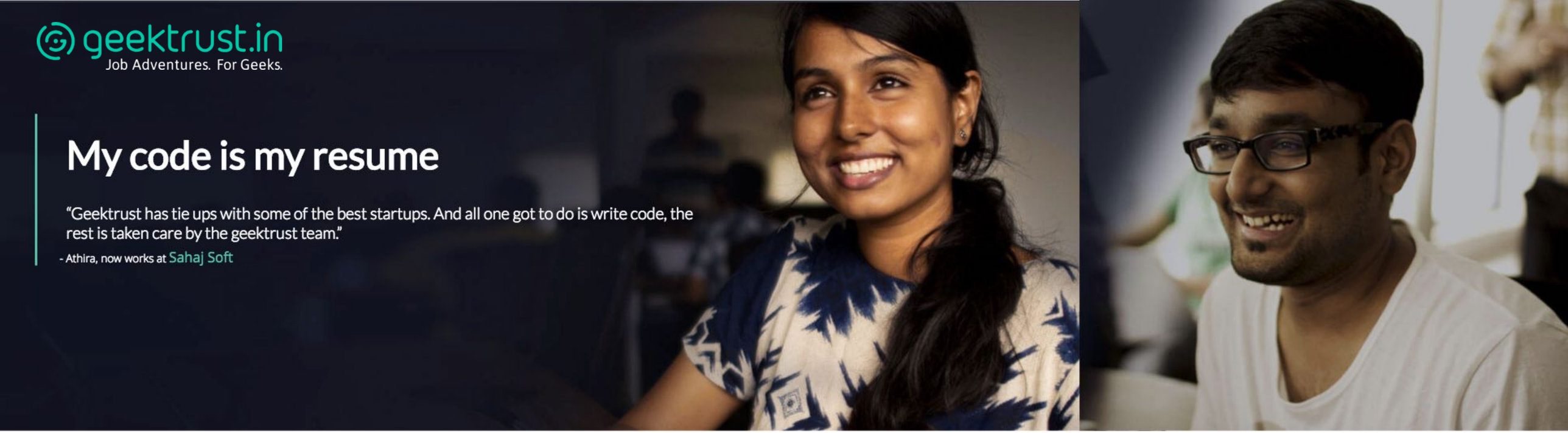


Athira (ex-Motorola), Souranil (ex-ThoughtWorks) and 200+ developers have solved Geektrust coding challenges to find great jobs over the last 4 years

- **Get priority** and be treated as a premium candidate to directly connect with decision makers at companies.
- **Get membership** and win an exclusive Geektrust DEVELOPER t-shirt.

Over 3000 developers from the best companies in the world have trusted us with their code. And we don't look just at the output, but how you get it is more important. We care about how well modelled your code is, how readable, extensible, well tested it is. Check out our coding help page to ensure you get a good score.

Getting started

1. Getting the output right is important, but more important is clean code and how well designed your code is. You should **absolutely** see our Help page post on what we look for in your code, and how to get started with the coding challenge.

