Find the shape with the largest area **using the OO approach**

One day, Mike goes and visits his cousin John. John has a set of toys with three different shapes, i.e., Triangle, Rectangle, and Circle. John picks up three pieces of toys and is curious about which piece is the largest. Now, your task is, as the role of Mike, to find out the answer with a program.

A set of shapes are firstly captured into the program in the order of Triangle, Rectangle, and Circle. For each kind of shape, a value N representing the number of toys in this particular shape is inputted, followed by the shape information on the next N lines. The information contained in each shape are as follows:

* Triangle: three values (a, b, c) represent the lengths of three sides of a triangle.
* Rectangle: two values (width, height) represent the lengths of two sides of a rectangle.
* Circle: a value (radius) represents the radius of a circle.

The input of shape information may contain mistakes by carelessness. Your program should check the validity of the shapes, and the area of any invalid shape is regarded as 0.

Then, three integer values (from 1 to N) are captured to select one toy with Triangle, Rectangle, and Circle shapes by the input order of the shape information, respectively. To simplify your coding effort, this question assumes that the input of at least one piece of picked-up shape is valid. Print the shape information and area of the shape with the largest area among the three selected ones.

Object orientation (OO) requirements:

* Your solution must use three classes to represent Triangle, Rectangle, and Circle.
* Individual instances of these shapes must be objects of the corresponding classes.
* All shape information should be captured as data members of the corresponding objects.
* The area of each object must be computed within the member function of the corresponding class.
* All console output (cout) statements that display the shape and area information about an object must be located within member functions of the class.
* To simplify your coding effort, assignment to values of data members of a class can be written outside the class (e.g., in the main() function).

Note:

1. You may refer to [Heron’s formula](https://en.wikipedia.org/wiki/Heron%27s_formula) for the computation of the area of a triangle.
2. **You MUST use the object-oriented approach to developing your solution.**
3. You should use **the M\_PI definition in <cmath>** when computing the area of a Circle object. In your .cpp file (in Visual Studio), define macro \_USE\_MATH\_DEFINES as the first line before any “#include” statement and include <cmath>. M\_PI is the pi value.
   1. #define \_USE\_MATH\_DEFINES

#include <cmath>

cout << M\_PI << endl; // example use

1. Your printed message should format the output with 2 decimal places, but the comparison of area should consider all digit places (your solution can assume that no two shapes have the same areas. ).

**Input is underlined.**

|  |
| --- |
| Sample 1 |
| Enter number of Triangles: 3  3 4 5  4 5 6  1 1 3  Enter number of Rectangles: 2  2 3  1 7  Enter number of Circles: 2  3  1  Enter three indexes for picking: 1 1 1  The largest shape is:  -- Circle --  radius = 3.00  area = 28.26 |
| Sample 2 |
| Enter number of Triangles: 1  1.0 2.2 3.1  Enter number of Rectangles: 4  1.1 2.2  3.1 3.2  0.1 -0.1  1.1 3.3  Enter number of Circles: 3  -2  2.5  3  Enter three indexes for picking: 1 3 1  The largest shape is:  -- Triangle --  a = 1.00  b = 2.20  c = 3.10  area = 0.57 |
| Sample 3 |
| Enter number of Triangles: 1  1.1 2.2 3.0  Enter number of Rectangles: 1  3 4  Enter number of Circles: 1  -1  Enter three indexes for each shape: 1 1 1  The largest shape is:  -- Rectangle --  width = 3.00  height = 4.00  area = 12.00 |