COMPLETE STRING

A string is called a complete string if every prefix of this string is also present in array 'A'. Ninja is challenged to find the longest complete string in the array 'A'. If there are multiple strings with the same length, return the lexicographically smallest one & if no string exists, return "None".

Ex:

I/p → n ni nin ninj ninja ninga ninga   o/p → ninja

```
Class Trie {
    Node * root;
Public:
    Trie() {
        root = new Node();
    }

    void insert (string word)
    {
        Node * node = root;
        for (int i=0; i< word.size (); i++)
        {
            if (! node→ contain (word[i]))
            {
                node →put (word[i], new Node());
            }
            node = node → get (word[i]);
        }
        node → setEnd();
    }
```

```
    bool Represent (string word)
    {
        Node * node = root;
        for (i=0 to i< word.size ())
        {
            if (node →
            {
                node = node → get (word[i]);
            }
            if (! node → contain (word[i]) ) return false;
            if (! node →isEnd()) return false;
        }
        return node → is End();
    }
};
```

```
Trie {
    arr[26];
    bool flag;
}
```

```cpp
String computeString ( int n, vector <string> & a)
{
    String ans = "";

    String ans = "";                                    P.( } → O(n len(word))
    Trie * t = new Trie();                                          avg. length

    for ( int i=0; i< a.size(); i++)
    {
        t → insert ( a[i] );
    }

    for ( int i=0; i<n; i++)
    {
        if ( t → isPresent ( a[i] ) )
        {
            if ( ans.size () < a[i].size ()) ans = a[i];
            else if ( ans.size () == a[i].size() && ans > a[i] ) ans = a[i];
        }
    }

    return ans.size() != 0 ? ans : "None";
}
```