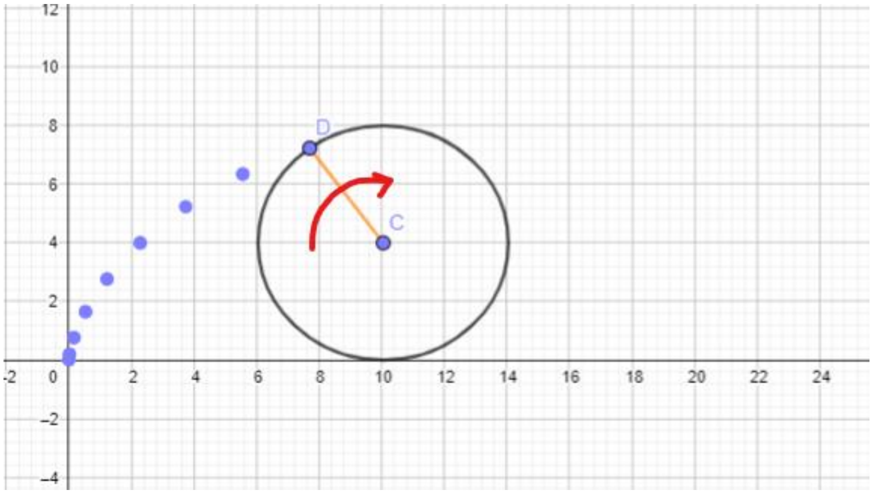
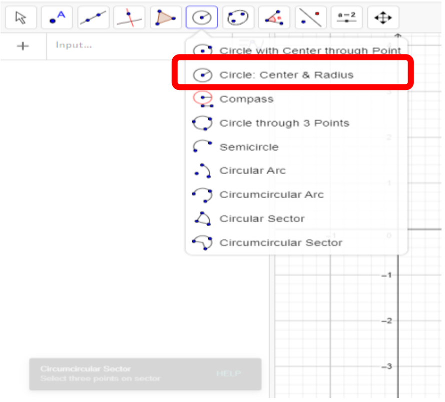
CELEN087 – LAB1 PROBLEMS

PROBLEM 1: *Simulating trace of a point on a rotating wheel.*

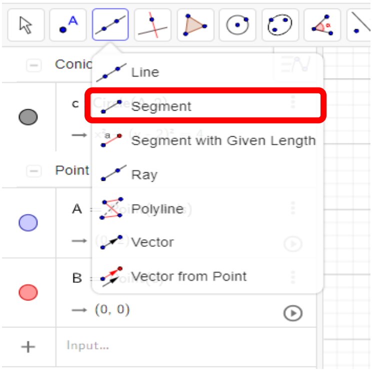


STEP1. Draw a circle centred at (0,4) with radius 4. You can use the toolbar for creating the circle.



STEP2. Choose any point on the circle and then move that point to (0,0). Change the colour of that point to red.

STEP3. Create a line segment from the centre of the circle to this point at (0,0); use the toolbar.



STEP4. Create a slider called `t = 0:pi/10:6pi`.

ASK ONE OF THE TUTORS TO CHECK YOUR PROGRESS

STEP5. Now, you should link the slider to the wheel (circle), i.e. as you change the slider, the wheel should rotate clockwise (or move to right). Think of a way to perform this link. ***Hint: the slider values represent angle of rotation (in radian), how much will the centre of the wheel move if the wheel rotates clockwise? Try to find a parametric equation for the centre of the wheel in terms of t.***

STEP6. Try to find a parametric equation for the point (0,0) on the circumference of the wheel in terms of `t`. ***Hint: you can search for it on the Internet, e.g. Google!***

