D. Biker's Trip Odometer

Source file: **biker.{c, java, cpp}**

Input file: **{stdin, System.in, cin}**

Output: **{stdout, System.out, cout}**

Most bicycle speedometers work by using a *Hall Effect* sensor fastened to the front fork of the bicycle. A magnet is attached to one of the spokes on the front wheel so that it will line up with the *Hall Effect* switch once per revolution of the wheel. The speedometer monitors the sensor to count wheel revolutions. If the diameter of the wheel is known, the distance traveled can be easily calculated if you know how many revolutions the wheel has made. In addition, fi the time it takes to complete the revolutions is known, the average speed can also be calculated.

For this problem, you will write a program to determine the total distance traveled (in miles) and the average speed (in Miles Per Hour) given the wheel diameter, the number of revolutions and the total time of the trip. You can assume that the front wheel never leaves the ground, and there is no slipping or skidding.

Input

In put consists of multiple data sets, one per line, of the form:

```
diameter revolutions time
```

The *diameter* is expressed in inches as a floating point value. The *revolutions* is an integer value. The *time* is expressed in seconds as a floating point value. Input ends when the value of *revolutions* is **0** (zero).

Output

For each data set, print:

```
Trip #N: distance MPH
```

Of course, *N* should be replaced by the data set number, *distance* by the total distance in miles (accurate to two decimal places) and *MPH* by the speed in miles per hour (accurate to two decimal places). Your program should not generate any output for the ending case when *revolutions* is **0**.

Constants

Use the following values:

$\pi = 3.1415927$
one mile = 5,280 feet

Sample Input

```
26 1000 5
27.25 873234 3000
26 0 1000
```

Sample Output

```
Trip #1: 1.29 928.20
Trip #2: 1179.86 1415.84
```