A computer connected to the internet is identified by an IP address. The most common way of displaying an IP address is the dotted quad method: four eight-bit (0-255 in base ten) numbers separated by periods.

Someone has given you a possible IP address, but the periods have been removed, leaving only a string of digits. Write a class IPConverter with a method possibleAddresses that takes a String ambiguousIP containing the digits and returns a String[] containing all the possible IP addresses that can be formed from those digits by inserting three periods to form a dotted quad. Sort the elements of the return value lexicographically, using their string ordering (the period character precedes all digit characters).

The numbers in each of the four positions can have any integer value between zero and 255, inclusive. However, a number may not have leading zeroes. For example, the digits 1902426 can form 1.90.24.26, 19.0.24.26, 190.2.4.26, and other IP addresses (see Example 0). However, it cannot form 19.02.4.26.

Definition

- **Class:** IPConverter
- **Method:** possibleAddresses
- **Parameters:** String
- **Returns:** String[]
- **Method signature:**

  ```java
  String[] possibleAddresses(String ambiguousIP)
  ```

  (be sure your method is public)

Class

```java
public class IPConverter {
    public String[] possibleAddresses(String ambiguousIP){
        // you write code here
    }
}
```

Constraints

- ambiguousIP will contain between 0 and 50 characters, inclusive.
- Each character of ambiguousIP will be between '0' and '9', inclusive.

Examples

1. "1902426"
Returns:
{ "1.90.24.26",
 "1.90.242.6",
 "19.0.24.26",
 "19.0.242.6",
 "190.2.4.26",
 "190.2.42.6",
 "190.24.2.6" }

This is the example from the problem statement.

2.
"000"

Returns: { }

3.
"

Returns: { }

4.
"0186290"

Returns: { "0.18.62.90", "0.186.2.90", "0.186.29.0" }

5.
"11111111"

Returns:
{ "1.1.111.111",
 "1.11.111.111",
 "1.111.111.11",
 "1.1111.1.11",
 "1.11111.11",
 "1.111111.1",
 "11.1.111.11",
 "11.11.111.1",
 "11.111.111.1",
 "11.1111.11",
 "11.11111.1",
 "11.111111.1"
 }

6.
"3082390871771742784899852251737850570843857369760"

Returns: { }

7. "256255255"
Returns: { "2.56.255.255", "25.6.255.255", "25.62.55.255" }
**Problem Statement**

I want a list of BioServices that are associated with each KindOfInput (such as "gene", "DNAFragment", "genome", etc.). What is available from my service provider is a list of strings, each containing the name of a service followed by all the KindsOfInput it requires.

Given a `String[]` services, return a `String[]` in which each element contains a KindOfInput followed by the names of all the services that use that kind of input.

Each KindOfInput should appear exactly once in the return. Within each element of the return, the service names should be in alphabetical order, and should be separated by the 2 characters ", " (comma space). The KindOfInput should be separated from the first service name by the 5 characters " ==> " (space eq eq gt space). The returned list should be in alphabetical order.

Note that all names are case-sensitive, and that "alphabetical order" refers to the ASCII ordering, where, for example, 'Z' precedes 'a'.

**Definition**

- **Class:** ServiceNames
- **Method:** makeList
- **Parameters:** `String[]` services
- **Returns:** `String[]`
- **Method signature:**
  
  ```java
  public String[] makeList(String[] services)
  ```

  (be sure your method is public)

**Class**

```java
public class ServiceNames {
    public String[] makeList(String[] services)
    {
        // fill in code here
    }
}
```

**Constraints**

- `services` will contain between 1 and 50 elements, inclusive.
- Each element of services will contain between 1 and 50 characters, inclusive.
- Each element of services will contain tokens separated by a single space (' ').
- Each token will consist of 1 or more letters ('A'-'Z' or 'a'-'z').
- The first tokens of the elements in services will be distinct.
- Within each element of services the KindsOfInput will be distinct.

Examples

1. "BLAST Genome Annotation Sensitivity", "PING", "X Annotation"
   Returns: {"Annotation ==> BLAST, X", "Genome ==> BLAST", "Sensitivity ==> BLAST" }

2. "PING"
   Returns: { }

   There are no KindsOfInput so the return has 0 elements.

3. "BLAST Genome annotation Sensitivity", "PING", "X Annotation", "Apple X ample"
   Returns:
   {"Annotation ==> X",
    "Genome ==> BLAST",
    "Sensitivity ==> BLAST",
    "X ==> Apple",
    "ample ==> Apple",
    "annotation ==> BLAST" }

   annotation and Annotation are distinct kinds of input. annotation comes later alphabetically than
   any name that starts with an uppercase letter.

---

Owen L. Astrachan
Last modified: Thu Nov 16 12:15:36 EST 2006
SpreadingNews APT

Problem Statement

You are the manager of a company, and you want all of your employees to be notified of an important news item as quickly as possible. Your company is organized in a tree-like structure: each employee has exactly one direct supervisor, no employee is his own direct or indirect supervisor, and every employee is your direct or indirect subordinate. You will make a phone call to each of your direct subordinates, one at a time. After hearing the news, each subordinate must notify each of his direct subordinates, one at a time. The process continues this way until everyone has heard the news. Each person may only call direct subordinates, and each phone call takes exactly one minute. Note that there may be multiple phone calls taking place simultaneously. Return the minimum amount of time, in minutes, required for this process to be completed.

Employees will be numbered starting from 1, while you will be numbered 0. Furthermore, every supervisor is numbered lower than his or her direct subordinates. You are given an int[] supervisors, the ith element of which is the direct supervisor of employee i. The first element of supervisors will be -1, since the manager has no supervisors.

Definition

- Class: SpreadingNews
- Method: minTime
- Parameters: int[]
- Returns: int
- Method signature:

  public int minTime(int[] supervisors)

  (be sure your method is public)

Constraints

- supervisors will contain between 1 and 50 elements, inclusive.
- Element i of supervisors will be between 0 and i-1, inclusive, except for the first element which will be -1.

Examples

1.

{-1, 0, 0}

Returns: 2

Call subordinate 1 at time 0 and then subordinate 2 at time 1. By time 2, both subordinates will have heard the news.
2. 

\{-1, 0, 0, 2, 2\}

Returns: 3

This time call employee 2 first, and then employee 1 at time 1. After hearing the news, employee 2 will call employee 3 at time 1 and employee 4 at time 2. It takes 3 minutes for everybody to hear the news.

3. 

\{-1, 0, 1, 2, 3\}

Returns: 4

Everyone in the company has only one subordinate, resulting in a chain of phone calls.

4. 

\{-1, 0, 0, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 5, 5, 5, 6, 7, 7, 8, 8, 12, 13, 14, 16, 16, 16\}

Returns: 7

Owen L. Astrachan

Last modified: Tue Nov 28 13:30:49 EST 2006