Winter Break! Python Bytes! Perfect Storm!

“Perfect Storm” is three coding projects that have nothing to do with each other… or do they?!?? Here are some notes on starting **Storm 1** and **Storm 2**.

Storm 1

Start with a string **`data`** and create a string **`newdata`**. We use a for-loop. We examine one character at a time. We follow the Rules below for each character. We start with an empty result: **`newdata = “”`** then add to it. Like **`newdata = newdata + “boopy”`**.

```
data = “x+y-x-x-x-y”
```

Rules

- “**`x`**” becomes “**`boopy`**”
- “**`y`**” becomes “**`xx+y`**”
- Everything else: Stays the same (just copy it)

Once we have **`newdata`** we **`print(newdata)`** to make sure it worked.

Then recreate the variable data like so:

```
data = newdata
```

Now we have reset **`data`** to be **`newdata`**. Go through **`data`** a second time, applying the same rules. Create a new version of **`newdata`**. Again **`print(newdata).`**

Then do this whole thing one more time.

Here is a peek ahead into the future: We will be doing something like this to generate a “program” for a turtle to follow. The turtle will then draw a picture. But first things first: Do the above procedure which I call “Project Boopy”.

Here is a starting point for writing the loop code:

```
for c in data:
    print(c)
    if c == “x”:
        print(“I found the letter x!”)
```

Storm 2

We have learned about logical operators AND, OR and Exclusive-OR. I will abbreviate the last one as XOR.

```
True  AND  True  = True
True  AND  False = False
False AND  True  = False
False AND  False = False

True  XOR  True  = False
True  XOR  False = True
False XOR  True  = True
False XOR  False = False

True   OR  True  = True
True   OR  False = True
False  OR  True  = True
False  OR  False = False
```

Activity Use XOR to fill in rows of the XOR Universe. Empty box = False, Dot = True.

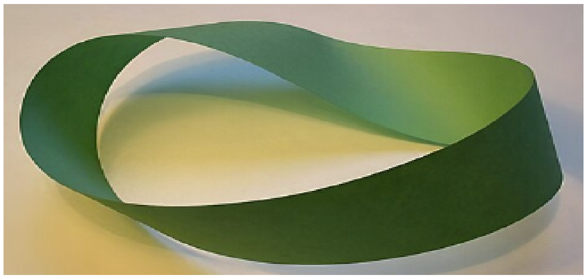
The very first row of the XOR Universe is all Empty Boxes except for one Dot in the middle of the row.

The activities from the first four Take Home Sheets are reprinted below, plus some solutions.

Activity: Predict out loud what these little bits of Python code will do. Bonus: Run them to test your predictions.

```
print(’harmony world’)

s = ’whirled peas’
print(s)

for c in ’abcdefg’:
    print(c)

for c in [1, 3, 5, 7, 9]:
    print(c+1)

a = 4
b = 5
c = 6
a = b
b = c
c = a
print(a, b, c)

# this comment is written in English
# this comment tries to print hello: print(’hello’)

a = 2
b = 3.2
print(a * b)
```

Second activity: Here is a picture of a Mobius strip. To make one: Take a strip of paper, give it a half-twist, and glue or tape the ends together to make this twisty loop.



If you pinch somewhere on the Mobius strip: Your fingers tell you it has two sides. If an ant walks along the mobius strip: How many sides will she say it has? How many edges?

To predict and then experiment: If you were to make **a slice in the center of the Mobius strip** (parallel to the edges); and then if you use scissors to cut all the way around its length: What will be the result? One loop? Two loops? Thinner Mobius strips? Interlocking? Something else???

Same question if you make the slice 1/3rd of the way across the width of the Mobius strip rather than ½ way; then continue this cut parallel to the nearby edge. What will be the end result?

Third activity: Farmer Diophantus has $100 and wants to buy exactly 100 animals for his farm. He must buy at least one **cow**, one **pig** and one **chicken**. Cows cost $10 each, pigs cost $3, and chickens cost $0.50. He must spend exactly $100… is this possible?

Fourth activity: You and your friend have a large supply of poker chips: Small circular disks all the same size. You have a large circular table at hand; and so your friend suggests playing a game: **You clear the table and then take turns placing poker chips one at a time.** The chips must be placed flat on the table; no stacking them, no leaning them; just place them flat on the table. The first player who has no place on the table to put their next poker chip loses. Your friend asks you if you want to go first or second. What strategy is sure to win? (You can place chips on the table with perfect precision.)

