



**FROOT
LOOPS**



Loops II



BLUFF CITY LAW

COMING SOON

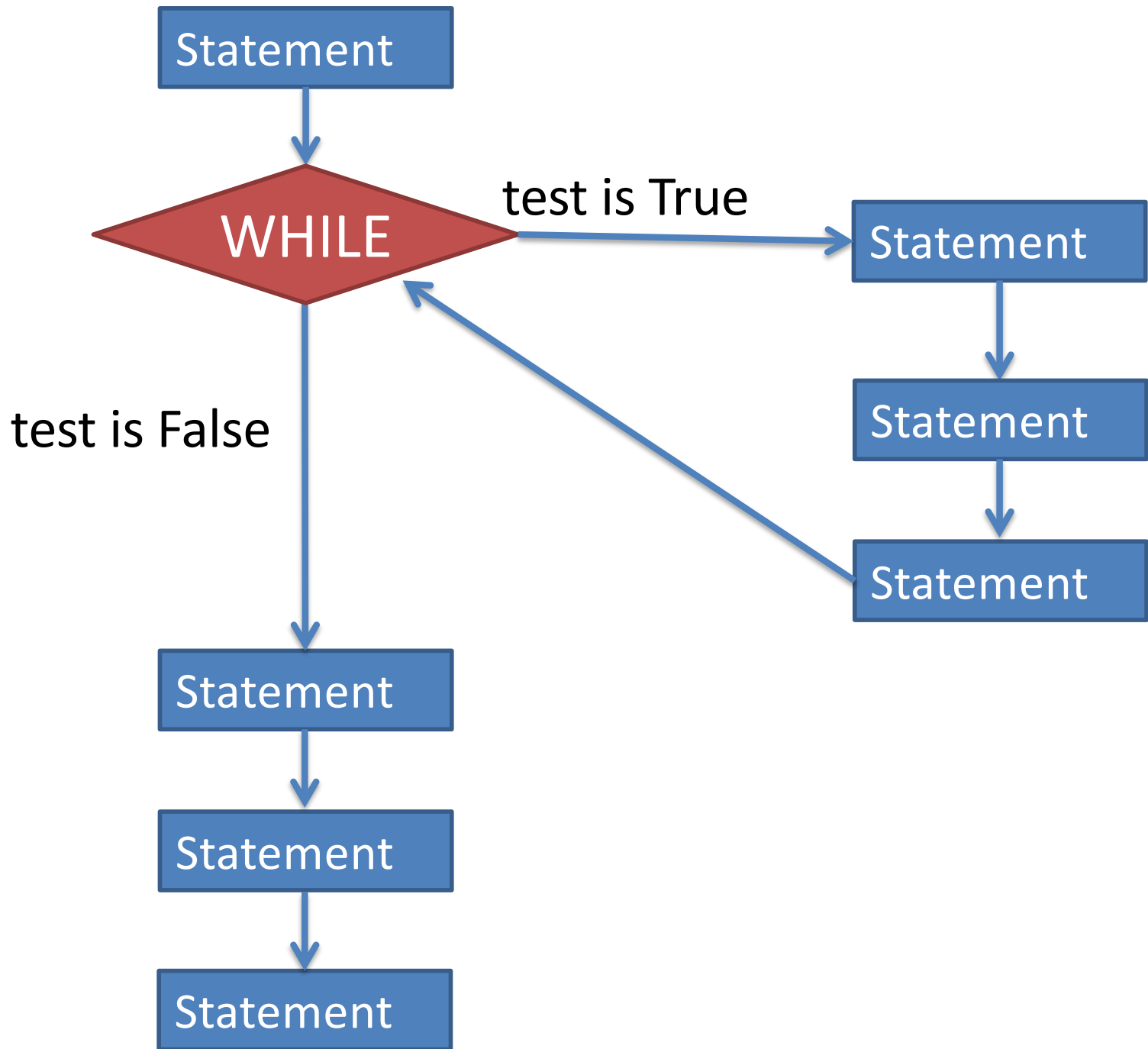


#BLUFFCITYLAW

```
while test :  
    statement  
    statement  
    more statements...  
statement  
statement  
more statements...
```

The *test* must be something that is True or False.

The indented statements are called the *body* of the loop.



- Pseudocode is an informal way of writing algorithms for humans to read (not computers!)
- Illustrates the logic of an algorithm, but omits details that people can fill in automatically.
- You get to make it up as you go along, as long as you (and other people) can easily understand it.

- Instead of saying

```
name = input("What is your name?")
```

- Pseudocode might use a line that says

```
name = ask user for name
```

- Instead of saying

```
if x >= 0 and x <= 100:  
    print("$%.2f" % x)
```

- Pseudocode might use a line that says

```
if x is between 0 and 100:  
    print x with 2 decimal places
```

The point is to get your ideas down on paper quickly, so you can worry about programming details and exact syntax later.

