

General instructions:

Read the questions carefully and make sure your plans work accordingly

If required, for each question you will receive an example output for that question, a student who does not follow these examples will have his grade reduced!

Homework is automatically checked according to the examples below,

Claims of various types arising from non-compliance with these requirements will not be considered.

Another highlight:

There is an organized skeleton that helps you not to make mistakes in printing, using it is a must and not a right, a student who nevertheless chooses not to use it and receives outputs that are different from the examples below (even at the character level) will have a score reduced and his appeal on the subject will be canceled.

Part A - Quadratic equation

Given the structure of a quadratic equation:

$$ax^2 + bx + c$$

Given (by the user) a,b,c check:

Is there a solution to the equation?

there are 2 cases:

- If there is a solution (single or 2 solutions) the function will output an appropriate print line (detailed below).
- If the case is that there is no solution/ infinite solutions (we do not work with imaginary numbers :)) of course a corresponding print line will be printed on the screen and it will appear later.

print the solution of the equation (of course if there is one or two solutions) in the correct print configuration as you learned in practice

Examples:

If a solution does not exist:

```
Please enter three numbers:  
1  
2  
3  
 $x^2+2x+3 = 0$  has no real roots
```

Single solution:

```
Please enter three numbers:  
1  
2  
1  
 $x^2+2x+1 = 0$  has one real root: -1
```

2 solutions:

```
Please enter three numbers:  
1  
5  
6  
 $x^2+5x+6 = 0$  has two real roots: -3 and -2
```

Infinite solutions

```
Please enter three numbers:  
0  
0  
0  
 $0 = 0$  has infinite real roots
```

Examples of additional cases:

```
Please enter three numbers:  
2  
0  
2  
2x^2+2 = 0 has no real roots
```

```
Please enter three numbers:  
1  
-1  
2  
x^2-x+2 = 0 has no real roots
```

```
Please enter three numbers:  
0  
1  
2  
x+2 = 0 has one real root: -2
```

The given skeleton will have several functions that you will have to complete:

void print_quadratic(double a, double b, double c)

A function that receives 3 numbers (of type DOUBLE) and prints as follows:

- If one of the numbers is 0 it will not be printed
- If a or b is 1, no $1x$ or $1x^2$ will appear (x and x^2 accordingly)
- If a or b is -1, $-1x$ or $-1x^2$ will not appear (-x and $-x^2$ accordingly)

int find_roots(double a, double b, double c, double* p1, double* p2)

A function that receives 3 numbers (of type DOUBLE) that represent

$$ax^2 + bx + c$$

Calculates the roots of the equation and outputs:

- 0 if there is no real solution to the equation
- 1 If there is a unique solution to the equation
- 2 if there are 2 solutions to the equation
- -1 if there are infinitely many solutions to the equation

Pay attention - the function also accepts p1,p2 which are the pointers to the variables that will hold the solution to the equation if there are any, they should of course be updated with the correct value).

if there is a single solution for the equation, p1 is the one that is updated with the solution value.

Part B: Triangle

In this section you will create functions that use the point structure and the triangle structure as shown below:

```
struct point {  
    double x, y;  
}Point;  
  
struct triangle {  
    Point p, q, r;  
}Triangle;
```

In this section, the input (the value of the coordinates of each point-p,q,r) will be accepted without SCANF functions.

Changing the input will be done manually by us after submission.

- **do not change the initial input given!**

The file that will be given to you will have the skeleton in which the functions are declared.

The definitions (DEFINE,INCLUDE) are already in the skeleton so that there is no confusion.

- **You cannot use libraries other than those included.**

Now for the exercise-

You will need to complete the following functions:

double perimeter(Triangle const* t)

A function that will return the perimeter of the triangle.

double area(Triangle const* t)

A function that will return the area of the triangle (HINT: Use the HERON formula for triangular area).

bool is_right(Triangle const* t)

A function that receives a structure of a triangle and checks if it is right angle triangle or not:

If it is right angle triangle return TRUE

if it is not a right angle triangle return FALSE

In addition to the above there are printing functions in the skeleton, you will have to complete the printing according to the attached example below, these are simple prints.

- **There is no space at the end of the sentence!**

Example: Given 3 points: (10,20) ,(-20,40) (5,5) The output at the end of the run:

```
The perimeter of a triangle with corners (10, 20), (-20, 40) and (5, 5) is 94.8785
The area of a triangle with corners (10, 20), (-20, 40) and (5, 5) is 275
The triangle with corners (10, 20), (-20, 40) and (5, 5) is not a right angle triangle
```

Another example:

```
The perimeter of a triangle with corners (0, 0), (0, 13) and (13, 0) is 44.3848
The area of a triangle with corners (0, 0), (0, 13) and (13, 0) is 84.5
The triangle with corners (0, 0), (0, 13) and (13, 0) is a right angle triangle
```

Make sure that the print is up to 4 digits after the decimal point.

Pay attention, in the second example you can see that there is only one number after the decimal point.

This is because after calculating and rounding the result, the print remains with one digit after the decimal point (using the %g).

Important highlights:

- Make sure the code is clean and has no duplicates (it will not damage the grade, but this is a tip for better work)
- A code that does not stop running in less than 5 seconds will not be checked and will be scored 0.
- The 2 programs must be saved as type C files inside the submission boxes of the form:
 - For the first part:
 - `ex4_parta_<id_num>.c`
 - For example: `ex4_parta_123456789.c`
 - For the second part:
 - `ex4_partb_<id_num>.c`
 - For example: `ex4_partb_123456789.c`