

## GivingForward Programming Assessment

### Software Engineer

Hello, and thank you for your interest in **Giving Forward**. This PDF contains instructions for an initial technical assessment that will be used to determine your technical competency for the positions we have available.

While the work you are applying for is voluntary, we deemed this assessment was necessary to make sure that candidates are capable of tasks at hand. There is good potential for a large user-base in the future, thus competency and the ability to learn efficiently are vital to keeping the components scalable, while ensuring that current / in-progress implementations won't crack under unexpected pressure.

The other reason we ask for an assessment is that in addition to our core platforms, we often have additional side projects that will allow us to deliver more social impact more quickly, accelerate our marketing and growth or streamline our operations. We anticipate creating more than one volunteer SCRUM team as soon as the primary core SCRUM team is stable. Therefore, if you don't end up selected for the core SCRUM team, not to worry you may be able to provide a greater impact on one of the sister teams.

This assessment will primarily aims to address your competency as a **software engineer**, focusing on your ability to work with external libraries as well as utilize data structures and sorting algorithms. The ideal candidate will have a strong background in software development outside of the web-stack. You don't necessarily need to know all of our stack, but the ability to adapt quickly and pick up new languages is encouraged. Our stack outside of the website currently consists of Javascript and Golang, but as we are in the early stages of development we are willing to consider other languages as a group on a case-by-case basis. Our backend components may be Golang-driven based on the competency of the team we assemble, which you would adapt to, unless we decide to use another language.

Keep in mind we are not only looking for *correct* solutions, this assessment is also used to determine your competency in implementing tasks on a high-level. Best practices / approaches are encouraged.

*Enough about us, let's get into some code.*

***Disclaimer:** The problem provided below is completely fictional.*

---

**The problem:** Dax is a GivingForward programmer. We gave him the task of building an ingestion system for handling disorganized bundles (arrays) of ad-related information. He thought he would improve security by encrypting all of the data in the bundles. He wasn't very familiar with asymmetric encryption, so he decided he would just transmit the password and

**Confidential - Do not duplicate or distribute without written consent from Giving Forward**

obfuscate it! What an *awful idea!* Regardless, you were tasked with making the backend system that accepts this data work as intended, and provide us with the original data.

Your software/function will have two inputs:

- a) An array of **AES-128** encrypted strings (in base64 encoded format)
  - i) The first 16 bits of each encrypted string is the IV
- b) The obfuscated 16-bit **AES** key

The data inside each array in the decrypted string is a unique "data-type" that has one of the following constraints, with examples provided below each.

**Decryption can be checked against:** <https://www.devglan.com/online-tools/aes-encryption-decryption>

- ❖ It is a phone number (formats below)
  - **(555) 555-5555**
  - **555-555-5555**
- ❖ It is a URL (may or may not have protocol identifier) (only .com, .net and .org TLDs)
  - **https://thisexample.com/newssource/november/this-article?r=new**
  - **wesponsoryou.net/newsArticle.php?id=1596**
- ❖ It is an integer in some format (decimal, hex, or binary). You must convert it to decimal format if it is hex or binary. All integers should be returned in decimal format, as a string.
  - **"101000"** ( output would be **"40"** )
  - **"450"** ( output would be **"450"** )
  - **"0x7f"** (output would be **"127"** )

You will return one output: a string array of the organized data, following these constraints:

- ❖ All integers are first in the output array, organized from low to high, store them as their string equivalent
- ❖ Phones numbers come next; feel free to store these in any order, including lexicographically, or no order at all.

- ❖ URLs come last, sorted by shortest to longest length.

The obfuscated password format conforms to the following format:

- ❖ In the original string, Dax decided that every 3rd character would be appended to the front of the string, in the order that they appear in the string.
- ❖ If he ran out of characters to append, he would stop and return the string.

**Confidential - Do not duplicate or distribute without written consent from Giving Foward**

Example obfuscated password: **cxkahHvQZc4TajwX**

Intermediate (the string of chars removed + resulting): **cxkah + HvQZc4TajwX**

Unobfuscated (the password you would use to decrypt): **HvcQZxc4kTaajwhX**

### Full example input (golang syntax)

```
var encryptedArray = []string{
    "kh4pOGsXY1cWiLsePjrpdbS9ZED7NKbzBiQ8GX9VrBg=",
    "sRzq2do3ANdxLaDvDO31FoHYPNnCBemJiwzxgrkhY7k=",
    "BUm/oY3Pi7yh2igwLTjSOp7B8rOQn68XMeGalLAUFeJs6TNktPqJRHnCKxssxPisMzBFz+DZteeJNF2xSIBor45RxS3mlnj9qkAd96mTHI
Zs=",
    "YyOH1fh1qrh38bnsabQXNsRw0hcMQzxmwkAOArX78tY=",
    "1ma/R67aAv5/a1RhKZfwhpFingaREa/lkGAV4PRXI/IZPLLQ3pGSoDGLz5xRRxZ+YhbTKelWd5RXGf4PVE/uOQ=="
}

var password string = "9zYgxeCUD6zZ3Jaa"
```

### Output of the above example (golang syntax)

```
var decryptedArray = []string{
    "138", "3152", "555-268-5725",
    "http://example.com/?r=gakjnv&utm_source=test",
    "https://cloudstuff.net/article/11-14-2019/news-article"
}
```

**SOLVE** Here is a pastebin of additional inputs: <https://pastebin.com/MahQAU7E>

You will provide the outputs of these inputs (in any format) in your submission as **Solutions.txt**. We will also test your code against hidden test cases.

## Submitting your assessment

Even if you do not complete the assessment, we do accept partial completion. Please explain your reason / roadblock for a partial completion.

**If you have any questions, please email [insert-my-giving-forward-email-here].**

**Confidential - Do not duplicate or distribute without written consent from Giving Forward**

You are allowed to submit your solution in the following languages:

- ❖ Python
- ❖ Golang \*
- ❖ Rust
- ❖ Elixir
- ❖ Javascript
- ❖ PHP \*\*
- ❖ C/C++
- ❖ C#
- ❖ Java

\* - As mentioned previously, we are definitely compelled by strong Golang candidates.

\*\* - Please have additional xp outside of PHP, and/or adapt to new languages quickly

To submit your assessment, please email it directly to **assessments@givingforward.org**. Your assessment should contain all relevant source code files as well as any other files you used as input, documentation, configuration, etc -- we should be able to run your software without hassle.

Please submit your assessment as **FirstInitialLastName\_gfpa.zip** ( ex John Doe would be **JDoe\_gfpa.zip** ) to our email at **assessments@givingforward.org**

**copyright 2019 Giving Forward**