2. See our evaluation parameters here and the badges to earn here.

3. We expect a command line app. So no web apps will be considered for evaluation. You don't need data stores either.

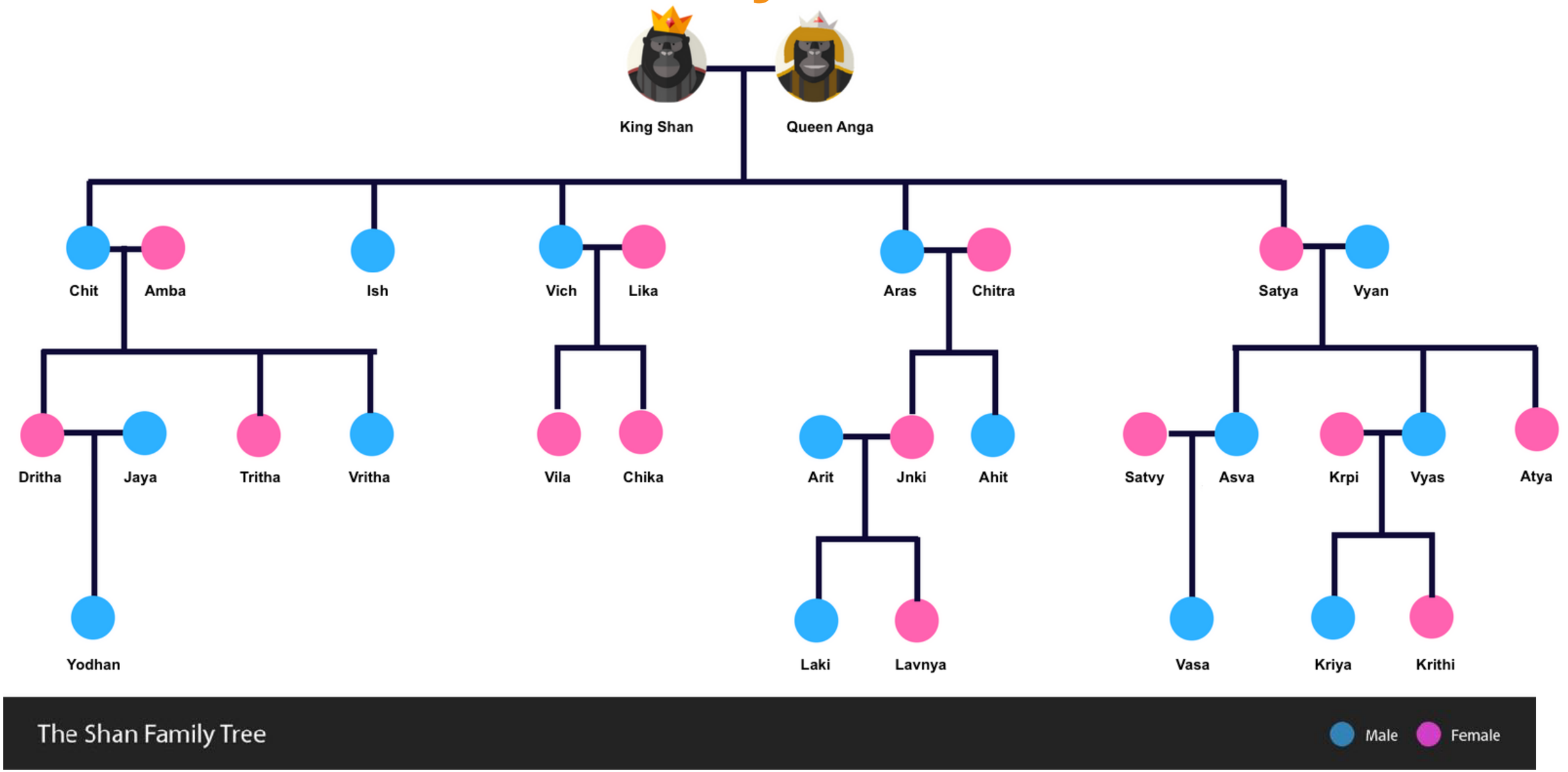
Problem Context

Our story is set in the planet of Lengaburu.....in the distant, distant galaxy of Tara B. And our protagonists are King Shan, Queen Anga & their family.

King Shan is the emperor of Lengaburu and has been ruling the planet for the last 350 years (they have long lives in Lengaburu, you see!). Let's write some code to get to know the family.

This coding problem is for backend and fullstack developers.

Family Tree



Meet The Family

Write code to model out the King Shan family tree so that:

- Given a ‘name’ and a ‘relationship’, you should output the people corresponding to the relationship in the order in which they were added to the family tree. Assume the names of the family members are unique.
- You should be able to add a child to any family in the tree through the mother.

Simple, right? But remember.. our evaluation is based not only on getting the right output, but on how you've written your code.

Relationships To Handle

There are many relations that could exist but at a minimum, your code needs to handle these relationships.

Relationships	Paternal-Uncle	Maternal-Uncle	Paternal-Aunt	Maternal-Aunt	Sister-In-Law	Brother-In-Law	Son	Daughter	Siblings
Definition	Father’s brothers	Mother’s brothers	Father’s sisters	Mother’s sisters	Spouse’s sisters, Wives of siblings	Spouse’s brothers, Husbands of siblings			

Sample Input/Output

Do initialise the existing family tree on program start. **Your program should take the location to the test file as parameter.** Input needs to be read from a text file, and output should be printed to the console. The test file will contain only commands to modify or verify the family tree.

