Group Member Names:

In [1]:
```
from sympy import *
from sympy.plotting import plot, plot_parametric
```

Lab 3 Template

Each part of each problem should be solved in its own cell.

Question 1

A conchoid is given by the parametric equations $x(t) = k\cos(t) + b, y(t) = k\sin(t) + b\tan(t)$, and the cartesian equation is given by $(x-b)^2(x^2+y^2) = k^2x^2$ where b and k are given constant.

a.) Show that the parametric equations are consistent with the cartesian equation by substituting $x(t)$ and $y(t)$ into the left and right hand sides of the cartesian equation and seeing that they are equal. You will need to use **.simplify()** to ensure they look the same.

b.) If $b = 2$ and $k = 3$, plot the two parts of the conchoid on one set of axes by plotting the conchoid first on [$-\frac{\pi}{2} + .1$, $\frac{\pi}{2} - .1$], then by plotting on [$\frac{\pi}{2} + .1$, $\frac{3\pi}{2} - .1$]. Be sure to use **plot_parametric** and show only the final plot with both parts of the conchoid on it.

c.) If $b = 0$ and $k = 1$, what shape do you get? Put your answer in a print statement. (Hint: Try plotting the parametric equations on [0, 2π], but do not include the plot in your final answer)

In [ ]:

In [ ]:

In [ ]:

Question 2

For this problem, we will be investigating the properties of limits numerically and analytically.

a.) Consider $(1+\frac{1}{t})^t$, for $t > 0$. Use list comprehension to evaluate this function at $t = 1, 10, 100, 1000, 10000$, making sure to use **.evalf()** to get decimal approximations. In a print statement, what number does this seem to be approaching?

b.) Using the **limit** command, what is $\lim_{t\to\infty}(1+\frac{1}{t})^t$?

c.) A factorial (usually denoted as an exclamation point as in $n!$) multiplies an integer by all integers smaller than it. For example $4! = 4 \cdot 3 \cdot 2 \cdot 1$. In sympy, you can compute a factorial using the **factorial** command i.e. **factorial(4)** would output 24. Consider $\frac{n}{(n!)^{1/n}}$ where n is a positive integer. Use list comprehension to evaluate this function at $n = 1, 10, 20, 30, 50, 100, 500, 1000$, making sure to use **.evalf()** to get decimal approximation. (This operation may take a few moments to process) In a print statement, what number does this seem to be slowly approaching? (Note: this limit comes from the well known "Sterling's Formula" which is used to approximate factorials.)

In [ ]:

```

```

In [ ]:

```

```

In [ ]:

```

```

Question 3

Consider the function $f(x) = \begin{cases} a - bx^2 & x \leq 0 \\ x + b & 0 < x \leq 2 \\ ax^3 - bx^2 + 6x & x > 2 \end{cases}$

Find constants a, b such that $f(x)$ is continuous everywhere, then plot the entirety of $f(x)$ on one plot from [-2, 3].

In [ ]:

```

```