Simulation Project

- Name: Aakansh Murali
- Class period: 4th Period
- Problem number: 3
- Problem text: A couple plans to have children until they have a girl or until they have four children, whichever comes first. Estimate the likelihood that they will have a girl. Assume that each child has probability 0.5 of being a girl and 0.5 of being a boy, and the sexes of successive children are independent.

Partial Trial Setup

- Example: 1 person walks into the ice cream parlor to order either vanilla, chocolate or strawberry ice cream

In [4]:

```python
# Write code to run a partial trial
# Refer to your previous simulations as a resource!
import numpy

single_baby = numpy.random.randint(1,101)

if single_baby <= 50:
    print("Boy")
else:
    print("Girl")
```

```
Boy
```

Partial Trial Function

- Can you give the above procedure a descriptive name so you can easily call it multiple times?

In [25]:

```
# Define a function that runs a partial trial and returns its result appropriat

# Description: Prints if you get a boy or girl based on a 50/50 chance
# Citations: None
# Input/Parameters: None
# Output: Boy or Girl

def single_baby_trial():
    single_baby = numpy.random.randint(1,101)

    if single_baby <= 50:
        return("Boy")
    else:
        return("Girl")

print(single_baby_trial())
```

```
Boy
```

Full Trial Setup

- Example: 5 people walk into the ice cream parlor together: do 2 or more order chocolate?

In [54]:

```
# Write code to run a FULL trial
# Refer to your previous simulations as a resource!

import numpy

boy_kids = 0
girl_kids = 0

while girl_kids < 1 and boy_kids < 4:
    result = single_baby_trial()
    if result == "Girl":
        girl_kids += 1
    else:
        boy_kids += 1

total_number_of_babies = girl_kids + boy_kids

print(total_number_of_babies)
```

```
4
```

Full Trial Function

- Can you give the above procedure a descriptive name so you can easily call it multiple times?

In [7]:

```
# Define a function that runs a FULL trial and returns its result appropriately

# Description: Returns the amount of babies out of 4, if there is one girl ther
# Citations: None
# Input/Parameters: Loop stops when there is a girl or 4 children
# Output: Total number of kids

def full_baby_trial():
    boy_kids = 0
    girl_kids = 0

    while girl_kids < 1 and boy_kids < 4:
        result = single_baby_trial()
        if result == "Girl":
            girl_kids += 1
        else:
            boy_kids += 1

    total_number_of_babies = girl_kids + boy_kids

    return(total_number_of_babies)
```

Full Simulation

- Make sure to look at your previous simulations for the number 10,000

In [55]:

```
# Run your full trial a large number of times
# Record your data in an array

baby_results = []

for i in range(10000):
    results = full_baby_trial()
    baby_results.append(results)

print(baby_results)
```

```
[1, 1, 1, 2, 2, 1, 4, 1, 1, 1, 4, 1, 4, 2, 1, 2, 2, 1, 4, 4, 1, 1, 1, 1, 2,
1, 1, 2, 1, 1, 1, 1, 4, 1, 3, 2, 1, 1, 2, 2, 1, 4, 2, 1, 1, 2, 1, 1, 3, 3,
1, 2, 1, 1, 1, 3, 2, 1, 3, 4, 4, 1, 1, 1, 2, 2, 2, 3, 1, 4, 1, 1, 1, 3, 3,
1, 2, 2, 3, 2, 3, 2, 1, 1, 3, 2, 1, 4, 2, 1, 1, 3, 2, 4, 1, 4, 1, 4, 4, 1,
1, 1, 2, 1, 1, 1, 1, 1, 1, 3, 1, 1, 1, 1, 1, 3, 2, 4, 2, 1, 1, 1, 1, 4, 1,
3, 1, 2, 4, 1, 2, 1, 4, 2, 2, 1, 3, 3, 3, 1, 1, 2, 1, 3, 1, 1, 1, 1, 1, 4,
1, 1, 1, 4, 1, 1, 2, 1, 2, 2, 1, 2, 3, 1, 4, 4, 4, 1, 1, 1, 2, 1, 2, 1, 2,
1, 3, 1, 1, 1, 2, 1, 1, 2, 2, 1, 2, 1, 4, 1, 2, 3, 4, 3, 1, 1, 3, 1, 2, 2,
2, 1, 2, 1, 2, 1, 4, 1, 3, 1, 2, 1, 4, 2, 1, 1, 4, 1, 1, 2, 4, 3, 1, 1, 2,
4, 1, 2, 4, 3, 2, 4, 1, 4, 2, 1, 2, 3, 2, 4, 1, 4, 4, 3, 1, 2, 4, 1, 2, 2,
2, 1, 4, 1, 4, 4, 1, 3, 1, 1, 3, 1, 1, 3, 1, 4, 2, 2, 2, 1, 2, 1, 2, 1, 1,
2, 2, 1, 4, 1, 4, 1, 3, 4, 1, 1, 1, 1, 2, 2, 1, 3, 1, 1, 1, 4, 1, 3, 1, 1,
1, 1, 1, 2, 1, 2, 3, 1, 2, 3, 1, 1, 4, 4, 1, 2, 2, 1, 4, 2, 1, 2, 2, 1, 2,
1, 1, 1, 2, 1, 4, 1, 2, 3, 3, 2, 3, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 1, 2,
1, 2, 3, 2, 1, 3, 4, 1, 4, 1, 1, 1, 1, 1, 1, 1, 1, 1, 3, 1, 2, 1, 1, 1, 1,
1, 4, 1, 1, 4, 1, 2, 2, 2, 1, 3, 1, 4, 1, 2, 1, 2, 2, 1, 1, 4, 1, 4, 1, 1,
3, 1, 1, 1, 4, 1, 1, 4, 1, 3, 1, 4, 4, 2, 2, 2, 4, 1, 2, 1, 2, 1, 2, 4, 2,
4, 1, 2, 1, 2, 2, 1, 2, 4, 1, 3, 1, 3, 2, 2, 1, 4, 1, 1, 2, 4, 1, 2, 1, 1,
2, 1, 4, 3, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 1, 3, 2, 2, 2, 1, 1, 1, 2,
```

