Function Name: `distCalc`

Inputs:

1. *(double)* The x-coordinate of a point A
2. *(double)* The y-coordinate of a point A
3. *(double)* The x-coordinate of a point B
4. *(double)* The y-coordinate of a point B

Outputs:

1. *(double)* The distance between points A and B

Function Description:

This function will calculate the distance between the two Cartesian points defined by the four inputs. The formula for calculating distance between two point is:

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Round your answer to the nearest thousandth.

Hints:

1. Look up the `sqrt()` and `round()` functions.

Function Name: `quadSolver`

Inputs:

1. *(double)* The coefficient A of a quadratic polynomial
2. *(double)* The coefficient B of a quadratic polynomial
3. *(double)* The coefficient C of a quadratic polynomial

Outputs:

1. *(double)* The positive root of the polynomial
2. *(double)* The negative root of the polynomial

Function Description:

Gone are the days of memorizing the quadratic formula. You can now write a MATLAB function to compute the roots of quadratic equations for you!

In terms of the coefficients A, B, and C, the positive are negative roots are given by:

$$positive\ root\ =\ \frac{-B+\sqrt{B^2-4AC}}{2A}$$

$$negative\ root\ =\ \frac{-B-\sqrt{B^2-4AC}}{2A}$$

Notes:

- The terms “positive root” and “negative root” only refer to which sign is used in the quadratic formula, and do not indicate the sign of each root. Do not worry if your “positive root” output is a negative number, or vice versa.
- You will not be given coefficients that produce complex roots (although MATLAB will handle this automatically).
- Round your answers to the nearest hundredth.

Function Name: `ASA`

Inputs:

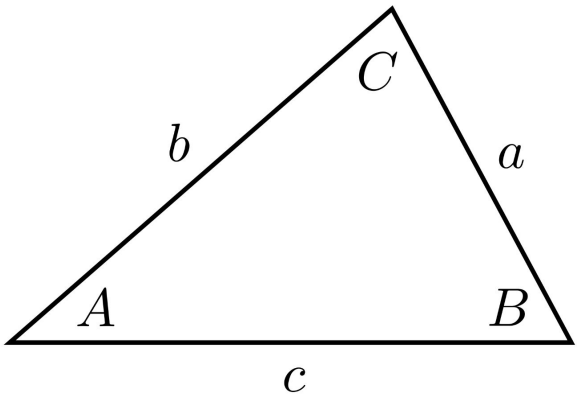
1. *(double)* An angle, "B", in degrees
2. *(double)* A side length, "a"
3. *(double)* An angle "C"

Outputs:

1. *(double)* The length of side "b"
2. *(double)* The length of side "c"

Function Description:

In case you don't remember middle school geometry, the Angle-Side-Angle formula for a triangle can be used to solve for the remaining side lengths when given a side and the two angles adjacent to that side. But no need to do any math, MATLAB can do it for you.



Using the labeling of the above triangle, you would be given the angle B as the first input, the side length a as the second input and the angle C as the third input. You should then calculate the lengths of sides b and c and output them as the first and second outputs, respectively. To help you calculate this, recall that the Law of Sines states:

$$\frac{sin(A)}{a} = \frac{sin(B)}{b} = \frac{sin(C)}{c}$$

Notes:

- You can use the `sind()` and `cosd()` function to perform trig operations in degrees.
- You should round your output to the nearest hundredth.

Function Name: `candy`

Inputs:

1. *(double)* size of the bag of candy
2. *(double)* number of kids

Outputs:

1. *(double)* pieces of candy per kid
2. *(double)* pieces of candy wasted

Function Description:

You are at a birthday party and buy a bag of candy to hand out to each of the kids who attend. But in order to be fair, every kid has to get the same number of pieces, and any pieces left over in the bag are considered to be wasted.

Write a function named "candy" that takes in the number of pieces of candy in a given bag and determine how many pieces of candy each kid gets, and how many pieces of candy are wasted.

As an example, if the size of the bag was 50 pieces, and there were 4 kids at the party, each kid would get 12 pieces of candy while 2 pieces of candy would be wasted.

Notes:

- You may find the `floor()` and `mod()` functions useful.