Starting with an empty list, add cities one at a time, maintaining the list in alphabetical order

```
Enter city name ('quit' to stop): Denver
Denver;
Enter city name ('quit' to stop): Chicago
Chicago;Denver;
Enter city name ('quit' to stop): New York
Chicago;Denver;New York;
Enter city name ('quit' to stop): Houston
Chicago;Denver;Houston;New York;
Enter city name ('quit' to stop): Not-legit
City name must have letters and spaces ONLY ('quit' to stop): Chicago
Chicago;Denver;Houston;New York;
Enter city name ('quit' to stop): quit
```

Algorithm to insert new city in the correct spot

```
Chicago;Denver;New York;
Enter city name ('quit' to stop): Houston
```

Scan the cities list and find the city BEFORE which Houston will go

```
                1111111111222222
0123456789012345678901234
Chicago;Denver;New York;
```

start *pos*

Scanning uses two index variables, *start* and *pos*, to look at each city in the cities string

From the *start* index, find where the next `;` is – this will be the *pos* index

The first time around, Chicago is the city marked off by *start..pos-1*

Checking the new city, Houston, against Chicago, we find that Houston is NOT "less than" Chicago, i.e. it will NOT appear BEFORE Chicago in alphabetical order, so on to the next city in the list

Algorithm to insert new city in the correct spot

```
Chicago;Denver;New York;
Enter city name ('quit' to stop): Houston
```

Scan the cities list and find the city BEFORE which Houston will go

```
                1111111111222222
0123456789012345678901234
Chicago;Denver;New York;
        ↑       ↑
      start    pos
```

Checking the new city, Houston, against Denver, we find that Houston is NOT "less than" Denver, i.e. it will NOT appear BEFORE Denver in alphabetical order, so on to the next city in the list

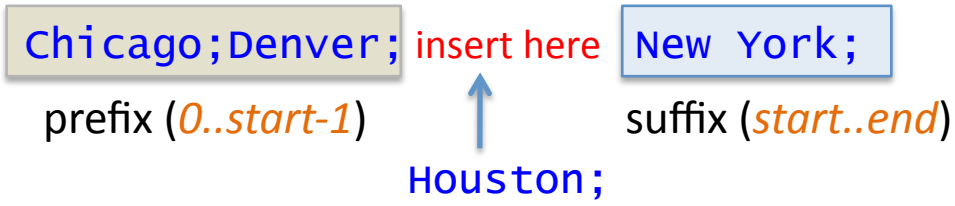
Algorithm to insert new city in the correct spot

```
Chicago;Denver;New York;
Enter city name ('quit' to stop): Houston
```

Scan the cities list and find the city BEFORE which Houston will go

```
          11111111112222
012345678901234567890123
Chicago;Denver;New York;
                ↑       ↑
              start    pos
```

Checking with the next city, Houston is found to be "less than" New York (i.e. it comes BEFORE New York), so Houston is inserted before New York



Special Cases

1. Initially cities list is empty

```
Enter city name ('quit' to stop): Denver
Denver;
```

2. New city is before the first city in list

```
Enter city name ('quit' to stop): Chicago
Chicago;Denver;
```

3. New city is after last city in list

```
Enter city name ('quit' to stop): New York
Chicago;Denver;New York;
```

4. New city is already in the list

```
Chicago;Denver;Houston;New York;
Enter city name ('quit' to stop): Chicago
Chicago;Denver;Houston;New York;
```