- Input format to add a child:

```
ADD_CHILD  ”Mother’s-Name"  "Child's-Name"
"Gender"
```

- Input format to find the people belonging to a relationship:

```
GET_RELATIONSHIP  ”Name”  “Relationship”
```

- Output format on finding the relationship:

```
”Name 1”  “Name 2”… “Name N”
```

- Example test file:

```
ADD_CHILD Chitra  Aria  Female
GET_RELATIONSHIP  Lavnya  Maternal-Aunt
GET_RELATIONSHIP  Aria  Siblings
```

- Output on finding the relationship:

```
CHILD_ADDITION_SUCCEEDED
Aria
Jnki Ahit
```

More sample input output scenarios.

Please stick to the Sample input output format as shown. This is very important as we are automating the correctness of the solution to give you a faster evaluation. You can find some sample input output files here.

Sample 1

```
ADD_CHILD  Pjali Srutak Male
GET_RELATIONSHIP Pjali Son
```

Output on finding the relationship:

```
PERSON_NOT_FOUND
PERSON_NOT_FOUND
```

Pjali does not exist in the family tree

Sample 2

```
ADD_CHILD  Asva Vani Female
GET_RELATIONSHIP  Vasa Siblings
```

Output on finding the relationship:

```
CHILD_ADDITION_FAILED
NONE
```

Asva is male, hence child addition failed

Sample 3

```
GET_RELATIONSHIP  Atya Sister-In-Law
```

Output on finding the relationship:

```
Satvy Krpi
```

Go - Instructions to Build & Execute

Management of dependencies needs to be done via **Go Modules**. Do provide the **go.mod** file if you are using it. We use **go tool** for building Go applications. All your code should reside inside a package named **geektrust** under your Go workspace, which is typically the **GOPATH**. The name of the package should be geektrust and **NOT** be anything else.

Create your geektrust package path with the command.

```
mkdir $GOPATH/src/geektrust
```

Your directory structure should then look like this.

```
bin/
   geektrust                  # command executable
src/
   geektrust                  # main package
      main.go                 # start program file
      file_1.go               # another file required
      subpackage1             # a sub package you may write
         subpackage1.go       # a file under that sub package
```

We build and execute the the solution by using the following command from the directory **$GOPATH/src/geektrust**. The executable **geektrust** will be generated in the directory **$GOPATH/src/geektrust** besides the **main.go** file. Read more.

```
go build .
. /geektrust <absolute_path_to_input_file>
```

Supported Language & Versions

Code submissions are run against a Linux virtualized instance.

Supported language and versions are below:

Language	Supported versions	Supported Tools
C#	dotnet core 2.2, 3.1	dotnet
Go	1.12.x	Go build tool
Java	1.8, 1.11	maven, gradle
Node.js	8.16.x, 10.16.x, 12.6.x	npm, yarn
Python	3.7, 3.8	pip
Ruby	1.9.x, 2.2.x, 2.6.x	rake, bundler-rake

You can upload code in any version of Clojure, C++, Erlang, Groovy, Kotlin, PHP, Scala. We don't have automated tests for these languages yet. So your evaluation will take longer than the others.

Check list - submitting code

1. Please compress the file before upload. We accept .zip, .rar, .gz and .gzip
2. Name of the file should be the problem set name you are solving. For e.g. if you have solved Family problem, please name your file ‘Family.zip’.
3. Please upload only source files and do not include any libraries or executables or node_modules folder.
4. Usage of non-essential 3rd party libraries will affect your evaluation.
5. Add a **readme** with how to get your code working, and how to test your code.
6. Your solution will be downloaded & seen by companies you’re interested in. Hence we advise you to provide a solution that will work on any system without any code changes/manual setup.

what next?

A few good developers

Write great code. Get membership. Explore jobs.

Write Code

Sign up to solve interesting coding problems

Be a Member

Clear evaluation and get featured on GeekTrust

Connect with Companies

Explore opportunities as companies reach out to you

Find the Perfect Job

Review options, interview & find the right job for you