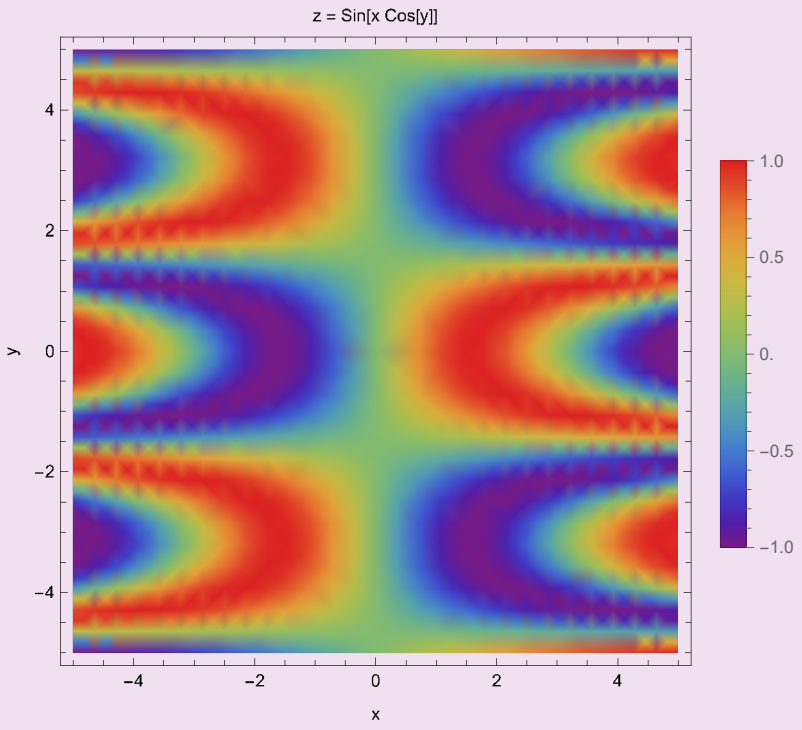
In[13]:=

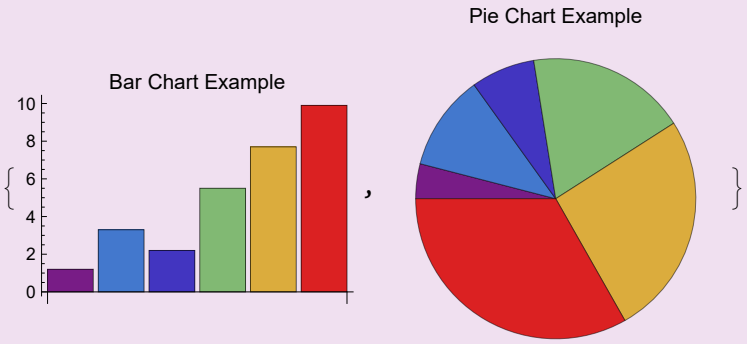
```
(*定义函数*)f[x_, y_] := Sin[x Cos[y]]
                         正弦   余弦

(*绘制密度图*)
densityPlot = DensityPlot[f[x, y], {x, -5, 5},
              密度图
    {y, -5, 5}, ColorFunction → "Rainbow", PlotLegends → Automatic,
                颜色函数                   绘图的图例          自动
    FrameLabel → {"x", "y", "z = Sin[x Cos[y]]"}, PlotRange → All];
    边框标签                  正弦   余弦          绘制范围        全部

(*绘制等值线图*)
contourPlot = ContourPlot[f[x, y] == z, {x, -5, 5}, {y, -5, 5},
              绘制等高线
    Contours → Range[-1, 1, 0.2], FrameLabel → {"x", "y"}, PlotLegends → Automatic];
    等高线          范围                  边框标签                  绘图的图例          自动

(*显示图形*)
Show[densityPlot, contourPlot]
显示
```

