

# ECE 321 - Introduction to Software Engineering

## Homework 3

### Exercise 1 - Length of a line segment

A Cartesian coordinate system is a coordinate system that specifies each point uniquely in a plane by a pair of numerical coordinates, which are the signed distances to the point from two fixed perpendicular directed lines, measured in the same unit of length. The position of each point in a plane is identified with an ordered pair of numbers, in the form  $(x, y)$ , called the coordinates of the point. In geometry, a line segment is a part of a line that is bounded by two distinct end points, and contains every point on the line between its endpoints.

Write a function with name `length` that takes as arguments the Cartesian coordinates of two points  $A(x_1, y_1)$  and  $B(x_2, y_2)$  and it returns the length of the line segment  $AB$  (Figure 1). The coordinates, which are given by the user, and the length must be real numbers. The calculation of the length must be performed by using the Pythagorean Theorem.

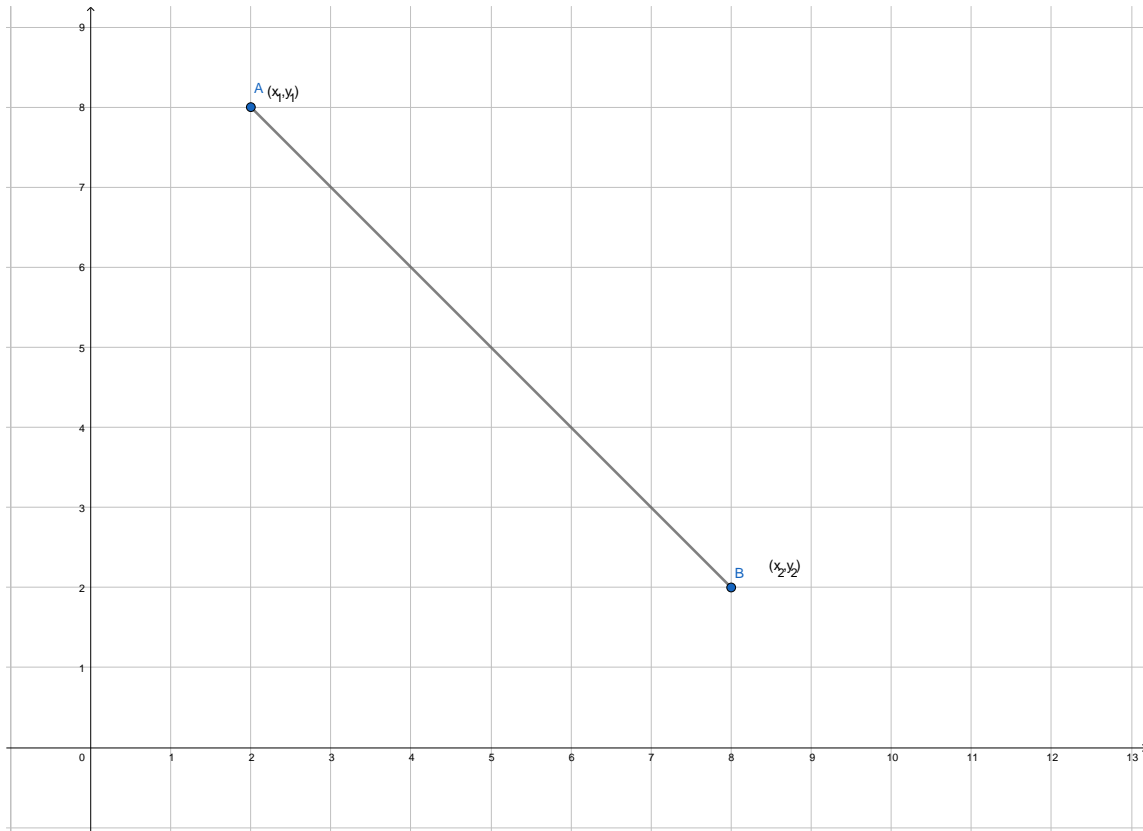


Figure 1: Line segment  $AB$  with Cartesian coordinates

Note: In order to calculate the length of the line segment, you will need to use the `sqrt` function that calculates the square root of a real number and which is declared in `<math.h>`.

## Exercise 2 - Area of a triangle

A triangle is a polygon with three edges and three vertices and it is one of the basic shapes in geometry. A triangle with vertices  $A$ ,  $B$ , and  $C$  is denoted as  $ABC$  (Figure 2). Write a function with the name **area**, which takes as arguments the Cartesian coordinates of the three vertices  $A(x_1, y_1)$ ,  $B(x_2, y_2)$  and  $C(x_3, y_3)$  and it returns the area of the triangle by using the Heron's formula:

$$A = \sqrt{\tau \cdot (\tau - a) \cdot (\tau - b) \cdot (\tau - c)}, \quad \tau = \frac{a + b + c}{2}$$

where  $a$ ,  $b$  and  $c$  are the lengths of the vertices  $BC$ ,  $AC$  and  $AB$  accordingly which you can calculate by calling the function **length** from Exercise 1.

