_results = 1000`, and expect the execution to take less than 5 minutes.

We will run your script on a held-out constraint list. The evaluation criteria has two components:

- Correctness: all of the produced vectors satisfy all of the constraints
- Efficiency: the vectors cover as much of the valid space as possible. You can think of this as them being "spread out" in the Euclidean space in which the valid space is embedded.

You are welcome to use whatever programming language/environment you'd like, as long as we are able to easily install and execute your script on our test constraint list.