Out[16]=



In[17]:=
```
(*示例数据*)data = {1.2, 3.3, 2.2, 5.5, 7.7, 9.9};

(*绘制棒图*)
barChart = BarChart[data, ChartLabels → Automatic, FrameLabel → {"Index", "Value"},
           条形图          图表标签          自动          边框标签
   PlotLabel → "Bar Chart Example", ColorFunction → "Rainbow"];
   绘图标签                        颜色函数

(*绘制饼图*)
pieChart = PieChart[data, ChartLabels → Automatic,
           饼图           图表标签          自动
   PlotLabel → "Pie Chart Example", ColorFunction → "Rainbow"];
   绘图标签                        颜色函数

(*显示图形*)
{barChart, pieChart}
```

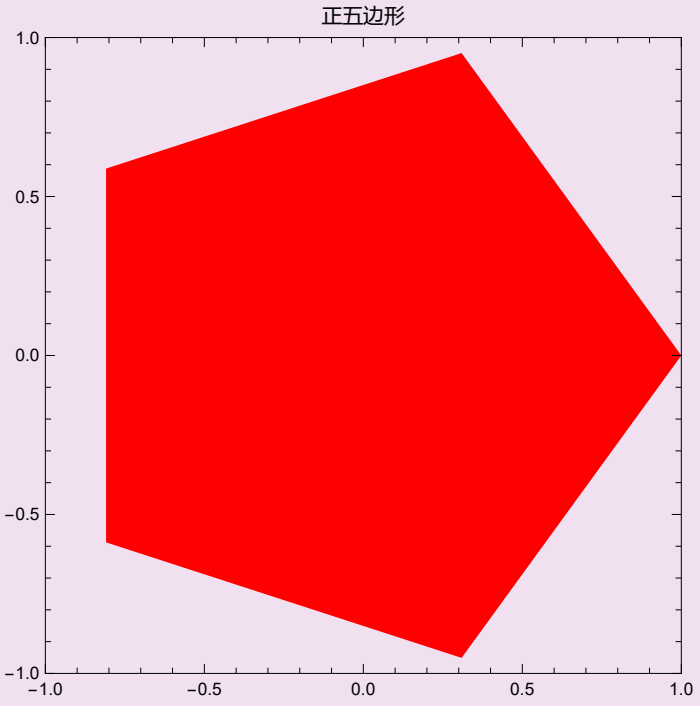
Out[20]=



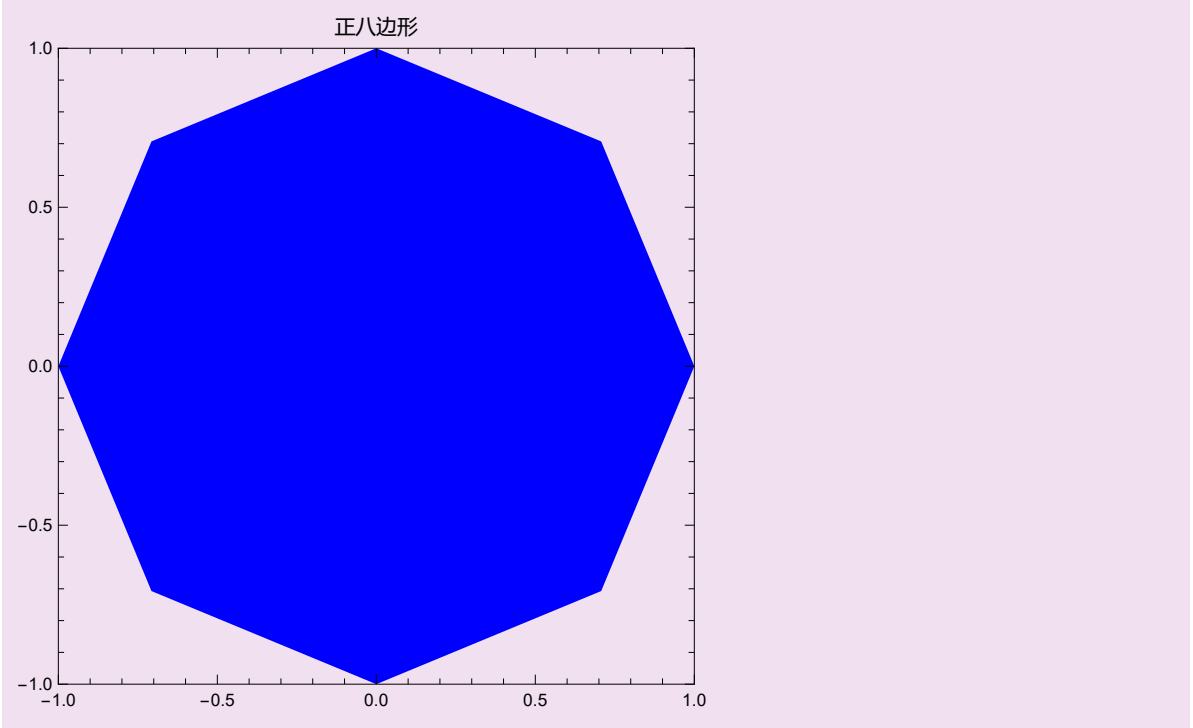
In[28]:=

```
(*定义正五边形的顶点*)
pentagonVertices = Table[{Cos[2 Pi k / 5], Sin[2 Pi k / 5]}, {k, 0, 4}];
                   表格     余弦   圆周率     正弦   圆周率

(*定义正八边形的顶点*)
octagonVertices = Table[{Cos[2 Pi k / 8], Sin[2 Pi k / 8]}, {k, 0, 7}];
