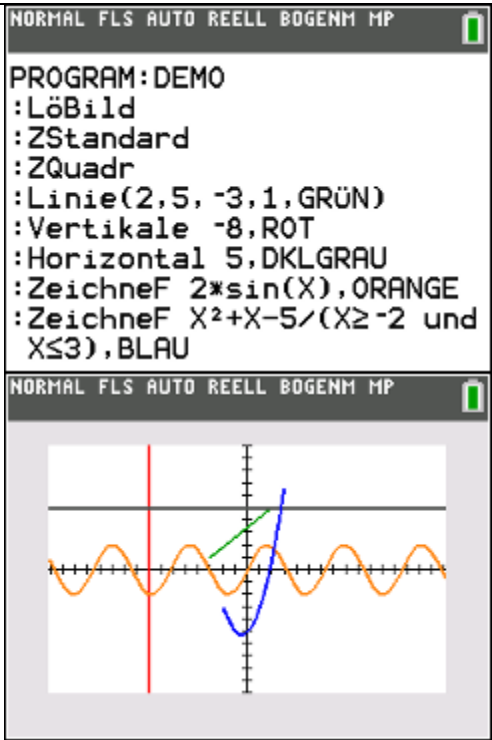

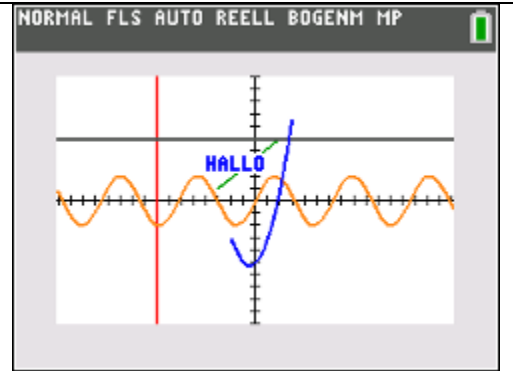


<p><b>Lektion 5: Grafiken</b></p>	<p><b>Übung 3: Geraden, Text und Farben</b></p>
<p>In der dritten Übung von Lektion 5 geht es um das Zeichnen von Linien und Text in verschiedenen Farben.</p>	<p><b>Lernziele:</b></p> <ul style="list-style-type: none"> <li>• Verwendung der Befehle, mit denen man Geraden, Funktionen und Text zeichnen kann.</li> <li>• Verwendung der Farbe in Zeichenbefehlen.</li> <li>• Einsatz von Formeln im Zusammenhang mit Grafiken.</li> </ul>
<p><b>Zeichnen von Geraden und Kurven</b></p> <p><b>Linie(X,Y,W,Z)</b> zeichnet eine Strecke zwischen den Punkten (X/Y) und (W/Z). Optionale Zusätze finden sich in der Kataloghilfe.</p> <p><b>Vertikale A</b> zeichnet die senkrechte Gerade X=A.</p> <p><b>Horizontal B</b> zeichnet die waagerechte Gerade Y=B.</p> <p><b>ZeichneF X<sup>2</sup>+X</b> zeichnet direkt den Funktionsgraphen von <math>Y = X^2+X</math>.</p> <p>Rechts sind ein paar Beispiele angegeben, die das Bild darunter erzeugen. Die Farben findet man im Menü <b>PRGM</b> FARBE, das es aber nicht auf dem TI-84 Plus gibt.</p> <p>Will man einen Funktionsgraphen nur in einem bestimmten Intervall zeichnen, so dividiert man durch das Intervall: <b>ZeichneF X<sup>2</sup>+X-5/(x≥-2 und x≤3)</b></p>	 <pre> NORMAL FLS AUTO REELL BOGENM MP PROGRAM: DEMO :LöBild :ZStandard :ZQuadr :Linie(2,5,-3,1,GRÜN) :Vertikale -8,ROT :Horizontal 5,DKLGRAU :ZeichneF 2*sin(X),ORANGE :ZeichneF X^2+X-5/(X≥-2 und X≤3),BLAU     </pre>
<p><b>Hinweis:</b> <b>ZeichneF</b> ist nicht dasselbe wie die Verwendung von <b>Y=</b> und <b>GRAPH</b>. Ein mit <b>ZeichneF</b> hergestellter Funktionsgraph kann nicht punktweise abgetastet oder für Berechnungen verwendet werden, denn <b>ZeichneF</b> zeichnet nur einzelne Pixel, die den Verlauf des Graphen wiedergeben. Löscht man das Display, ist jegliche mit <i>draw</i> erstellte Graphik verschwunden und muss mit einem Befehl wieder neu gezeichnet werden im Gegensatz zu den mit <b>Y=</b> erstellten Grafiken, die automatisch wieder neu gezeichnet werden. <b>Linie( )</b> zeichnet eine Strecke, <b>Vertikale</b> und <b>Horizontal</b> hingegen Geraden, die über den ganzen Bildschirm verlaufen. Aber natürlich kann man <b>Linie</b> so programmieren, dass die Geraden auch über das ganze Display verlaufen (s. Übung unten).</p>	
<p><b>Zeichnen von Text</b></p> <p>Der Befehl <b>Text(</b> ist eine Besonderheit, denn er verwendet für die Positionierung Pixel statt der sonst üblichen Punkte. Außerdem gibt es im Editiermodus den separaten Befehl <b>TextFarbe(</b>, mit dem die Farbe für den nächsten Text bestimmt wird.</p>	 <pre> NORMAL FLS AUTO REELL BOGENM MP ZEICHNE PKTE SPCH HINTERGR 3↑Horizontal 4:Vertikale 5:Tangente( 6:ZeichneF 7:Schraff( 8:ZchInv 9:Kreis( 0:Text( A:TextFarbe(     </pre>



