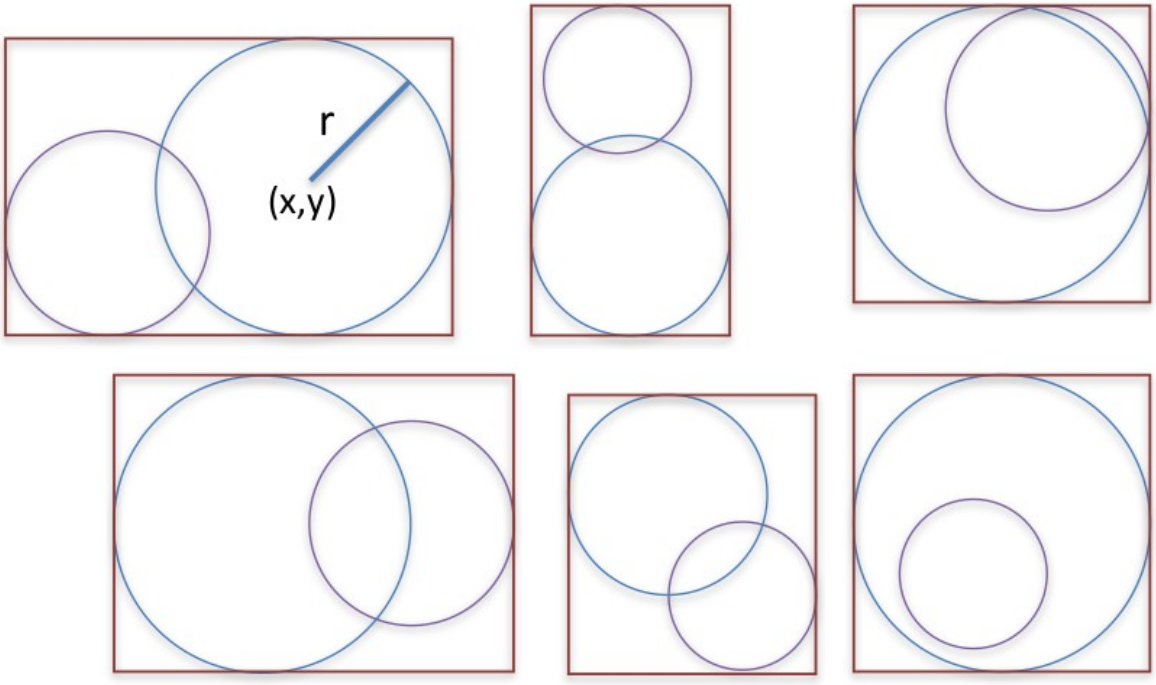
ANKARA UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
Computer Programming 1
Fall 2022-23

Programming Assignment 1
Assoc. Prof. Dr. İrem ÜLKÜ

Deadline: 14/11/2022; 23:55

Write a Python program which takes a sequence of circle parameters and computes the total areas of the bounding rectangles of the circles. The bounding rectangle of two intersecting circles is *the smallest rectangle that touches to at least one of the circles from left-right-up and down*. Some samples are depicted below to exemplify possible cases, note that these are not showing all possible cases.



All the circles that are provided through standard input will have bounding rectangles, either as pairs or as single circles. You will be printing the total area of all the bounding rectangles. The output format is specified below. Please also look at the sample I/O files.

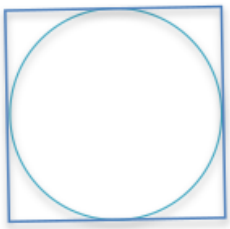
Specification details:

- Circle center (x, y) points and radius (r) parameters will be provided from the std. input. Each line will define a new circle. All the parameters will be provided as **integer** values. There will be at least one circle.

- The number of circles will be specified as the first argument from the input.

- A circle can intersect with at most one other circle. In other words, a circle ether insects with another circle or not. There is no other alternative.

- If a circle is not intersecting with another circle, the bounding rectangle will be defined using that circle only, as depicted below:



Constraints:

- You are *not allowed to use* ***any*** external modules for intersection computations. You are only allowed to use math.sqrt() function, if you need. You can import math module by: **import math** statement. Then you can call the sqrt function using math.sqrt(). All other necessary functions should be written by you. We'll test for originality and check your source contents for import statements. This constraint is rigid. You are free to use built-in types and operators and methods that work with these types of course.

Hints:

I suggest you to write functions for primary computational parts of this assignment. Otherwise, it will not be as easy as using functions, to construct the structure of your programs, i.e. it will probably be error prone. When you define a function, test it using some sample inputs that you design particularly for testing that function. Plan your progress in an iterative manner. That is, first implement some primary functions that are useful, then test and validate them. Then go on calling them to the next step forward. Do not try to implement everything from scratch without validating primitive parts. In every new step, such as reading the inputs part, computing circle intersections etc. check if everything is fine by testing, i.e monitoring the content that you read (like printing temporarily or debugging etc.) or testing your circle intersection function etc.

I/O Format: Please also check the sample IO files for the inputs and output formats
Input format:

<number_of_circles>

[<x_coord>[Space]<y_coord>[space]<radius>[Newline]]$^{+}$

Output format:

Print each circle data to the output in this format:

[**(** [Space] **<x_coord>**[Space]**<y_coord>**[Space] **)** [Space] **rad:** [Space] **<radius>** [Newline]]$^{+}$

Then, print the total rectangle area in this format:

Total rect area: <total_area>

Note that:

1- Sample input and output files are provided.

2- $[x]^+$ means, one or more elements of x

Testing:

You are provided with some sample I/O files. Assume that input.txt stands for a sample input and output.txt stands for the corresponding output file and your source code is named as PA1.py; you can test your program from the command line using the following commands. (>: stands for command prompt)

> python PA1.py < input.txt > my_output.txt

This command redirects the inputs in the input.txt file as if they are provided from the command line to your python code. The outputs, which you generate using the print() function in your source codes, are redirected to my_output.txt file.

You can then check if my_output.txt file is exactly the same with the provided output.txt file, using diff command from the command prompt:

> diff output.txt my_output.txt

Submission:

1- Name your Python source file as <student_id>.py; replace <student_id> using your student id number.

2- Upload your python file using the interface provided in e-kampüs course page.

3-Do not upload any files, other than your source file.

Important:

Please ask, if you have any questions related to the PA specifications, to the Announcement/ Duyurular part of our course e-kampüs page. I will not answer any PA related questions that are directed to me through my emails. You are also encouraged to share your additional test I/O files for your friends, for additional testing in that platform under “PA1-test data sharing” subject. All of you will benefit from the crowdsourcing that part in collaboration.

Have fun :)