                  表格     余弦   圆周率     正弦   圆周率

(*绘制正五边形和正八边形*)
Graphics[{Red, Polygon[pentagonVertices](*正五边形*)},
图形       红色  多边形

 PlotRange → {{-1.5, 1.5}}, AspectRatio → 1, Frame → True, PlotLabel → "正五边形"]
 绘制范围                    宽高比             边框      真      绘图标签

Graphics[{Blue, Polygon[octagonVertices]} , PlotRange → {{-1.5, 1.5}},
图形       蓝色  多边形                                  绘制范围

 AspectRatio → 1, Frame → True, PlotLabel → "正八边形"
 宽高比           边框      真      绘图标签
]
```

Out[30]=



Out[31]=



In[42]:=

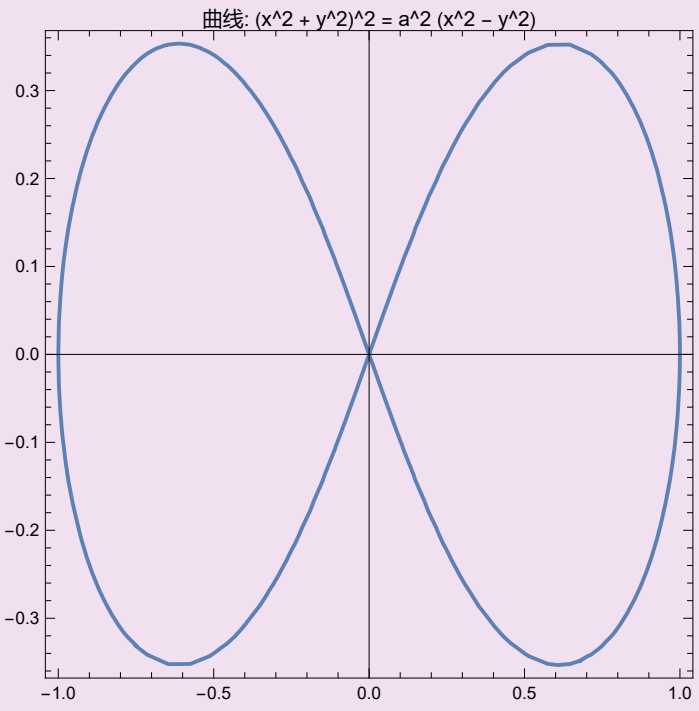
```
(*心脏线*)heartParam = Table[
                       表格
    {16 Sin[t]^3, 13 Cos[t] - 5 Cos[2 t] - 2 Cos[3 t] - Cos[4 t]}, {t, 0, 2 Pi, 0.01}];
      正弦          余弦        余弦          余弦        余弦              圆周率

(*绘制心脏线*)
heartShape = Graphics[{Pink, Line[heartParam]}];
             图形      粉色  线段

(*显示图形*)
heartShape
```