Calculate the Results

In [56]:

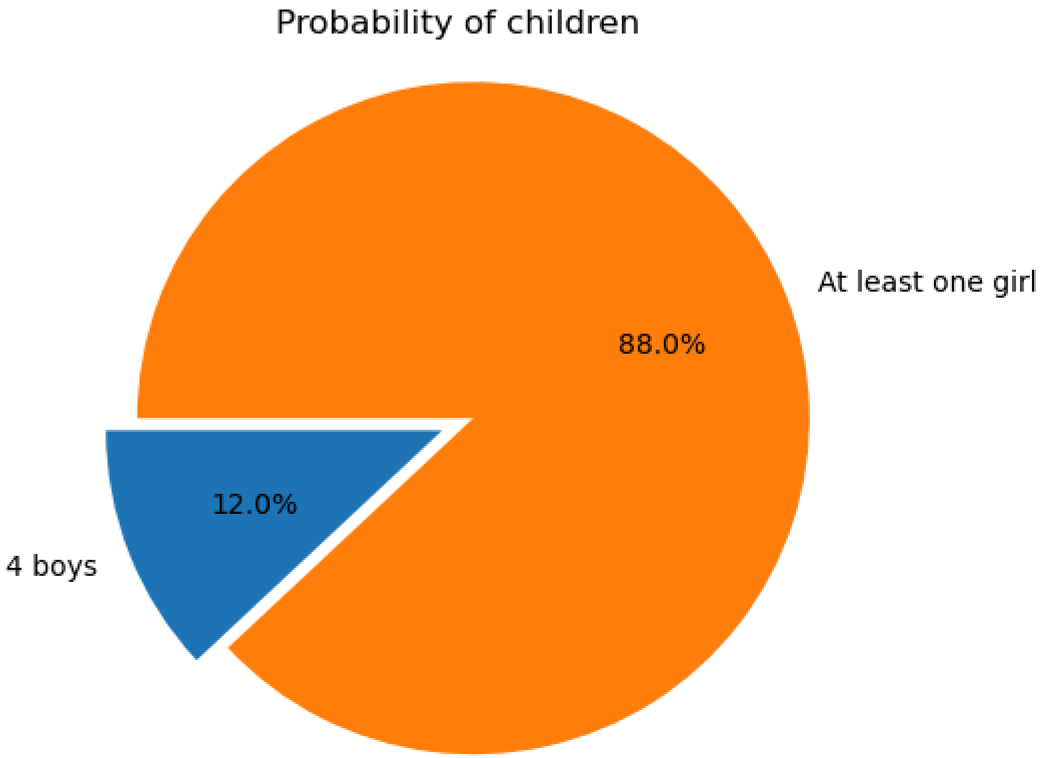
```
# Using the results array, grab appropriate data and perform the necessary calc

sum_of_boys = 0
sum_of_girls = 0

for trial in baby_results:
    if trial == 4:
        sum_of_boys += 1
    else:
        sum_of_girls += 1

probability_boys = sum_of_boys/100
probability_girls = sum_of_girls/100

print(probability_girls)
```

```
88.03
```

Display the Results

In [61]:

```
# Graphically display results related directly to your question

import matplotlib.pyplot as plot
%matplotlib inline
# Magic to allow the graph to display directly in this notebook

# Create an array of labels
labels = ["4 boys", "At least one girl"]

# Create an array of your results
results = [probability_boys, probability_girls]

# Explode option
    # 'Slices' appear distanced from the center
        # Larger numbers = further explosion
    # Explode array should be same size as labels
explode = (0, 0.1)

# Use matplotlib module subplots() to get data for various charts
    # Returns a tuple in the form (figure,axes)
fig1,ax1 = plot.subplots()

# Use axes to create a pie chart
    # ax1.pie(data array, explode array, labels array, starting angle)
ax1.pie(results, explode, labels, autopct='%1.1f%%', startangle=180)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax1.axis('equal')
plot.title("Probability of children")

plot.show()
```


Probability of children
At least one girl
88.0%
12.0%
4 boys

Answer

According to my simulation, there is a 88.0% likelihood that they will have a girl out of 4 babies

Additional Question

- State one additional question of interest (of your choice) using the data from your simulation

What is the probability that there second child will be a girl?