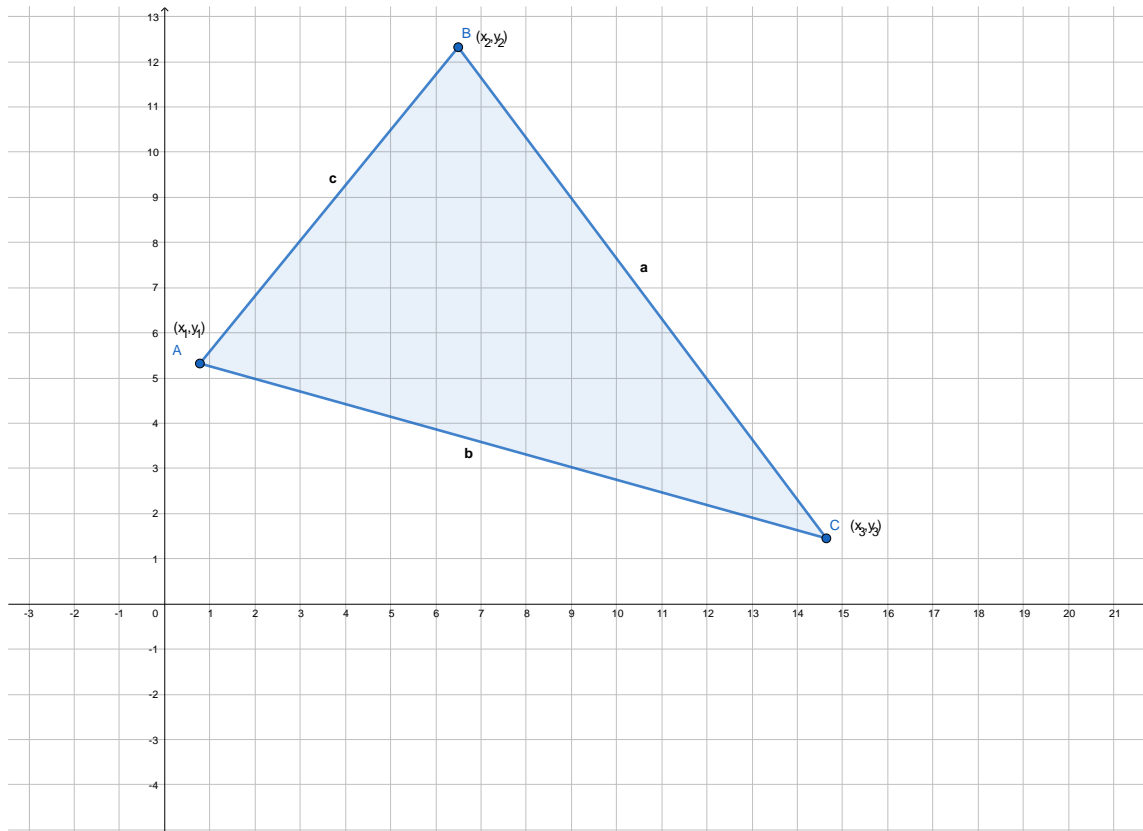


Figure 2: A triangle  $ABC$  with vertices  $A$ ,  $B$ , and  $C$

You should call the function **area** from a main program which will ask from the user to enter the coordinates  $x_1, y_1, x_2, y_2, x_3, y_3$  and call the function **area** with these values as arguments. The function must return the area of the triangle. The program should be continuous until the area is 0.

### Exercise 3 - Calculation of the square root of a number

The square root  $x$  of a real number  $r > 0$  with relative precision  $e > 0$  can be calculated with the following recursive process (Newton's method):

$$x_0 = \frac{r}{2}, \quad x_{n+1} = \frac{1}{2} \left( x_n + \frac{r}{x_n} \right), \quad \text{until} \quad \frac{|x_n - x_{n-1}|}{x_n} < e$$

Write a function with the name `mysqrt` that takes as arguments two real numbers  $r$  and  $e$  and calculates the square root according to the aforementioned process. Use the function `mysqrt` in a program where the user is asked to provide a positive real number and the program prints the square root of this number with relative precision  $10^{-6}$ . The program should be continuous until the user enters a negative number.

Note: In order to calculate the square root, you will need to use the `fabs` function that calculates the absolute value of a real number and which is declared in `<math.h>`.

### Exercise 4 - Comparison of different functions

Modify your solution in Exercise 2 in order to additionally print the area of the triangle using the `mysqrt` function from Exercise 4. Compare the accuracy of the two solutions by printing the error (difference) of the two values. How does the parameter  $e$  affect the error?

# General Software Development Standards and Guidelines

**Windows users:** *In order to use the functions as they were implemented in libc, you **may** have to put the following line as the first line of your code: (if you see an error message about function unsafety)*

***#define \_CRT\_SECURE\_NO\_WARNINGS***

Submissions must include only the source files (.c/.cpp) files. Also, each file must have an appropriate name e.g. `hw1-ex1.c`

Programming assignments will be evaluated based upon fulfillment of functional requirements, coding style and documentation.

Internal documentation improves the readability of a software module. Many of the general software development guidelines are focused on using good internal documentation practices. A file containing one or more software modules or a shell script file should have a comment block at its beginning containing the following basic information:

- The name of the author who created the file
- The date the file was created
- Description (overview of the purpose of the modules)
- A statement that declares who contributed in the development of the code
- Any other information that you believe it is necessary

General coding standards pertain to how the developer writes code:

**Indentation:** Proper and consistent indentation is important in producing easy to read and maintainable programs. Indentation should be used to:

- Emphasize the body of a control statement such as a loop or a select statement
- Emphasize the body of a conditional statement
- Emphasize a new scope block

A minimum of 3 spaces shall be used to indent. Generally, indenting by three or four spaces is considered to be adequate. Once the programmer chooses the number of spaces to indent by, then it is important that this indentation amount be consistently applied throughout the program.

**Structured Programming:** Structured (or modular) programming techniques shall be used. GO TO statements shall not be used as they lead to code, which is hard to read and maintain.

**Variable Names:** Variables shall have mnemonic or meaningful names that convey to a casual observer, the intent of its use.