bool empty_set(Rect R, Disk D)

Description:
The function detects the intersection of a rectangle and a circle.
Input parameters:
The function receives two parameters:

- **R** is the rectangle, the element of class **Rect** with characteristics:
  - **x0, y0** are coordinates of the bottom left corner;
  - **x1, y1** are coordinates of the top right corner.

  To access these characteristics using the methods **get_x0()**, **get_y0()**, **get_x1()**, **get_y1()**, implemented in the class **Rect**.
- **D** is the circle, the element of class **Disk** with characteristics:
  - **c1, c2** is the center of the circle;
  - **r** is the radius of the circle.

  To access these characteristics using the methods **get_c1()**, **get_c2()**, **get_r()**, implemented in the class **Disk**.

Output parameters:

- The function returns the boolean value of the parameter **empty**.
  - $empty = true$ the rectangle and circle do not intersect;
  - $empty = false$ the rectangle and circle have an intersection.

Algorithm
First of all, we check the condition: the center of the circle is inside the rectangle.
If this condition is satisfied, the figure intersect.
Else we consider four half-planes formed by the rectangle:

- left half-plane ($c1 < x0$);
- right half-plane ($c1 > x1$);
- top half-plane ($c2 > y1$);
- bottom half-plane ($c2 < y0$).

In the equation of the circle, we substitute the equations of the boundaries of each half-plane and find the discriminants of the obtained square equations.
A circle does not intersect with a straight line if the equation has no solutions, i.e. the discriminant is negative.
We check the intersection of the circle with the half-plane boundary. If there is no intersection, then the parameter $empty = true$. In case there is an intersection, we also check the intersection of the circle with the straight lines passing through the sides of the rectangle orthogonal to the half-plane boundary. In the case of an intersection, the parameter $empty = false$.

Rect intersection (Rect R, Disk D)

Description:
The function approximates the area of intersection of the rectangle and the circle with orthogonal lines. Based on the intersection points of these lines, we construct a rectangle with a minimum area, which contains the intersection area of the rectangle and the circle.
Input parameters:
The function receives two parameters:

- **R** is the rectangle, the element of class **Rect** with characteristics:
  - **x0, y0** are coordinates of the bottom left corner;

- **x1, y1** are coordinates of the top right corner.

To access these characteristics using the methods **get_x0()**, **get_y0()**, **get_x1()**, **get_y1()**, implemented in the class **Rect**.

- **D** is the circle, the element of class **Disk** with characteristics:
  - **c1, c2** is the center of the circle;
  - **r** is the radius of the circle.

  To access these characteristics using the methods **get_c1()**, **get_c2()**, **get_r()**, implemented in the class **Disk**.

Note: The rectangle and the circle must intersect.
Output parameters:

- The function returns new rectangle **newR** (the element of class **Rect**) with a minimum area, which contains the intersection area of the rectangle **R** and the circle **D**.

The rectangle is formed as a result of the intersection of orthogonal lines that approximate the intersection area of the rectangle and the circle.
Algorithm
Preprocessing
We define two variables $(t1, t2)$ for the potential intersection points of the orthogonal line and the circle. We consider four half-planes formed by the rectangle:

- left half-plane $(c1 < x0)$;
- right half-plane $(c1 > x1)$;
- top half-plane $(c2 > y1)$;
- bottom half-plane $(c2 < y0)$.

In the equation of the circle, we substitute the equations of the boundaries of each half-plane and find the discriminants $(dx0, dx1, dy0, dr1)$ of the obtained square equations.
Approximation
We need to consider the following cases:

- the center of the circle is inside the rectangle:
  - If the center of the disk inside the rectangle we define the characteristics of rectangle **newR** as:
    $x0 = \max\{x0, c1 - r\}$;
    $x1 = \min\{x1, c1 + r\}$;
    $y0 = \max\{y0, c2 - r\}$;
    $y1 = \min\{y1, c2 + r\}$;
- the center of the circle lies in one of the half-planes and the circle and the rectangle have only one point of the intersection (discriminant is equal 0):
  - If the center of the circle lies in one of the half-planes and the circle and the rectangle have only one point of the intersection (discriminant is equal 0) we define the characteristics of rectangle **newR** as the point:
    $(x0, c2)$ – for left the half-plane;
    $(x1, c2)$ – for the right half-plane;
    $(c1, y1)$ – for top half-plane;
    $(c1, y0)$ – for bottom half-plane.
- the center of the circle lies in one of the half-planes and intersects the boundary of the half-plane at two points (discriminant is greater 0):

- If the center of the circle lies in one of the half-planes and intersects the boundary of the half-plane at two points (discriminant is greater 0) we consider half-plane corresponding to center of circle and solve the problem of finding local extremum of a function (circle) in a bounded domain ($x0 \leq x \leq x1, y0 \leq y \leq y1$).

Note: We consider the procedure in more detail using an example of the bottom half-plane.

The bottom half-plane

For the bottom half-plane we need consider the top half of the circle.
First, we find the points of intersection of the circle and the border of the half-plane. If the points lie on the side of rectangle, then we redefine $x0$ and $x1$ as:

$x0 = \max\{x0, t1\}$;

$x1 = \min\{x0, t2\}$.

If boundary point $(c1, c2+r)$ of the circle is inside the rectangle we can redefine $y1 = c2+r$, else we need to find the points of intersection (in our code $t1, t2$) of the circle with the sides orthogonal to the boundary of the half-plane and redefine $y1$ as:

$y1 = \min\{y1, \max\{t1, t2\}\}$.

Note: $x1 = \min\{x1, \max\{t1, t2\}\}$ for the left half-plane; $x0 = \max\{x0, \min\{t1, t2\}\}$ for right half-plane; $y0 = \max\{y0, \min\{t1, t2\}\}$ for top half-plane; $y1 = \min\{y1, \max\{t1, t2\}\}$ for bottom half-plane.

Output:

After we have updated all the characteristics, we form the required rectangle **newR**.