Polynomial Regression

Importing the libraries

In [21]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

In [22]:
```
dataset = pd.read_csv('Position_Salaries.csv')
x = dataset.iloc[:, 1:-1].values
y = dataset.iloc[:, -1].values
```

In [23]:
```
print(x)
```

```
[[ 1]
 [ 2]
 [ 3]
 [ 4]
 [ 5]
 [ 6]
 [ 7]
 [ 8]
 [ 9]
 [10]]
```

In [24]:
```
print(y)
```

```
[  45000   50000   60000   80000  110000  150000  200000  300000  500000
 1000000]
```

Training the Linear Regression model on the whole dataset

In [25]:
```
from sklearn.linear_model import LinearRegression
modelLinear = LinearRegression()
modelLinear.fit(x, y)
```

Out[25]:
```
LinearRegression()
```

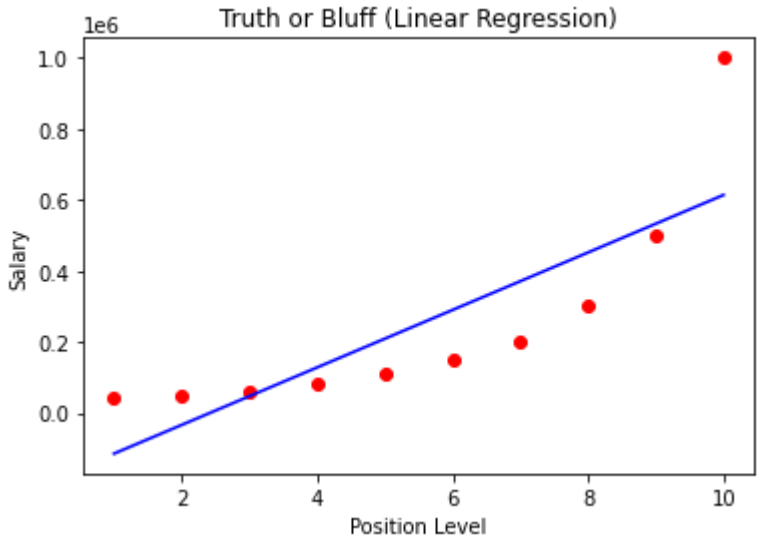
Training the Polynomial Regression model on the whole dataset

In [26]:
```
from sklearn.preprocessing import PolynomialFeatures
modelPoly = PolynomialFeatures(degree = 6)
x_poly = modelPoly.fit_transform(x)
modelLinearPoly = LinearRegression()
modelLinearPoly.fit(x_poly, y)
```

Out[26]:
```
LinearRegression()
```

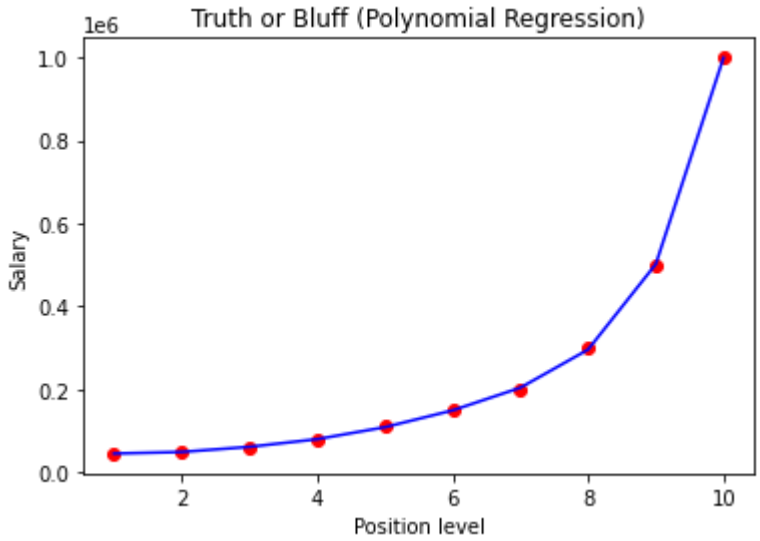
Visualising the Linear Regression results

In [27]:
```
plt.scatter(x, y, color = 'red')
plt.plot(x, modelLinear.predict(x), color = 'blue')
plt.title('Truth or Bluff (Linear Regression)')
plt.xlabel('Position Level')
plt.ylabel('Salary')
plt.show()
```



Visualising the Polynomial Regression results

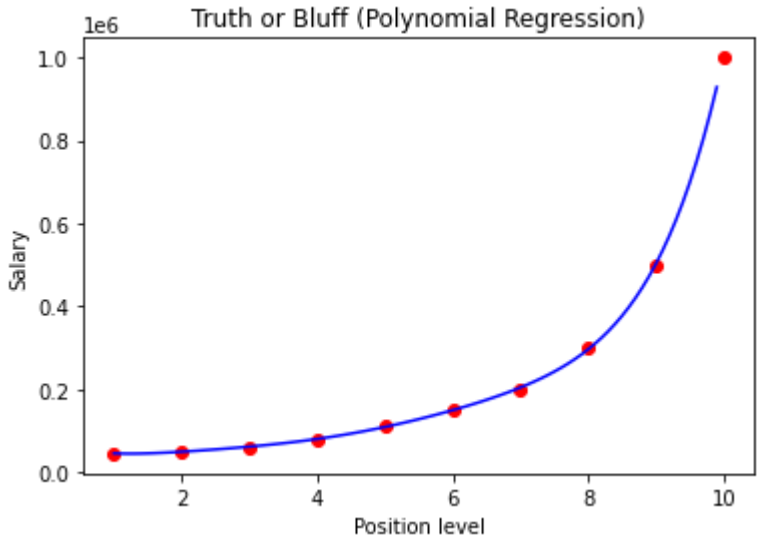
In [28]:
```
plt.scatter(x, y, color = 'red')
plt.plot(x, modelLinearPoly.predict(modelPoly.fit_transform(x)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
plt.ylabel('Salary')
plt.show()
```



Visualising the Polynomial Regression results (for higher resolution and smoother curve)

In [29]:
```
X_grid = np.arange(min(x), max(x), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(x, y, color = 'red')
plt.plot(X_grid, modelLinearPoly.predict(modelPoly.fit_transform(X_grid)), color = 'blue')
plt.title('Truth or Bluff (Polynomial Regression)')
plt.xlabel('Position level')
```

```
plt.ylabel('Salary')
plt.show()
```



Predicting a new result with Linear Regression

In [30]:
```
modelLinear.predict([[6.5]])
```

Out[30]:
```
array([330378.78787879])
```

Predicting a new result with Polynomial Regression

In [31]:
```
modelLinearPoly.predict(modelPoly.fit_transform([[6.5]]))
```

Out[31]:
```
array([174192.81930584])
```

In [ ]:
```

```