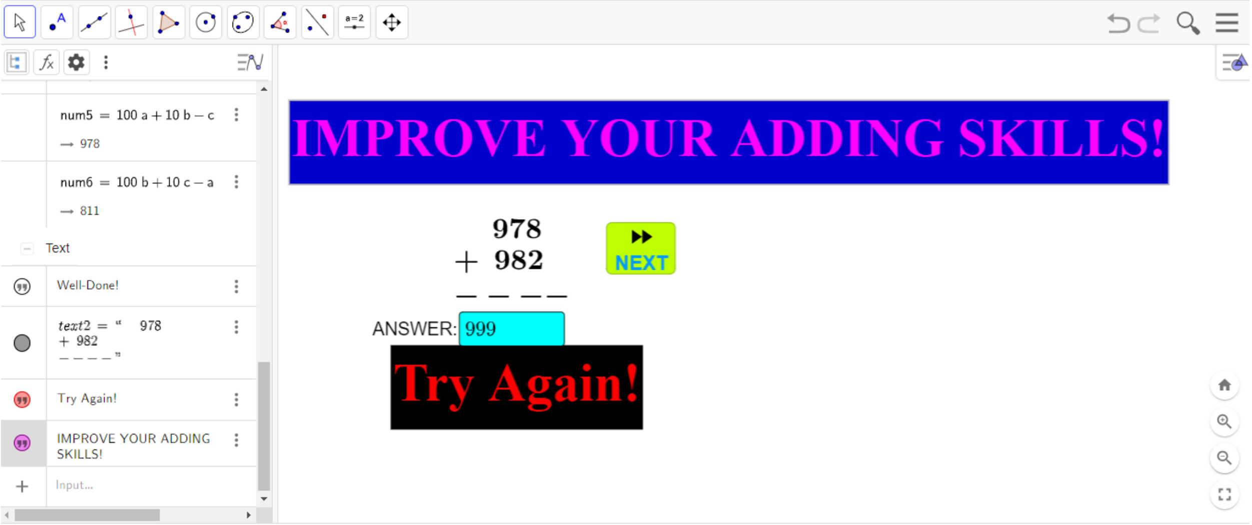
STEP7. Now change the expression of the centre point of the wheel and the point (0,0) in the input menu of GeoGebra. You need to type the parametric equations in terms of the slider value `t`.

STEP8. Now as you change the slider the wheel should move. Now right-click on the point on the circle and choose `"Show Trace"` option. What happens now when the wheel rotates?

ASK ONE OF THE TUTORS TO CHECK YOUR PROGRESS

OPTIONAL: how can you make the point on the circle to draw a solid-line curve as the wheel moves towards right?

PROBLEM 2: *Creating a calculator for primary school children.*



Users of this applet will add two 3-digit integers. Each integer contains the digits 1-9 (0 is not included). You are going to make each integer randomly.

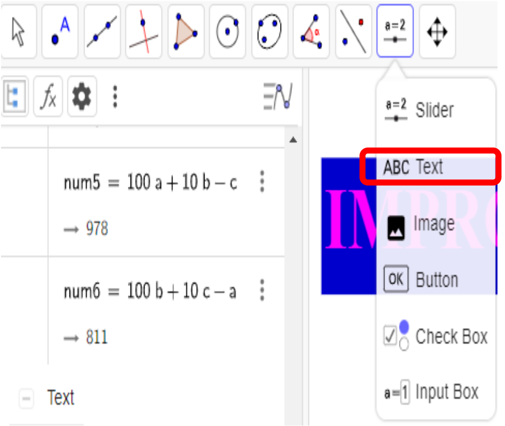
STEP1. Create three random variables **`a`**, **`b`** and **`c`**. These variables will make up the digits of your integers. Use **`RandomBetween()`** command.

STEP2. Create 6 different integers from these random digits. Call these variables **`num1`**, **`num2`** … **`num6`**. For example: **`num1=100a+10b+c`**.

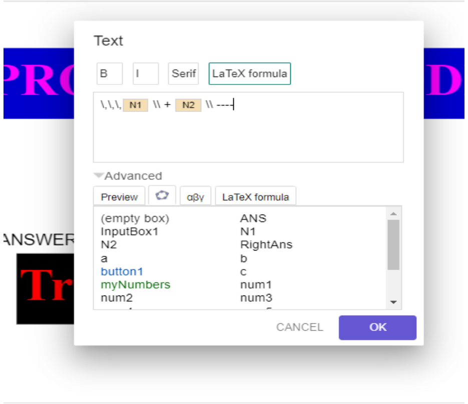
STEP3. Make a list called **`myNumbers`** and put all the 6 integers created above inside this list.

