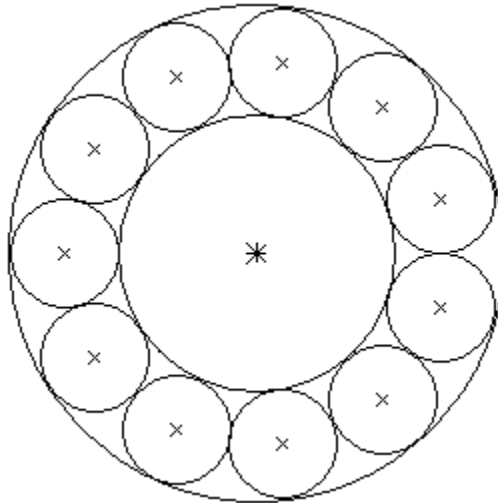


C • Arthur's Table

King Arthur is well known for holding that all knights are equal and for his round table.

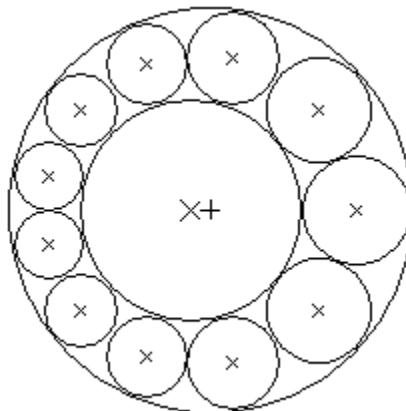
There is a central serving platter (did Merlin invent the *lazy-susan*?) and each knight has a circular trencher to eat from:



Unfortunately, Merlin accidentally called up a pre-shade of George Orwell and Arthur started muttering something like:

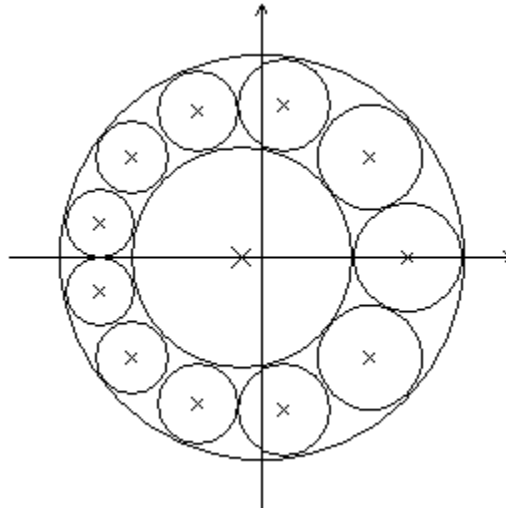
... but some knights are more equal than others...

So he asked Merlin to move the center platter away from him a bit so he and his closest aides could have a bit more room:



Merlin complained that all the trenchers and the serving platter would have to be remade. Arthur, like a good leader said "Deal with it."

Write a program to help Merlin make the new trenchers and serving platter. To describe the layout we will use a coordinate system with origin at the center of the **table**, with the **x** axis pointing to Arthur's seat and the **y** axis perpendicular:



Input

Input consists of a single line containing the diameter, D , ($8 \leq D \leq 30$) of the table (in kingly feet), the number, N , ($7 \leq N \leq 40$) of knights to be seated, and the offset, O , ($0.1 \leq O < \text{radius of an original trencher}$) from the center of the table to the center of the central serving platter (as a decimal number of kingly feet).

Output

The output consists five lines. The first output line gives the radius of the central platter (in kingly feet) to 3 decimal places with center at **(-offset, 0)**. (offset is O from the input.)

The next four lines consist of three space separated floating point values to three decimal places. These values represent the **x** coordinate of the center of a trencher, the **y** coordinate of the center of the trencher and the **radius** of the trencher (all in kingly feet) of Arthur's trencher and the next three trenchers counter-clockwise from Arthur's (one set of values for each line of output).

Sample 1:

Sample Input	Sample Output
10 11 0.5	2.761
	3.631 0.000 1.369
	2.703 2.513 1.310
	0.572 3.785 1.172
	-1.625 3.621 1.031