Out[44]=

In[54]:=

```
(*定义参数 a*)a = 1;

(*绘制曲线*)
curvePlot = ContourPlot[(x^2 + y^2) ^2 == a^2 (x^2 - y^2),
            绘制等高线
    {x, -2, 2}, {y, -2, 2}, Contours → 50, PlotRange → All, Axes → True,
                             等高线          绘制范围      全部  坐标轴  真
    AspectRatio → 1, PlotLabel → "曲线: (x^2 + y^2)^2 = a^2 (x^2 - y^2)"];
    宽高比              绘图标签

(*显示图形*)
curvePlot
```

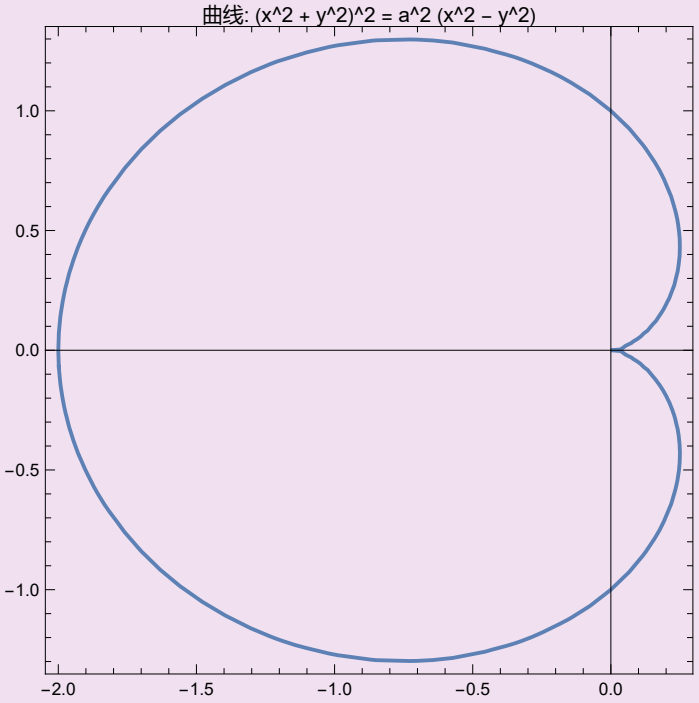
Out[56]=



In[66]:=

```
(*定义参数 a*)a = 1;

(*绘制曲线*)
curvePlot = ContourPlot[(x^2 + y^2) == a (Sqrt[x^2 + y^2] - x),
            绘制等高线                    平方根
    {x, -2, 2}, {y, -2, 2}, PlotRange → All, Axes → True, AspectRatio → 1,
                            绘制范围        全部  坐标轴  真      宽高比
    PlotLabel → "曲线: (x^2 + y^2)^2 = a^2 (x^2 - y^2)"];
    绘图标签

(*显示图形*)
curvePlot
```

Out[68]=



In[69]:=
```
(*定义函数*)f[x_] := x^2 (*可以修改为其他函数*)

(*割线的两个点*)
a = -2; (*左边点*)
b = 2; (*右边点*)

(*动态演示*)
Manipulate[Module[{slope, tangentLine}, slope = (f[b] - f[a]) / (b - a);
交互式操作  模块
  (*计算割线的斜率*)tangentLine[x_] := f[a] + slope (x - a);
  (*割线方程*)(*绘图*)Plot[{f[x], tangentLine[x]}, {x, -3, 3}, PlotStyle → {Blue,
                          绘图                                  绘图样式  蓝色
    Red}, PlotRange → {-1, 5}, AxesLabel → {"x", "y"}, PlotLabel → "割线与切线逼近",
    红色  绘制范围              坐标轴标签                  绘图标签
   Epilog → {PointSize[Large], Point[{a, f[a]}], Point[{b, f[b]}]}]],
   绘制主…  点的大小  大  点  点
 {a, -2, 2}, (*左边点可移动*){b, a + 0.1, 3} (*右边点必须大于左边点*)]
```

Out[72]=