STEP4. Now you choose two numbers ***randomly,*** from this list. Call the first one **`N1`** and the second one **`N2`**. Use **`RandomElement()`** command to define **`N1`** and **`N2`**.

STEP5. Click on Text from the toolbar menu above in order to create a *Dynamic TextBox*.



STEP6. In the dialogue box choose ***LaTeX Formula*** option. Then include **`N1`** and **`N2`** from the ***Advanced*** menu below it. See the example given here ⇒⇒



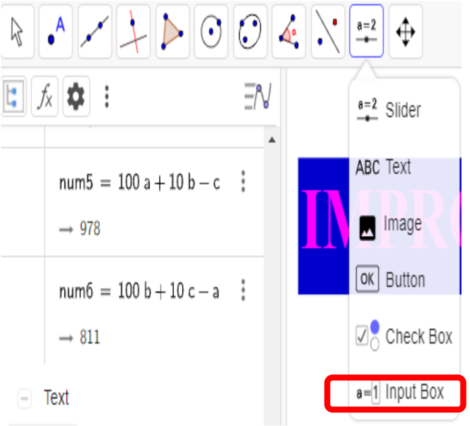
Text

`\,\,\, N1 \\ + N2 \\ ----`

ASK ONE OF THE TUTORS TO CHECK YOUR PROGRESS

STEP7. Create two variables called: **`ANS`** and **`RightAns`**. Let **`ANS=0`**; this is the answer that the user will input. The variable **`RightAns`** is the actual answer of **`N1+N2`** to be compared with the user's input.

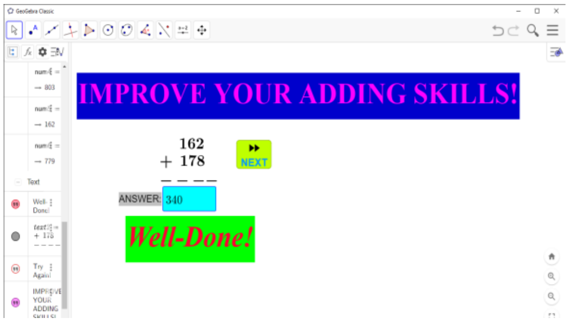
STEP8. From the toolbar menu choose input box. Name the input box as ***"ANSWER="***. ***HINT: The linked object to the input box should be the `ANS` variable.***



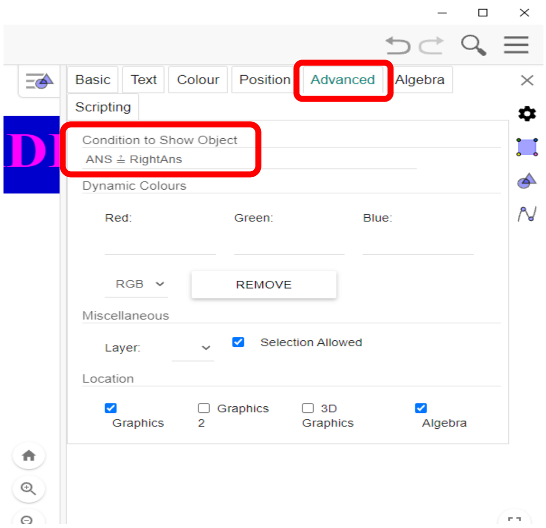
num5 = 100 a + 10 b − c → 978

num6 = 100 b + 10 c − a → 811

STEP9. Now the user will put the answer (sum) in the input box and here your APP should decide whether the input answer is correct or not. For this we need a TEXT (see **STEP5**) to appear with a message like "WELL-DONE!" when the answer is correct and "TRY AGAIN!!" when the answer is wrong.



You should modify the "condition to show object" option in the settings of each of these text messages.



