Claremont (AP): Beginning next week, CS 5 will be taught by a pair of penguins, announced an HMC dean. “Penguins are very smart,” said the dean, “and they are also quite adorable, which is more than we can say about the CS 5 faculty. Plus, we won’t need to heat their offices in the winter.”
This Week

Homework 2:
- Reading on AI and Jeopardy
- Lab: Fractal art
- Problem 2: Higher Order Functions
- Problem 3: Sequence Alignment
- Problem 4: RNA Folding

More practice with use-it-or-lose-it!

A note on Getting Help…

So many places to get help on CS5 homework!

And another note on Illness…

Please e-mail the prof as soon as you feel bad!

Comparing DNA via Longest Common Subsequence (LCS)

```python
>>> LCS("AGGACAT", "ATTACGAT")
5
>>> LCS("can", "man!")
2
```
Recursive Approach…

```python
def LCS(s1, s2):
    if BASE CASE:
        ???
    else:
        LCS("spam", "sam!")
```

Try this in your notes!

Solution follows

---

Turtle Graphics

Turtle graphics are built into Python!

```python
>>> import turtle
>>> turtle.forward(50)
>>> turtle.right(90)
>>> turtle.backward(50)
```

<table>
<thead>
<tr>
<th>degrees</th>
<th>radians</th>
<th>reset()</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear()</td>
<td>tracer()</td>
<td>forward()</td>
</tr>
<tr>
<td>backward()</td>
<td>left()</td>
<td>right()</td>
</tr>
<tr>
<td>put()</td>
<td>down()</td>
<td>width()/width()</td>
</tr>
<tr>
<td>pen()</td>
<td>fill()</td>
<td>heading()</td>
</tr>
<tr>
<td>pencolor()</td>
<td>penwidth()</td>
<td>pencolor()</td>
</tr>
<tr>
<td>pencol()</td>
<td>pensize()</td>
<td>pensize()</td>
</tr>
</tbody>
</table>

Problem 2 has a link to the turtle documentation

---

Fractals
“I Wonder About Trees” – Robert Frost

SnowFlake Fractals

The Koch Snowflake Fractal:

And now for something completely different...

(...or at least it will seem different until we see that it's not!)
Tuples ("Immutable Lists")

>>> foo = (42, 'hello', (5, 'spam'), 'penguin')
>>> foo
(42, 'hello', (5, 'spam'), 'penguin')
>>> foo[0]
42
>>> foo[-1]
'penguin'
>>> foo[0:2]
(42, 'hello')
>>> foo[0:1]
(42,)

Dictionaries

>>> D = {}
>>> D['Julie'] = "chocolate"
>>> D['Zach'] = "coffee"
>>> D['Alien'] = 42
>>> D['Julie']
'chocolate'
>>> D['Alien']
42
>>> D['Suicide Squad']
BARF!

"Julie", "Zach", and "Alien" are called the "keys" in the dictionary. Any immutable object can be a key.

Sometimes We Need to Make More Than 2 Recursive Calls!

Google maps

>>> FiveDists[ ('B', 'C') ]
42
Finding Shortest Paths

```python
>>> shortestPath (FiveCities, FiveDists)
10
>>> shortestPath ("C", "D", "E")
7
>>> shortestPath ("E")
0
```