- Suppose we want to write the following loop:
- We have a super-secret password-protected computer, and we need to ask the user for their password. They can have as many tries as they want until they get it right.

1. Ask user for password.

2. If password is wrong, **keep going**. If password is correct, **exit the loop**.

3. Print incorrect password message.



Magic Formula for Any While Loop

- Figure out what the steps of the loop are. Write them in pseudocode as numbered steps in an actual circle.
- Be sure to include ***exactly one step*** of the loop that has a ***condition that keeps the loop going***, and the opposite condition ***ends the loop***.
- Starting from step #1, turn each line of pseudocode into real Python code.
- When you get to the step of the loop that involves the condition that keeps the loop going, that is where you place the while *condition* : part, filling in the blank with the condition.
- Continue turning the pseudocode into Python code, but now place subsequent steps *inside* the while loop (indented). Keep writing code inside the loop until you get back to step #1. You must duplicate any pseudocode steps of the loop that have lower numbered steps than the loop condition step. In other words, if your loop condition step is #3, then steps #1 and #2 will appear before the loop **and** inside it.

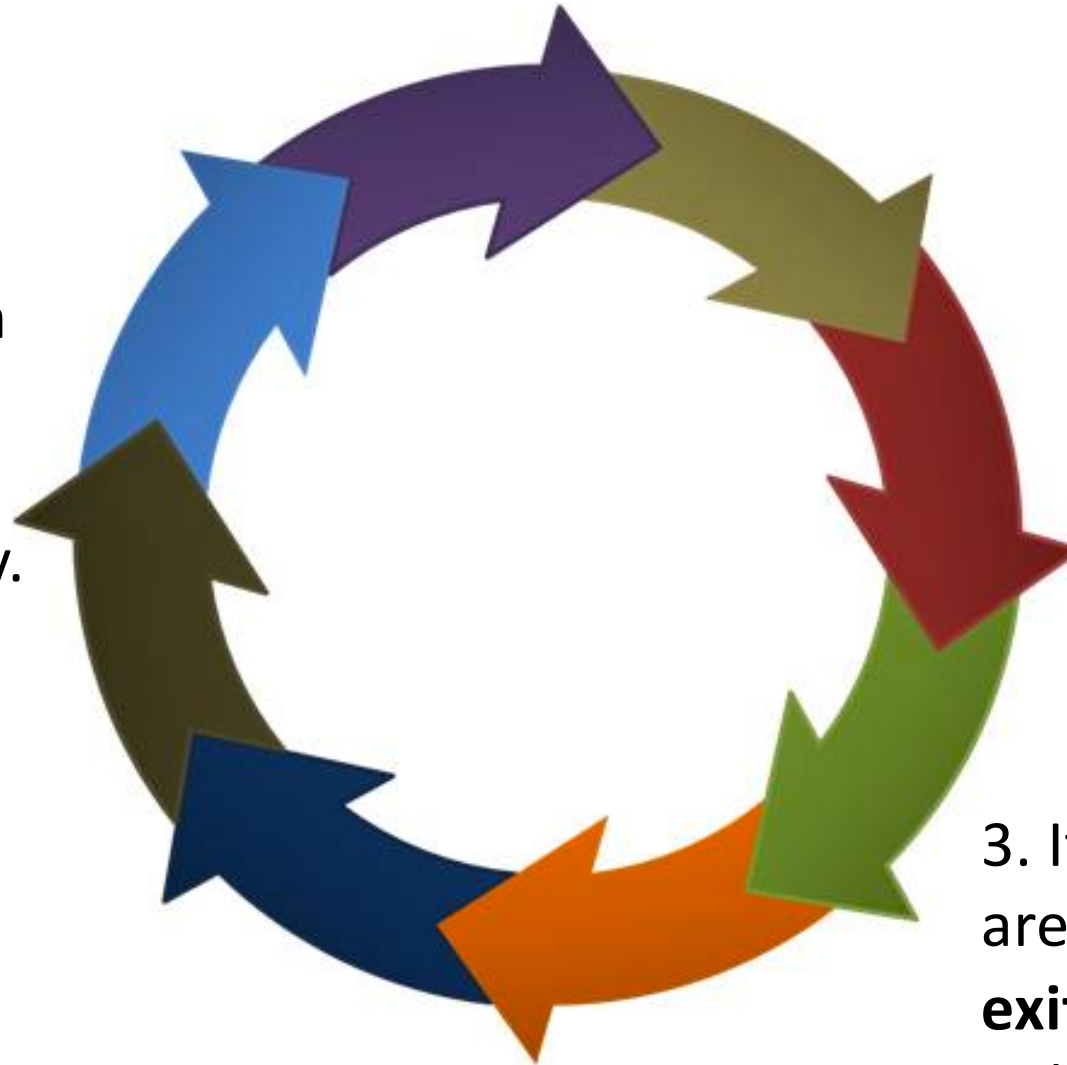
```
password = input("What is your pw?" )  
while password != secret:  
    print("Error! Wrong password!")  
    password = input("What is your pw? ")
```

