Circles (1) P84786_en

To solve this exercise you will need the definition of @Point@ and @distance()@ of problem P46254.

Write a procedure

void *move*(*Point*& *p1*, **const** *Point*& *p2*);

that moves the point @p1@ according to the coordinates indicated by the point @p2@.

For instance, being @p1@ the point $(2, 1)$, and @p2@ the point $(-0.5, 4)$. Then @move(p1, p2)@ would do that @p1@ was $(1.5, 5)$.

Additionally, using the definition

```
struct Circle {
    Point center;
    double radius;
};
```

write two procedures,

void *scale*(*Circle*& *c*, **double** *sca*);

that scales the circle @c@ proportionately to the real strictly positive @sca@, and

void *move*(*Circle*& *c*, **const** *Point*& *p*);

that moves the circle @c@ according to the coordinates indicated by @p@.

For instance, being @c@ a circle of center $(1, 2)$ and radius 3. Then, @scale(c, 2)@ would obtain a circle of center $(1, 2)$ and radius 6. However, if @p@ is $(3.5, -1)$, @move(c, p)@ would obtain a circle of center $(4.5, 1)$ and radius 3.

Write also a function that prints if a point @p@ is inside a circle @c@:

bool *is_inside*(**const** *Point*& *p*, **const** *Circle*& *c*);

Suppose that the radii are always strictly positive, and that @p@ will never be exactly in the border of @c@.

Observation

You only need to submit the required classes; your main program will be ignored. Strictly obey the type definitions of the statement.

Problem information

Author: Salvador Roura
Translator: Carlos Molina

Generation: 2026-01-25T12:04:09.507Z