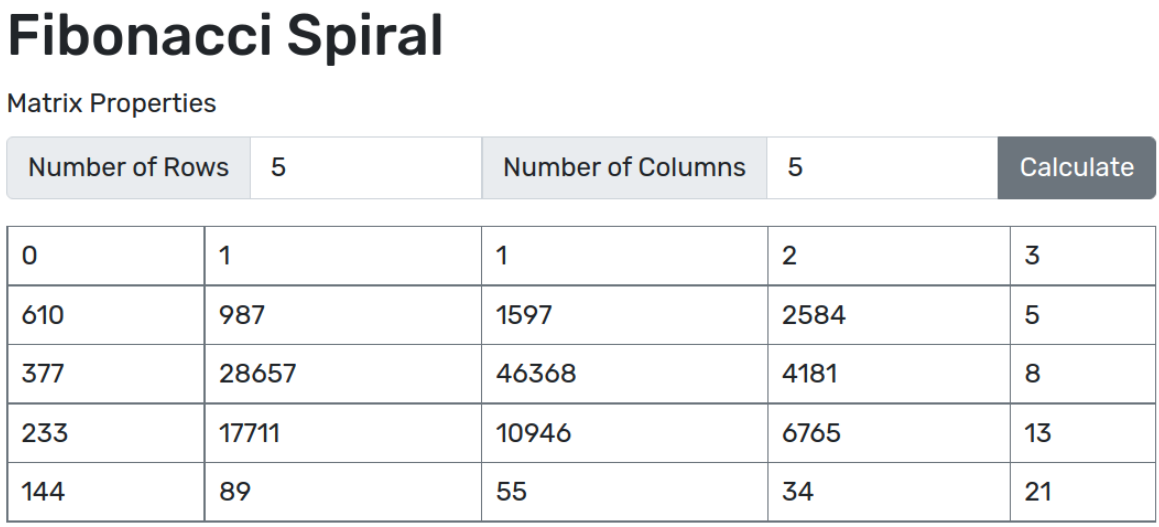
The Fibonacci Spiral Matrix

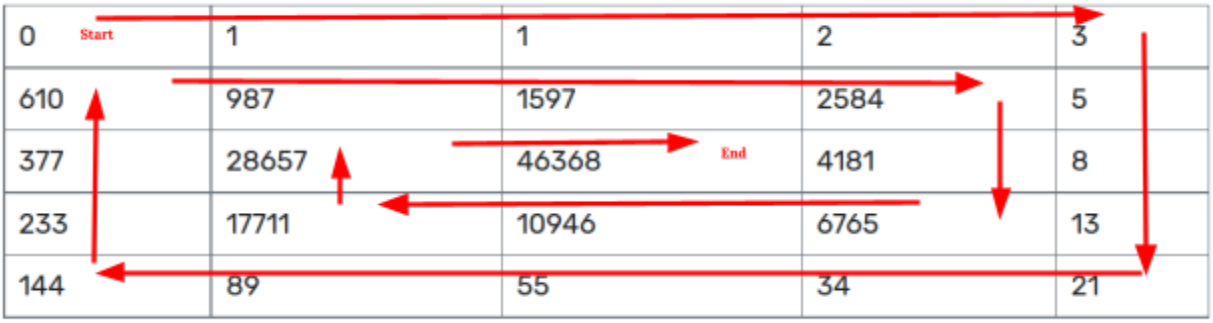
R2 Engineering: Take Home Project

Project Description

You will be building an API-like service and a frontend client to serve the famous **spiral matrix**, but with an interesting twist: *The numbers inside the matrix will be those of the Fibonacci sequence, up to the number of cells in the matrix.* This resulting frontend should look like the following example (at least, as similar as possible):



If you look closely, the matrix should represent the Fibonacci sequence in spiral ordering:



Your Tasks

- Design an API service which, given an input param, returns the response as a JSON value. E. g.:

```
curl http://localhost:8080/spiral?rows=5&cols=5
HTTP/1.1 200 OK
Content-Length: 168
Content-Type: application/json; charset=utf-8
Date: Thu, 17 Jun 2021 18:04:00 GMT

{
   "ts": "1623959127000",
   "rows": [
               [0, 1, 1, 2, 3],
               [610, 987, 1597, 2584, 5],
               [377, 28657, 46368, 4181, 8],
               [233, 17711, 10946, 6765, 13],
               [144, 89, 55, 34, 21]
           ]
}
```

- Implement a frontend site that can render the matrix using the provided rows and columns params (see the proposed interface above). At R2 we prefer React, but you can use whatever you feel comfortable with.
- Develop and expose the service using any framework/language of your preference. Try to exploit the features of the chosen framework/language to decouple the components.

- Upload your code to Github or Gitlab and provide viewer access to the user **jscappini**.

Bonus Points

- Implement an authentication mechanism.
- Provide a sequence diagram of your service components.
- Provide an architecture diagram of the overall implementation.
- Containerize your application.
- Think about how you would expose this service to the world. What are those considerations you should think about and what topics should be on your launch-list?

Things to Remember

This is as much an exercise in your ability to break a medium-sized project down into small bite-sized pieces as an engineering exercise.

Please be ready to talk through your approach as well as extend your service/application while we do pair-programming during the technical interview.

If you're unable to complete the take-home project in under 2-3 days, don't worry we can complete it during the interview. We'd much rather you to bring a well-thought-out but incomplete project, than something feature complete but fragile.

Feel free to reach out to js@r2.co if you have any questions.