`Text(50,100,"HALLO")` lässt den Text HALLO immer an derselben Stelle im Display erscheinen unabhängig von den Fenstereinstellungen. Zeile 50 Spalte 100 bezieht sich dabei auf die obere linke Ecke des darzustellenden Textes.

*Hinweis: Beim TI-84 Plus gibt es 96 Spalten x 64 Reihen, beim TI-84 Plus CE-T 265 Spalten und 165 Zeilen.*



**Hinweis:** Der `Text`-Befehl ist tückisch, denn er verwendet als einziger der Zeichenbefehle Pixelkoordinaten anstelle von Fensterkoordinaten. Man kann sie aber umrechnen – s.u..

#### Programmieren mit Algebra und dem Linie - Befehl.

Mit diesem Programm wird der Befehl `Linie` („verlängert“, denn aus den Punktkoordinaten werden neue Koordinaten berechnet, so dass die Gerade bis zu den Fenstergrenzen oder sogar darüber hinaus verlängert wird. Allerdings braucht man dazu Algebrakenntnisse – also Vorsicht!

1. Das neue Programm soll `LINIE` heißen.
2. Zunächst kommen die üblichen Graphik-Befehle.
3. Mit zwei `Input` – Befehlen werden auf dem Grafikbildschirm die Koordinaten zweier Punkte eingelesen. Da `Input` in den Variablen `X` und `Y` abspeichert, werden die Werte des ersten `Input` auf `A` und `B` übertragen.

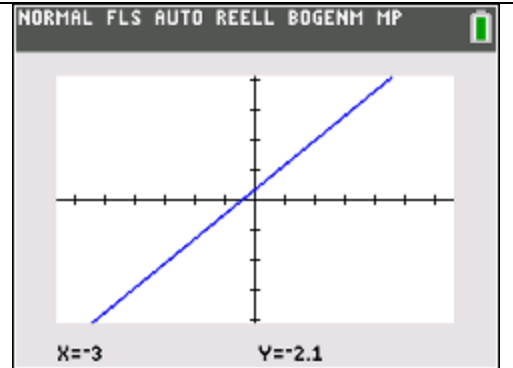


4. Jetzt kann man die Steigung der Geraden berechnen:  $M = (Y - B) / (X - A)$

5. Die Punkte am linken und rechten Fensterrand haben die Koordinaten `Xmin` und `Xmax`. Die dazugehörigen y-Koordinaten berechnet man mit der Punkt-Steigungs-Form:  $Y = M \cdot (X - A) + B$

#### Aufgabe:

1. Diese y-Werte sollen berechnet und in den Variablen `Q` und `R` abgespeichert werden.



**Lösung:**  $M \cdot (Xmin - A) + B \rightarrow Q$   
 $M \cdot (Xmax - A) + B \rightarrow R$

2. Jetzt kann die neue Strecke gezeichnet werden:  
`Linie(Xmin,Q,Xmax,R)`

#### Erweiterung:

1. Einbau einer Schleife, so dass man mehrere Strecken zeichnen kann, ohne dass das Programm erneut gestartet werden muss, wodurch alle vorhergehenden Strecken gelöscht würden.
2. Bei einer senkrechten Linie erfolgt eine Fehlermeldung – warum? Durch den Einbau eines `If` – Befehles kann dieser Spezialfall aufgefangen werden.