Condition to Show Object
ANS ≟ RightAns

Dynamic Colours
Red:
Green:
Blue:

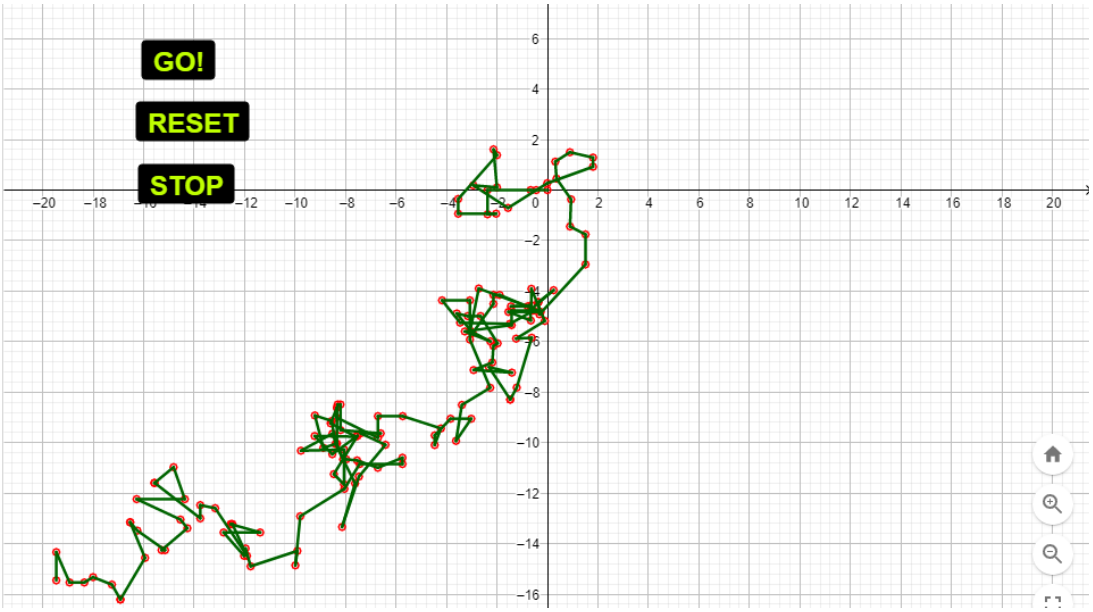
Miscellaneous
Layer:
☑ Selection Allowed

Location
☑ Graphics
□ Graphics 2
□ 3D Graphics
☑ Algebra

STEP10. Add a button that will update the entire construction. It should create two new 3-digit integers (**`N1,N2`**) and it should also reset the input box value to 0.

ASK ONE OF THE TUTORS TO CHECK YOUR PROGRESS

PROBLEM 3: *Simulating a random walk.*



In this problem you will create random points on the 2D plane, i.e. the x and y-coordinates are selected randomly in a range from -1 to 1. You can accomplish this by using the following commands:

```
xx= RandomUniform(-1,1)
yy= RandomUniform(-1,1)
```

Each time that the above commands are executed, new random coordinate values are generated. So you can make a random point by the following command:

```
A=(xx,yy)
```

You then need to make a slider for the number of points that you want to make and a **GO** button to start the process. The button should start the animation by linking to the slider. Suppose your slider name is **jumps**, then inside the scripting option for your **GO** button you should add:

```
StartAnimation(jumps,true)
```

This will start the slider, i.e. counts the number of random jumps. Now to update the coordinates of the point **A** that you created above, you should call the following command inside the slider:

```
UpdateConstruction()
```

This way, each time that the slider value changes, a new random point will be created.

Now you need to **append** all these points into a list to create a list of points. Also, the value of each new random point **A**, must be added to the previous point in the list. See the following example:

```
First random point A: A=(0.34,-0.10)

Append to list of points: L={(0.34,-0.10)}

Second random point A: A=(-0.20,0.15)

Append to list of points: L={(0.34,-0.10),(0.34-0.20,-0.10+0.15)}

So the updated list is: L={(0.34,-0.10),(0.14,0.05)}

Next random point A: A=(-0.54,-0.89)

Append to list: L={(0.34,-0.10),(0.14,0.05),(-0.40,-0.84)}
```

………….

You should use **`SetValue()`** and **`Append()`** command inside your slider scripting section.