# Challenge Introduction

* Candidates need to attempt all 3 challenges
* The response to the programming challenges can be implemented in your programming language of a choice, with a preference for Python 3.x if possible

# Challenge #1 - Programming

## Introduction

Publish a small service on the web that has two endpoints:

1. /messages takes a message (a string) as a POST and returns the SHA256 hash digest of that message (in hexadecimal format)
2. /messages/<hash> is a GET request that returns the original message. A request to a non-existent <hash> should return a 404 error.

## Example

|  |  |
| --- | --- |
|  | Let’s say you publish to [http://mywebsite.com/](http://yourwebsite.com/) (you don’t need a custom domain for this project, any IP address we can access will do):  $ curl -X POST -H "Content-Type: application/json" -d '{"message": "foo"}' [http://mywebsite.com/messages](http://yourwebsite.com/)  {  "digest": "2c26b46b68ffc68ff99b453c1d30413413422d706483bfa0f98a5e886266e7ae"  }  You can calculate that your result is correct on the command line:  $ echo -n "foo" | shasum -a 256  2c26b46b68ffc68ff99b453c1d30413413422d706483bfa0f98a5e886266e7ae -  You can now query your service for the original message:  $ curl http://mywebsite.com/messages/2c26b46b68ffc68ff99b453c1d30413413422d706483bfa0f98a5e886266e7ae  {  "message": "foo"  }  $ curl -i http://mywebsite.com/messages/aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  HTTP/1.0 404 NOT FOUND  Content-Type: application/json  Content-Length: 36  Server: Werkzeug/0.11.5 Python/3.5.1  Date: Wed, 31 Aug 2016 14:21:11 GMT  {  "err\_msg": "Message not found"  }  (your specifics may vary, all that matters is that you get a 404) |

Hint: When does ordering of messages you POST vs digests you GET matter?

## How to Submit

* Send us your source code by email or online repository
* Send us example curl calls by email or online repository
* Send us the URL of your service so we can test it with a few API calls

# Challenge #2 - Programming

Given a list of sorted values (let’s call them utxos), find exactly two distinct members of that list that are at or minimally over a certain amount (say, target). If no pair of utxos are at or over the target, we print “Not possible”

The program takes two inputs, the first is a filename with a list of values and the second input is the target value.

The file looks like this:

$ cat input\_file.txt

abcdef 17

e478ab 20

a84739 23

…

e4738a 137

fff483 141

The first column is a unique identifier of the utxo and the second is the value of that utxo,which is always a positive integer.

$ find-pair <filename> <some value>

You should return a pair of utxos and their values such that the sum of the values is greater than or equal to the target value.

Some examples:

$ cat utxos1.txt

abcdef 5

48a92b 7

e478ab 10

13474a 14

a84739 20

$ find-pair utxos1.txt 25

abcdef 5, a84739 20

$ cat utxos2.txt

abcdef 1

e478ab 2

74738a 2

a84739 22

$ find-pair utxos2.txt 25

Not possible

$ cat utxos3.txt

147bce 2

abcdef 6

e478ab 20

a84739 24

$ find-pair utxos3.txt 25

abcdef 6, e478ab 20

Note that you don’t have to return every possible pair that is minimally over the threshold, just one such pair. The goal is to be equal to the target value, and if that’s not possible, to be as minimally above the target value as possible.

Note: There may be a lot of rows in the file, so an optimized program is strongly preferred.

Bonus: Instead of choosing exactly 2 utxos, allow for 1 or more utxos. Warning: The bonus is NOT an easy problem!

# Challenge #3 - Programming

You are given a string composed of only 1s, 0s, and Xs.

Write a program that will print out every possible combination where you replace the X with **both** 0 and 1.

Some examples:

$ myprogram X0

00

10

$ myprogram 10X10X0

1001000

1001010

1011000

1011010

While your program will take longer to run based on the number of combinations you output, your program shouldn’t crash on a large input.

What is the big O notation for this program?