**Lösung:** Die Steigung ist dann undefiniert. Das Programm versucht eine Division durch Null, was durch die Fehlermeldung angezeigt wird.

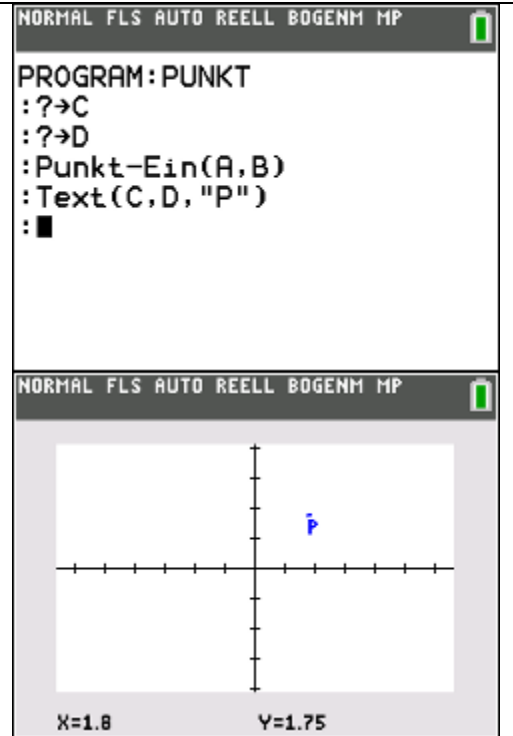
**Punktkoordinaten in Pixelkoordinaten umwandeln**

Mit dem Befehl **Punkt-Ein**( soll der Punkt (A/B) gezeichnet und anschließend mit dem Text „P“ bezeichnet werden. Wie müssen die Koordinaten aussehen?

Dazu muss man zwei Formeln entwerfen, die **A** und **B** in die Pixelkoordinaten **C** und **D** umwandeln, damit man mit dem Befehl **Text**( arbeiten kann. Die Tabelle für den TI-84 Plus CE-T kann hier hilfreich sein:

<u>Fenster</u>	<u>Pixel</u>
<b>Xmin</b>	0
<b>Xmax</b>	264
<b>A</b>	?
<b>Ymax</b>	0
<b>Ymin</b>	164
<b>B</b>	?

*Hinweis: Im Befehl **Text**( ist das erste Argument die Reihe, gehört also zur y-Koordinate des Punktes!*



**Hinweis:** Hier hilft wieder eine lineare Transformation. Dazu muss man (Xmin/0) und (Xmax/264) als Punkte einer Geraden ansehen. Ihre Steigung ist:

$$(264-0)/(Xmax-Xmin)$$

Damit wird aus der x-Koordinate A die Zeilennummer D:

$$264/(Xmax-Xmin) \cdot (A-Xmin) \rightarrow D$$

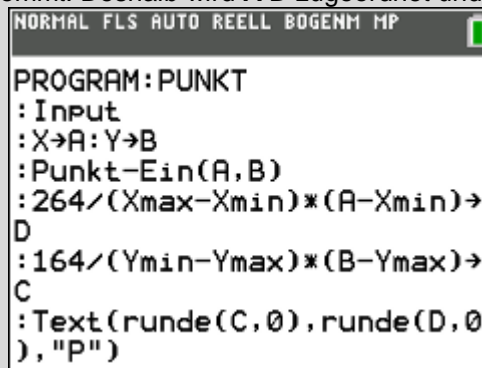
Für y erhält man auf gleiche Weise:

$$(164-0)/(Ymin-Ymax)$$

Damit wird aus der y-Koordinate B die Spaltennummer C:

$$164/(Ymin-Ymax) \cdot (B-Ymax) \rightarrow C$$

Auch hier sei noch einmal darauf verwiesen, dass bei pixel-orientierten Befehlen zuerst die Spalte, dann die Reihe kommt. Deshalb wird **A D** zugeordnet und **B C**.



**runde()** ist notwendig, da der Textbefehl unbedingt ganze Zahlen aus dem definierten Zahlbereich benötigt. Ohne diesen Befehl erhält man eine Fehlermeldung!

Beim TI-84 Plus muss 95 anstelle von 264 und 63 anstelle von 164 verwendet werden.