

Contact Tracing

A recent pandemic outbreak is spreading fast. CodeIT-Suisse is one of the very few countries where the number of infected cases is only a handful. The administration of CodeIT-Suisse wants to contain the spread of the outbreak within the country. As part of the containment plan, it wants to trace how each infected case in the country has contracted the infection.

A Health Organisation says that the infected can be traced back to the origin by analysing the genome patterns of the infected against the available infected genome data set of the pandemic.

When the virus mutates and spreads, its genome code can be altered by one or two characters in its genome string of three character instructions. The minimal the alterations between the two genome strings, the more likely that they both are related.

Also, if the genome string has more than one instruction with instruction's first character altered, the mutation is said to be non-silent. In case of non-silent mutation, the virus exhibits change in its effectiveness.

Can you write a program that could output the possible tracing paths of the infected and whether there is a non silent mutation of the virus in the trace?

Input and Output

Input

- You will be given a JSON string with Infected, Origin, and cluster of genomes information.

Output

- You should return a String array of possible trace paths from Infected to Origin.

Rules and Constraints

- Expose a post endpoint `/contact_trace` that takes JSON input and returns a string array of trace paths.
- Your output should contain all possible trace paths.
- All genome strings in the input are of same and fixed length.
- "name" field in the input json is unique.
- In case of non-silent mutation indicate it with an `*` suffixed to the name in the trace path

Trace String format

```
(genome-name)(* [if non-silent mutation])(white-space)->(genome-name)(* [if non-silent mutation])(white-space)
```

Sample Input/Output

Post Endpoint

`/contact_trace`

Input #1

```
{
  "infected": {
    "name": "orange",
    "genome": "acg-gcu-uca-gca-acu-ccc-gua-acg-gcu-uca-gca-acu-cac-gaa"
  },
  "origin": {
    "name": "turquoise",
    "genome": "acg-gcu-uca-gca-acu-ccc-gua-acg-gcu-uca-gca-acu-cac-gaa"
  },
  "cluster": [
    {
      "name": "magenta",
      "genome": "acg-gcu-uca-gca-acu-ccc-gua-acg-gcu-uca-gca-acu-cac-gaa"
    }
  ]
}
```

Output #1

```
[
  "orange -> magenta",
  "orange -> turquoise"
]
```

Explanation

- All genomes of infected, origin, and cluster ones in the above test case are all same which means the infected could be contacted the virus either from "magenta" or from "turquoise". Hence the output.

Input #2

```
{
  "infected": {
    "name": "plastic",
    "genome": "acg-gcu-uca-gca-acu-ccc-gua-acg-gcu-uca-gca-acu-cac-gaa"
  },
  "origin": {
    "name": "metal",
    "genome": "acg-acu-uca-gca-acu-ccc-gua-acg-ccu-uca-gca-acu-cac-gac"
  },
  "cluster": [
    {
      "name": "thread",
      "genome": "acg-acu-uca-gca-acu-ccc-gua-acg-ccu-uca-gca-acu-cac-gaa"
    }
  ]
}
```

Output #2

```
["plastic* -> thread -> metal"]
```

Explanation