# Job Desciption

**WHO YOU ARE:**

-Experienced engineer who can develop backend application of industrial standard quality, with security, maintainability, reliability design in consideration.

-Proficient in manage our live system infrastructure, measure and anticipate upcoming scaling challenges. Able to prepare, scale the system and alert the team accordingly.

- Reliable team player who is responsible for the on-schedule delivery of tasks.

**RESPONSIBILITIES:**

- Develop and deliver good quality backend applications for the platform.
- Responsible for overall performance and reliability of the system infrastructure.
- Design, implement and maintain the tools that automate building reliable and performant system. Drive efficiencies in systems and processes: capacity planning, configuration management, performance tuning, monitoring and issue root cause analysis.

**SKILL SETS:**

**General:**

- A Computer Science/ Computer Engineering Bachelor degree.
- 4+ years working experience in web backend system application development.

Proficient in backend development using Java/Golang.

- Experience in building large, scalable distributed systems with good understanding of microservices architecture and associated principles.
-Proficient in Linux (e.g Ubuntu, CentOS, etc) server system management.
- Experience with performance benchmarking, diagnostic tools, monitoring and alerting platforms (e.g AWS cloudwatch, Datadog, Prometheus, etc).
- Experience on managing large infrastructure clusters such as container orchestration on Kubernetes, or relevant cloud plaforms (e.g AWS EKS, etc)

**Bonus:**

- Team Lead experience in previous work.

-Experience on development/technical management on Microsoft Outlook resource calendar/Google Workspace resource calendar.

-DevOps Experience with automation tools. (e.g Ansible, etc)

-Management Experience on CI/CD workflow. (e.g. Jenkins etc)

**WHAT YOU CAN GET:**

Plenty of learning opportunities and exposure to challenges on real industry system management. Great experience of how a start-up works and grows, from both technical side and nontechnical side.

# Coding Task

## GALAXY MERCHANT TRADING GUIDE

**Problem Description**

You decided to give up on earth after the latest financial collapse left 99.99% of the earth's population with 0.01% of the wealth. Luckily, with the scant sum of money that is left in your account, you are able to afford to rent a spaceship, leave earth, and fly all over the galaxy to sell common metals and dirt (which apparently is worth a lot). Buying and selling over the galaxy requires you to convert numbers and units, and you decided to write a program to help you.The numbers used for intergalactic transactions follows similar convention to the roman numerals and you have painstakingly collected the appropriate translation between them. Roman numerals are based on seven symbols:

Symbol Value

I 1

V 5

X 10

L 50

C 100

D 500

M 1,000

Numbers are formed by combining symbols together and adding the values. For example, MMVI is 1000 + 1000 + 5 + 1 = 2006. Generally, symbols are placed in order of value, starting with the largest values. When smaller values precede larger values, the smaller values are subtracted from the larger values, and the result is added to the total. For example MCMXLIV = 1000 + (1000 − 100) + (50 − 10) + (5 − 1) = 1944.

The symbols "I", "X", "C", and "M" can be repeated three times in succession, but no more. (They may appear four times if the third and fourth are separated by a smaller value, such as XXXIX.)

"D", "L", and "V" can never be repeated.

"I" can be subtracted from "V" and "X" only. "X" can be subtracted from "L" and "C" only. "C" can be subtracted from "D" and "M" only. "V", "L", and "D" can never be subtracted.

Only one small-value symbol may be subtracted from any large-value symbol.

A number written in Arabic numerals can be broken into digits. For example, 1903 is composed of 1, 9, 0, and 3. To write the Roman numeral, each of the non-zero digits should be treated separately. In the above example, 1,000 = M, 900 = CM, and 3 = III. Therefore, 1903 = MCMIII.

-- Source: Wikipedia (http://en.wikipedia.org/wiki/Roman_numerals)

**Sample Input:**

---

glob is I

prok is V

pish is X

tegj is L

glob glob Silver is 34 Credits

glob prok Gold is 57800 Credits

pish pish Iron is 3910 Credits

how much is pish tegj glob glob ?

how many Credits is glob prok Silver ?

how many Credits is glob glob Gold ?

how many Credits is glob glob glob glob glob glob Gold ?

how many Credits is pish tegj glob Iron ?

Does pish tegj glob glob Iron has more Credits than glob glob Gold ?

Does glob glob Gold has less Credits than pish tegj glob glob Iron?

Is glob prok larger than pish pish?

Istegj glob glob smaller than glob prok?

how much wood could a woodchuck chuck if a woodchuck could chuck wood ?

**Sample Output:**

---

pish tegj glob glob is 42

glob prok Silver is 68 Credits

glob glob Gold is 28900 Credits

Requested number is in invalid format

pish tegj glob Iron is 8015.5 Credits

pish tegj glob glob Iron has less Credits than glob prok Gold

glob glob Gold has more Credits than pish tegj glob glob

glob prok is smaller than pish pish

tegj glob glob is larger than glob prok

I have no idea what you are talking about

****NOTE****: The above input/output SHOULD NOT be used to HARDCODE into your application for usage. The application is meant to read the plain text input with the format like the sample, parse the content, process them and output answers of the questions in input with format given by the sample output.

**Requirements**

• Input to your program consists of lines of text detailing your notes on the conversion between intergalactic units and roman numerals. You are expected to handle invalid queries appropriately.

• Basic components such as Input/Output processing and business logic core are essential. But please design your application as complete and modular as possible, following a standard software system requirements such as error handling, etc.

• Apply proper design pattern and follow object oriented design style to design your application.

• Well refactor the code to follow good code practices. (Some good relevant resources can be found here: http://en.wikipedia.org/wiki/Clean_Code)

• Unit/Integration tests covering all the scenarios are good to have to show proper testing skills.

• Implement your application using Java / Golang.

• Please document clearly about your system design solution (feel free to use professional diagrams if you think necessary), nontrivial assumptions as well as the clear instructions on compiling and invoking the program with program environment details. Proper server setup instructions are also essential if you develop a web application for presenting the results.

****NOTE**:**

• Please DO NOT copy the solution online. We have already collected a list of typical solutions online. If we found your work is exactly same with some of them, we may have to terminate the review process.

• This code challenge is mean to demonstrate your software engineering skills and practice so Please structure your project code clean and clearly.

**Submission**

• You are expected to submit your source code project with the README document containing the required content stated above.

• Create a repository on Github or Bitbucket and commit your work there. Please submit the public link of the repository containing the required source to us after you finished. We are willing to have a look on how you build the application step by step.