Laboratory 3

For weeks starting October 11 and October 18.

In teams of two, write two different versions of a package of programs that will store information about a circle without revealing how that information is stored internally. The two versions must use different internal representations. The interface of such a package is specified below.

As a separate program, write a user interface program that allows users to input data defining a circle and to inquire about characteristics of that circle. For example, a user may wish to input the coordinates of the centre and two points, the centre and the radius, or the coordinates of the centre and the area of the circle. The user interface program must accomplish its task by using the programs provided in package described below.

Test both packages and the user interface program until you are satisfied that they are correct, then combine them to produce *two* systems that can be demonstrated. Completed this *by the end of the first week.*

In the second week, you will be asked to exchange your packages with another group, to test the other group's packages, and then combine your interface program with both of the packages that you got from the other group. When writing, testing, and demonstrating these programs, log all errors and write a final lab report - due one week after the lab is complete. Your report should analyse the difficulties that you encountered and explain how they could have been avoided.

Keep a log of all major decisions and all errors discovered. This log will become a part of your laboratory report. If you consult with anyone, summarise that conversation in your log.

The package interface is defined as follows:

1. SETTOL(epsilon) - see previous assignments. The programs below will use the value of eps as the tolerance in determining equality of real numbers.

2. S3P(x_{A0}, y_{A0}, x_{A1}, y_{A1}, x_{A2}, y_{A2}); all arguments are real; return value is integer. Return an integer (code) in accordance with the following table.

	normal	twosame	¬ (normal ∨ twosame ∨ singlepoint)	singlepoint
return	1	-1	-2	0

If $0 \le code \le 1$ store data corresponding to a circle with the centre at (x_{A0}, y_{A0}) and
radius = dist(x_{A1}, y_{A1}, x_{A0}, y_{A0}). For all other codes store data values indicating that no circle is stored.

3. SCR(x_{A0}, y_{A0}, r)**;** arguments are real; return value is integer specified by the table below:

	r > 0	r = 0	r < 0
return	1	0	-1

If $0 \le code \le 1$ store data corresponding to a circle with the centre at (x_{A0}, y_{A0}) and radius = r. For all other codes store data values that indicate that no circle is stored.

4. SCA(x_{A0}, y_{A0}, a)**;** arguments are real; return value is integer

Return an integer code in accordance with the table below:

	a > 0	a = 0	a < 0
return	1	0	-1

If $0 \le code \le 1$ store data corresponding to a circle with the centre at (x_{A0}, y_{A0}) and area = a. For all other codes store data values that indicate that no circle is stored.

5. GA(a); a is a real valued variable.

If no circle is stored, return -1 otherwise return 1.

If a circle is stored, return the area of the circle as the value of a.

6. GR(r); r is a real valued variable.

If no circle is stored, return -1 otherwise return 1.

If a circle is stored, return the length of the radius as the value of r.

7. GC(x, y); x and y are real.

If no circle is stored, return -1 otherwise return 1.

If a circle is stored, return the coordinates of the centre as **(**x, y).

8. INCIRC**(**x, y); x and y are real.

If no circle is stored, return -1 otherwise return 1 if (x, y) is inside the circle, 2 if (x, y) is on the circumference, 0 if the point is outside the circle.

Definitions:

Since we are working with real numbers, we must define "=" to allow for numerical round-off.

$a = b \equiv |a-b| \le eps$, where eps should be a small positive value (a parameter). Note that this version of equality is reflexive, symmetric, but not transitive.

Def: $dist(x_1, y_1, x_2, y_2) \equiv sqrt((x_1 - x_2)^2 + (y_1 - y_2)^2)$

Def: $normal \equiv (dist(x_{A1}, y_{A1}, x_{A0}, y_{A0}) = dist(x_{A2}, y_{A2}, x_{A0}, y_{A0}) \wedge \neg (twosame \vee singlepoint)$

Def: $twosame \equiv ((x_{A0}, y_{A0}) = (x_{A1}, y_{A1}) \vee (x_{A0}, y_{A0}) = (x_{A2}, y_{A2}) \vee (x_{A1}, y_{A1}) = (x_{A2}, y_{A2}))) \wedge (\neg (singlepoint))$

Def: $singlepoint \equiv ((x_{A2}, y_{A2}) = (x_{A0}, y_{A0}) = (x_{A1}, y_{A1}))$

Please note that every package must include all 8 of the programs. Every package offers 1 program to set the tolerance, 3 programs to input the dimensions, and 4 programs to return circle characteristics. A user should not be able to tell which program was used to enter the circle dimensions by calling the "G" programs.