Calculate the Results

In [11]:

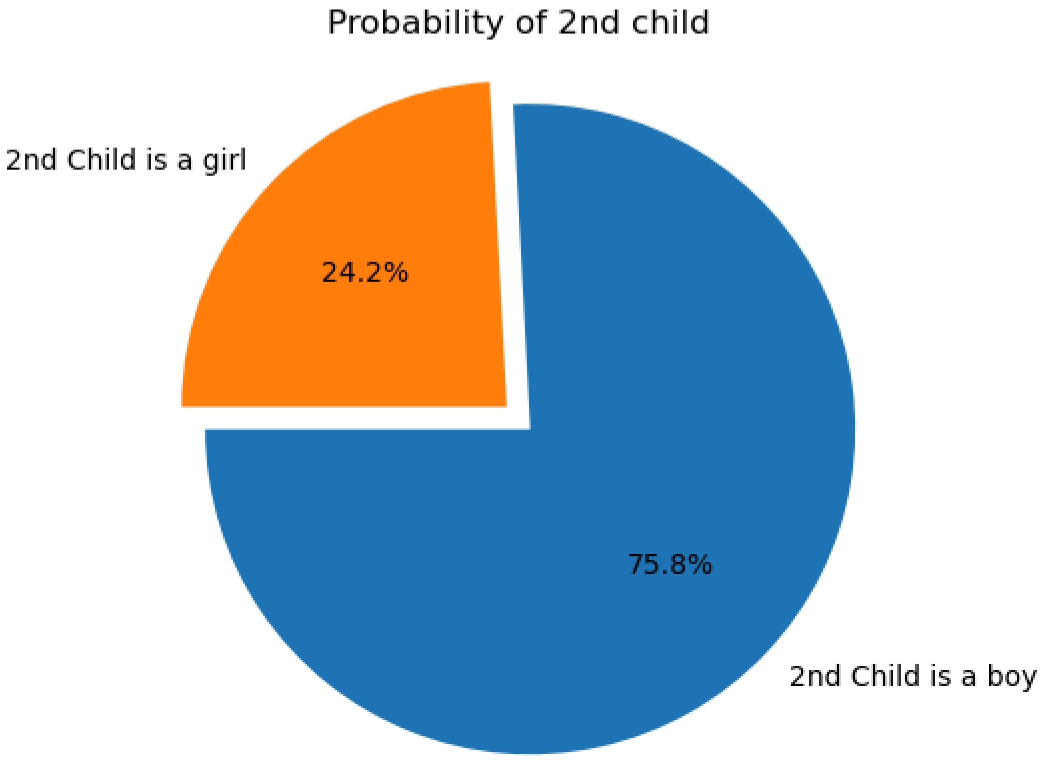
```
# Using the results array, grab appropriate data and perform the necessary calc

second_girl = 0
second_boy = 0

for trial in baby_results:
    if trial == 2:
        second_girl += 1
    else:
        second_boy += 1

prob_second_girl = second_girl/100
prob_second_boy = second_boy/100

print(prob_second_girl)
```

```
24.16
```

Display the Results

In [62]:

```
# Graphically display results related directly to your question

import matplotlib.pyplot as plot
%matplotlib inline
# Magic to allow the graph to display directly in this notebook

# Create an array of labels
labels = ["2nd Child is a boy", "2nd Child is a girl"]

# Create an array of your results
results = [prob_second_boy, prob_second_girl]

# Explode option
    # 'Slices' appear distanced from the center
        # Larger numbers = further explosion
    # Explode array should be same size as labels
explode = (0, 0.1)

# Use matplotlib module subplots() to get data for various charts
    # Returns a tuple in the form (figure,axes)
fig1,ax1 = plot.subplots()

# Use axes to create a pie chart
    # ax1.pie(data array, explode array, labels array, starting angle)
ax1.pie(results, explode, labels, autopct='%1.1f%%', startangle=180)

# Equal aspect ratio ensures that pie is drawn as a circle.
ax1.axis('equal')
plot.title("Probability of 2nd child")

plot.show()
```



According to my simulation, there is a 24.2% likelihood that there second child will be a girl

In [ ]: