Problem 6—Great Circles

It is a remarkable coincidence that every planet in the Republic is a perfect sphere. Furthermore, every culture developed the terms “North Pole,” “South Pole,” “Equator,” and “Prime Meridian” just as humans did. Every culture also measures angles in degrees and distances in kilometers (“parsecs,” of course, being a measure of time). Also, although the planets are of different sizes, every culture has a latitude and longitude system just as Earth does: latitude is measured in degrees north and south of the equator, up to 90; longitude is measured in degrees east and west of the prime meridian, up to 180.

Every culture has an airline system, and these airlines want to fly the shortest possible route connecting points on their planet. It turns out that the shortest route between points on a sphere is called the Great Circle Route. A great circle is a circle on the sphere whose center is also the center of the sphere. The shortest distance between two points on a sphere is along a great circle containing both of those points. Computing this distance isn't too difficult either. Here's one way: first you draw a triangle (with straight lines) whose vertices are the two given points and the center of the planet. You then find the lengths of these sides: two of the sides have length equal to the radius of the planet, and the length of the third side is the direct (underground) distance between the two given points. Since you have all sides of the triangle, you can then compute the angle at the center of the planet, and with this piece of information and the radius of the planet, you can then compute the length of the arc of the great circle that connects these two points.

Using this or another method, compute great circle distances for various points on various planets in the Republic.

**INPUT SPECIFICATION.** The input file will consist of several lines. Each line (but the last) will contain a decimal integer inclusively between 1000 and 100000 indicating the circumference of the planet (in km), one space, one planetary point, one space, a second planetary point, **<EOLN>**. A planetary point consists of this information: a decimal integer inclusively between 0 and 90 indicating the latitude, one space, “N” or “S” indicating the direction of latitude, one space, a decimal integer inclusively between 0 and 180 indicating the longitude, one space, “E” or “W” indicating the direction of longitude. The last line will simply be 100**<EOLN>**, and indicates the end of input.

**OUTPUT SPECIFICATION.** The output cases are to appear in the same order in which they appear in the input file. Each output case should be of the form: “Case c: Cities are d km apart.” c and d are decimal integers; c is the number of the case being processed (starting with 1) and d is the distance in km between the two points, rounded to the nearest integer. Each line should be terminated by exactly one **<EOLN>**. Remember: format counts. Be sure your output is formatted *exactly* as demonstrated below.

**SAMPLE INPUT.**

```
40000·90·N·23·E·90·S·38·W<EOLN>
25000·46·N·87·W·40·N·90·W<EOLN>
100<EOLN>
<EOF>
```

**SAMPLE OUTPUT.**

```
Case·1:··Cities·are·20000·km·apart.<EOLN>
Case·2:··Cities·are·444·km·apart.<EOLN>
<EOF>
```