1. Ask user for first name.

2. Ask user for second name.

3. If both names are STOP, then **exit the loop.** Otherwise, **keep going.**

4. Print which name comes first alphabetically.



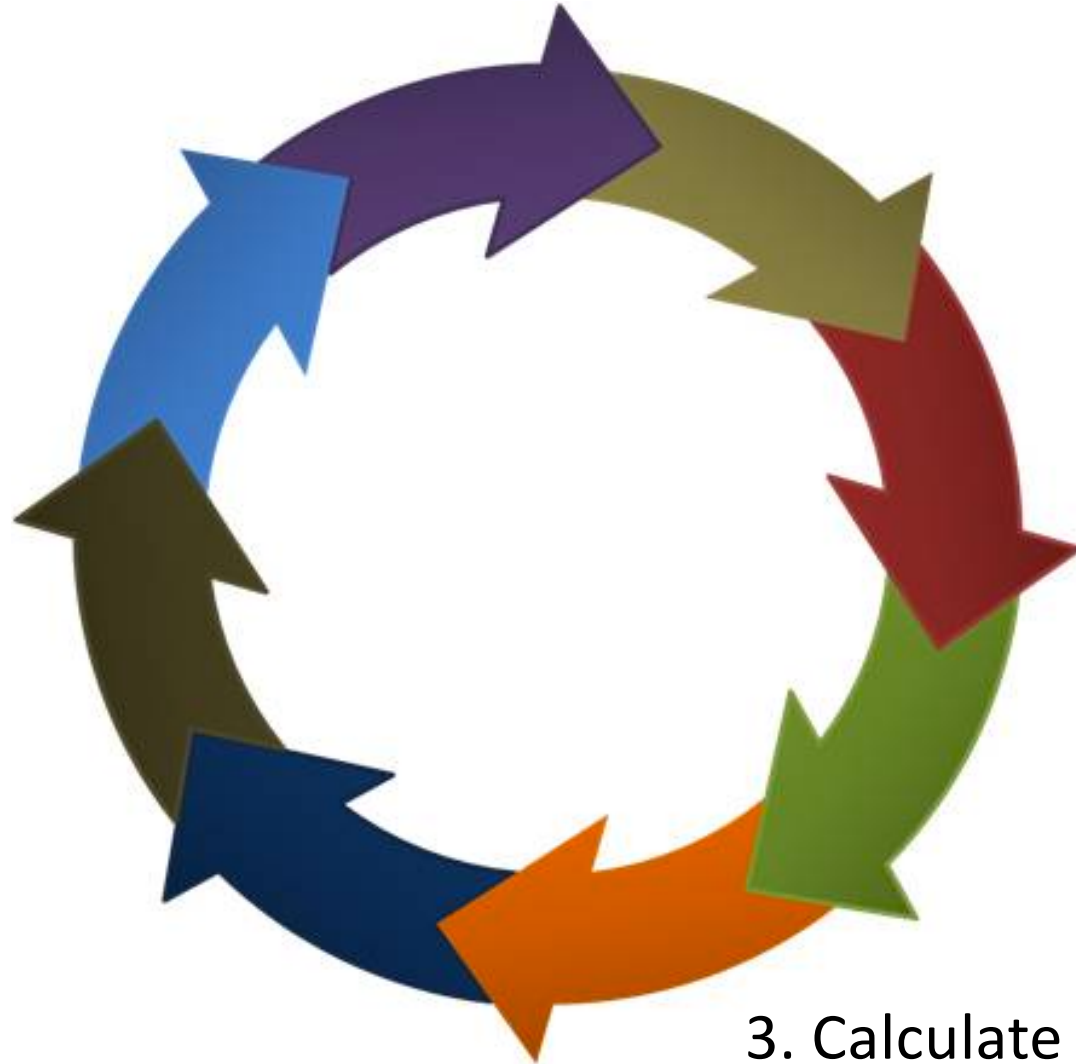
```
name1 = input("What is name #1? ")
name2 = input("What is name #2? ")
if not(name1 == "STOP" and name2 == "STOP"):
    if name1 < name2:
        print(name1, "is first.")
    else:
        print(name2, "is first.")
name1 = input("What is name #1? ")
name2 = input("What is name #2? ")
```

1. Ask user for envelope weight.

2. If weight is zero, exit loop. Otherwise, keep going.

3. Calculate postage.

4. Print postage.



```
weight = int(input("What is the wt? "))
if weight != 0:
    money = postage(weight)
    print("You must pay", money)
weight = int(input("What is the wt?"))
```