

Software Engineer

Code Exercise

MAY 2018

Please attempt each of Q1 – Q5 if you can't finish any or ran into a serious blocker please add an explanation of what the issue was and what you tried to fix it.

Submit the results as code-test-<name>.zip containing the root directory of your project, include any notes in a README file.

**Estimated time to complete (2 - 3 Hours)**

Question 1.

### Build a word jumble function jumble(string, n)

### This function expects a string input and an integer n that is between 1 and 1000. It will transform each character of the input string using the following instructions:

* if it is a letter of the alphabet (a,b,c ... z) shift it to the right in the alphabet by n letters, if you reach z return to a and continue, however many times is required.
* if it is a number (1,2,3 ... ) or a space leave it the same.
* if it is any other character remove it from the string

it then returns the resulting string

**Example - Pseudocode**

jumble('test 123!', 0) == 'test 123'

jumble('test 123!', 1) == 'uftu 123'

jumble('test 123!', 100) == 'paop 123'

jumble('test 123!', 26) == 'test 123'

Question 2.

### Build Unit Tests for the Jumble function

Write appropriate unit tests for the function you wrote in Q1

Add instructions on how to run them in the README

Question 3.

### Build The Jumble Service

Build a REST endpoint POST /api/jumble/{n} which will accept a JSON payload with the message to be jumbled and n determined by the route.

// request

POST /api/jumble/1

{

"message": "test 123!"

}

// response

200 OK

{

"jumbled": "uftu 123"

}

In the README provide details of any libraries/packages you used to build it and any design considerations or assumptions you took as well as instructions required to setup development environment

Question 4.

### Build the Jumble Container

For this question, you will prepare the API you built for deployment. There are several options here, **please select one only.**

1. Add a Dockerfile which will build a container for your API
2. Deploy to a cloud environment of your choice and provide the URL for testing as well as deployment instructions in the README file
   1. Heroku
   2. AWS Lambda
   3. Cloud Functions
   4. AWS EC2
   5. Azure
   6. Any other cloud environment of your choice
   7. If you are not sure what to use try <https://signup.heroku.com/> - A good way of getting started with containers
3. If you can't deploy it to any of the above please provide instructions for getting it running on a windows environment

Please provide the URL of your service in the README if deployed online

Question 5.

### Design/Build a rate limit on the API  (Bonus)

Build or discuss possible solutions to the following problem....

JumbleAAS or Jumble as a Service has become the latest craze across the internet, now your brand new API is being attacked by hackers jumbling 1000's of words per minute.

Design a rate limiter mechanism that will respond with a **429** Too Many Requests to any user that makes more than 300 requests in a minute