Fifth activity: Turtle time! If you have Turtle graphics available you can proceed to the code below. If you need a sandbox to code in: Go to https://www.pythonsandbox.com/turtle.

This code draws a picture. See if you can draw it: A *prediction*. Then run this code!

```
from turtle import Turtle
t = Turtle()
t.forward(100)
t.left(90)
t.pencolor(’red’)
t.forward(200)
t.home()
t.pencolor(’blue’)
t.left(90)
t.forward(200)
t.left(90)
t.up()
t.forward(30)
t.dot()
t.home()
```

Solutions

- Mobius cut in half the long way produces…
- Mobius cut in thirds produces: …
    - You might find it interesting to argue for why these cuts turn out as they do.
- Farmer Diophantus purchases 5 cows, 1 pig and 94 chickens. You might also conclude he purchases -5 cows, 39 pigs and 66 chickens… but what is a negative cow?
- Poker chips: You choose to go first and you place a poker chip in the center of the table. Your opponent places a chip somewhere. You respond by placing your next chip precisely opposite theirs across the table. You are certain to have this location open to you. Eventually your opponent runs out of locations to place a next poker chip.

Activity: Predict out loud what the code below does. Run the code to check!

```
my_message = ("sled" + " " + "dog" + " - ") * 5
my_message = my_message + "SLED"
print(my_message)

for c in "bank vault":
    print(c)

import random
roll_sum = 0
for dice_rolls in [1, 2, 3, 4, 5]:
    this_roll = random.randint(1, 6)
    print("you rolled a " + str(this_roll))
    roll_sum = roll_sum + this_roll
number_of_rolls = 5
average_roll = roll_sum / number_of_rolls
print("Your average dice roll was " + str(average_roll))

name_string = input("What is your name? ")
print("Hello " + name_string + "!!!")
name_reverse = name_string.reverse()
print("Your name backwards is " + name_reverse)

number_string = input("Enter a number: ")
number = int(number_string)
if number % 2 == 0: print(number, "is even")
else:               print(number, "is odd")
```

Activity: A king calls three wise guys to his court to choose a new advisor. They are forbidden to speak to one another as the king places a hat (either blue or white) on each of their heads: Each guy could see the other two hats but not his own. “**At least one of you wears a blue hat…**” says the king, meaning there could be one, two, or three blue hats, not zero, “**...and this contest is fair to all of you.**” The king then states that whichever person stands up first to correctly say the colour of his own hat: He would be the new advisor. The wise guys sit for a time; then one stands up and announces his hat color. What color? How does he know?

Activity: There is a peculiar island inhabited by two types of people: Knights and Knaves who are indistinguishable in appearance. However in response to a direct question a Knight will always tell the truth and a Knave will always lie. Suppose finding yourself on this island and walking along a path you see that just ahead the path branches into two paths. You have been reliably informed that one branch leads into a dangerous swamp full of vicious tigers, while the other leads to a safe and pleasant town on the coast. Standing guard just at the fork in the path you see a resident of the island (who knows which path is which). Can you ask them questions to determine which path is safe? Is it possible to ask only a single question?

Activity: David Hilbert owns a hotel that happens to have an infinite number of rooms. On one particular night every room is occupied by a guest and so the neon sign out front says “No Vacancy”. A weary traveler arrives at the hotel and enters the office. The traveler asks “Can you find me a room?” David immediately says “Yes of course. There are no empty rooms but fortunately there are infinitely many of them; so I can provide you with a room of your own directly.” What do you suppose David Hilbert does next? (Read the important detail below.)

An important detail: In his hotel office David Hilbert has a microphone. The microphone is connected to little speakers, one in each room of the hotel. By speaking into the microphone David Hilbert can make announcements that will be heard by each guest. If he makes a request of the guests: They will do their very best to do just as he says.

Activity: Two turtles! Turtle website: https://www.pythonsandbox.com/turtle.

```
from turtle import Turtle
t = Turtle()
u = Turtle()        # t and u are both turtles
t.left(60)
u.right(60)
t.forward(100)
u.forward(100)
t.right(120)
u.left(120)
t.forward(80)
u.forward(-80)
```

