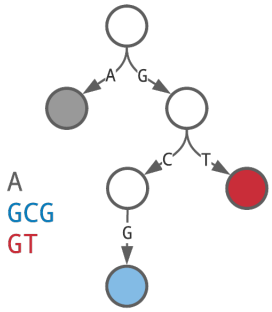
9A Construct a Trie from a Collection of Patterns

Trie Construction Problem
Construct a trie from a set of patterns.

Input: A collection of strings *Patterns*.
Output: TRIE(*Patterns*).



Formatting

Input: A space-separated list of strings *Patterns*.
Output: Each edge of TRIE(*Patterns*) will be newline-separated and encoded by a triple: the first two members of the triple must be the integers labeling the initial and terminal nodes of the edge, respectively; the third member of the triple must be the symbol labeling the edge.

Constraints

- The number of patterns in the string-set *Patterns* will be between 1 and 10^3.
- The length of any one pattern in *Patterns* will be between 1 and 10^3.
- No pattern is a prefix of another pattern.

Test Cases

Case 1

Description: The sample dataset is not actually run on your code.

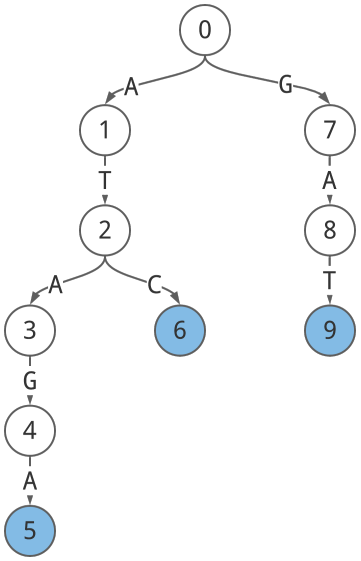
Input:

```
ATAGA ATC GAT
```

Output:

```
0 1 A
0 7 G
1 2 T
2 3 A
2 6 C
3 4 G
4 5 A
7 8 A
8 9 T
```

Figure:



Shown above is the trie containing the words `ATAGC`, `ATC`, and `GAT`. These words are outlined by the paths from the root node (labeled 0) to the leaf nodes (labeled 5, 6, and 9, colored blue).

Case 2

Description: No two patterns share the same prefix.

Input:

```
ATCG TCGA CGAT
```

Output:

```
0 1 A
0 5 T
0 9 C
1 2 T
2 3 G
3 4 C
5 6 C
6 7 G
7 8 A
9 10 G
10 11 A
11 12 T
```

Case 3

Description: All patterns share a prefix, but have distinct suffixes.

Input:

```
GAGC GAGA GAGT
```

Output:

```
0 1 G
1 2 A
2 3 G
3 4 C
3 5 A
3 6 T
```

Case 4

Description: Patterns have common prefixes and suffixes.

Input:

```
ATAGC ATGGC
```

Output:

```
0 1 A
1 2 T
2 3 A
3 4 G
4 5 C
2 6 G
6 7 G
7 8 C
```

Case 5

Description: Patterns comprised of repeats or palindromes.

Input:

```
ATA AGGA
```

Output:

```
0 1 A
1 2 T
2 3 A
1 4 G
4 5 G
5 6 A
```

Case 6

Description: A larger dataset of the same size as that provided by the randomized autograder. Check input/output folders for this dataset.