$\frac{1}{4} - \frac{9x}{4}$

2c

```
x, y = sp.symbols('x y')
curve_eq = y**2 - (x**3 + 3*x**2)

#differentiate implicitly
dy_dx = sp.diff(curve_eq, y) * sp.Derivative(y, x) + sp.diff(curve_eq, x)
dy_dx_solution = sp.solve(dy_dx, sp.Derivative(y, x))[0]

#set the derivative equal to zero for horizontal tangents
horizontal_tangent_eq = sp.Eq(dy_dx_solution, 0)

#solve for x where the tangent is horizontal
horizontal_tangent_x = sp.solve(horizontal_tangent_eq, x)

#find corresponding y values for each x
horizontal_tangent_points = []
for x_val in horizontal_tangent_x:
    y_vals = sp.solve(curve_eq.subs(x, x_val), y)
    for y_val in y_vals:
        horizontal_tangent_points.append((x_val, y_val))

#output the points of horizontal tangents
print(f"Horizontal tangents at: {horizontal_tangent_points}")
```

```
Horizontal tangents at: [(-2, -2), (-2, 2), (0, 0)]
```

2d

```
#define the curve equation
x, y = sp.symbols('x y')
curve_eq = y**2 - (x**3 + 3*x**2)

#get tan line
tangent_line_eq = sp.Eq(y, slope1 * (x - x1) + y1)

#find horizontal tan points
horizontal_tangent_points = []
for x_val in horizontal_tangent_x:
    y_vals = sp.solve(curve_eq.subs(x, x_val), y)
    horizontal_tangent_points.extend([(x_val, y) for y in y_vals])

horizontal_tangent_lines = [sp.Eq(y, y_h) for _, y_h in horizontal_tangent_points]

#plot the curve
p1 = plot_implicit(curve_eq, (x, -4, 4), (y, -4, 4), line_color='blue', show=False, title="Tschirnhauser’s Cubic and Tangent Lines")
p2 = plot_implicit(tangent_line_eq, (x, -4, 4), (y, -4, 4), line_color='red', show=False)
p2.title = "Tangent Line at (1, -2)"

#add horiztonal tan lines
for h_line in horizontal_tangent_lines:
    p_h = plot_implicit(h_line, (x, -4, 4), (y, -4, 4), line_color='green', show=False)
    p1.extend(p_h)

#combine and show the plots
p1.extend(p2)
p1.show()
```