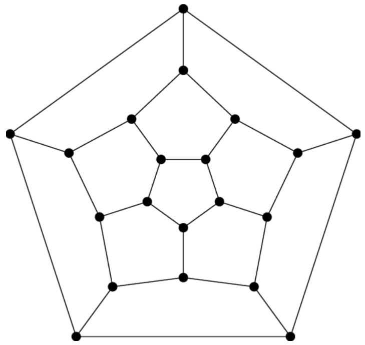
Solutions

- The key to the wise guys puzzle is that it is a ***fair challenge***. The only fair situation is one with an equal distribution of information; therefore all three hats must be blue.
- Determine ‘Knight or Knave’ for a resident by asking “Is the sky usually blue on a sunny day?” A Knave will answer “No” so next ask which path is safe and take the other. A one-question solution: “If I ask you which path is safe will you say it’s the left one?”
- David Hilbert announces over the microphone: “Everyone please move from your room to the next room up the hall. Guest in Room 1: Move to Room 2; room 2 move to room 3; and so on.” Now room 1 is vacant; available for the late-arriving person.

Activity: Predict out loud what the code below does. Run the code to check!

```
def add_them_up(n, a):
    if a == n:
        return a
    else:
        return add_them_up(n, a + 1) + a

n = int(input('Enter a positive number: '))
if n < 1 or n > 10000:
    print('something went wrong')
else:
    print('The sum is', add_them_up(n, 1))
```

Activity: Find a path following lines in this figure where you touch each dot just once and end back where you started. Bonus: How many regions does this figure have? Bonus: What might this figure look like in three dimensions?



For your information: Figures like this are called *graphs*. This one has the interesting feature that every region is surrounded by five lines and five dots. But… is the word 'surrounded' accurate for the 12th region?

Activity: Four identical bugs named **A**lpha, **B**ravo, **C**harlie and **D**elta stand at the four corners of a square table. **A** faces **B** who faces **C** who faces **D** who faces **A**. At a signal: Each bug begins walking towards the bug they face all at the same constant speed. Emphasis: This is a pursuit problem where the bugs can modify their walking direction but not their walking speed. Do the four bugs meet? If so: Where, and how far do they walk to get there?

Activity: David Hilbert owns a hotel with the peculiar feature: It has an infinite number of rooms. If you have not read about David Hilbert's Hotel: Do so now (see take home sheet number 2; the solution to that puzzle is given above). Now the new puzzle: A small bus pulls up to the Hilbert Hotel at 10PM. The neon sign still says "No Vacancy" due to the infinite number of guests in the infinite number of rooms. The driver says "I have 29 passengers plus me: We each need a room for the night." David Hilbert says "No problem!" Now: How does he accommodate these 30 sleepy people?

Activity: Turtle chase!! This code uses three new turtle functions. We have two turtles **r** and **q**. The first method is **r.towards(q)**. This gives us the heading angle from turtle **r** to turtle **q**. This angle will change as **r** and **q** move about on the screen. Next we have **r.setheading(h)** which points turtle **r** in heading direction **h**. We can then use **r.forward(10)** to move in that direction, the direction we are facing. Thirdly we can find out how far it is from turtle **r** to turtle **q** by using **r.distance(q)**.

```
from turtle import Turtle

q = Turtle()          # two turtles named q and r
r = Turtle()
q.penup()
q.goto(-300, -150)
q.setheading(20)
q.pendown()
q.pencolor("blue")
r.penup()
r.goto(-200, 250)
r.pendown()
r.pencolor("red")

for t in range(220):
    q.forward(3)
    r_heading_to_q = r.towards(q)
    r.setheading(r_heading_to_q)
    if r.distance(q) < 100: r.forward(3.9)  # speed up when close
    else:                   r.forward(3)
```

Solutions to previous puzzles

- The key to the wise guys puzzle is that it is a ***fair challenge***. The only fair situation is one with an equal distribution of information; therefore all three hats must be blue.
- Determine 'Knight or Knave' for a resident by asking "Is the sky usually blue on a sunny day?" A Knave will answer "No" so next ask which path is safe and take the other. A one-question solution: "If I ask you which path is safe will you say it's the left one?"
- David Hilbert announces over the microphone: "Everyone please move from your room to the next room up the hall. Guest in Room 1: Move to Room 2; room 2 move to room 3; and so on." Now room 1 is vacant; available for the late-arriving person.

Coding Projects Reminder

1. Solve Ms. Halfway. Hint: Use a loop and three variables.
2. Print the first 100 Fibonacci numbers. Hint: Use a loop and three variables.
3. Create a number-guessing game. Computer picks, User guesses.
4. Use a turtle to draw a spiral
5. Use four turtles to solve the Four Bugs Problem: If the square table has four sides each length 1 meter: How far do the bugs walk before they meet at the center?
6. Project Boopy