Physics 40: Laboratory Sixteen

Thursday, May 21, 2020

Today's Goals:

Python scripts
More python syntax (loops, etc)

[1] It's awkward to use python in interactive mode unless you want to do something really simple (ie. just a few lines). For longer tasks one write a python 'script' (which is much like the C programs you have been creating) and then imports the script into python. This of course has the big advantage that you can make changes to, or re-use, the script without typing it all over again.

As a first, use an editor to create a file named 'ifelse.py' containing:

```
n = int(input("Please enter an integer: "))
if n%2 ==0:
    print('n is even')
else:
    print('n is odd')
```

Then (make sure you are in the same directory/folder where your file is) type:
python ./ifelse.py
The result should work precisely the same way as if you had written and compiled a C code. You will be queried for a value for the integer, and then python will tell you whether the integer is odd or even. Try it for a few different inputs.

[2] We'll now just write a bunch of scripts to learn how python does things.

Write this python script (you might call it 'fact.py')

```
N = int(input("Please enter an integer: "))
fact=1
for i in range (1,N+1):
    fact=fact*i
    print(fact)
```

Then execute it with python:
python ./fact.py
What does it do? If you run your script for N=30, does it still work?

[HW8-9] Write a C code to compute factorials making the variable storing the factorial an "int". (If you have trouble, look at HW8-10 below.) For which factorial does your code first break down? Why is the C code giving the wrong answer? For which factorial does your code first break down if you make the variable storing the factorial a "long int"? Compare the effectiveness of python and C at the task of computing factorials.

[HW8-10] Change your script to this alternate. The only difference is the indentation in the last line.

```
N = int(input("Please enter an integer: "))
fact=1
for i in range (1,N+1):
    fact=fact*i
print(fact)
```

What happens? Explain.

[HW8-11] This C code computes factorials in a very similar way to your original python factorial script:

```
#include <stdio.h>
#include <math.h>
int main(void)
{
int j,N,fact;
printf("Enter N");
printf("\n");
scanf("%d",&N);
fact=1;
for (j=1; j<N; j=j+1)
{
fact=fact*j;
printf("\n   %i     %i",j,fact);
}
printf("\n");
return 0;
}
```

How would you change it so that it behaves like your *second* python script?

[HW8-11] Change your script to this third version. Again, the only differences are indentations.

```
N = int(input("Please enter an integer: "))
fact=1
for i in range (1,N+1):
fact=fact*i
print(fact)
```

What happens? What mistake would give a similar error message when compiling a C code? Does C care about indentation?

[3.] Type in this python script. What does it compute?

```
N=input('Enter N:  ')
M=input('Enter M:  ')
Nfact=1
for i in range (1,N+1):
    Nfact=Nfact*i
Mfact=1
for i in range (1,M+1):
    Mfact=Mfact*i
NminusMfact=1
for i in range (1,N-M+1):
    NminusMfact=NminusMfact*i
result=Nfact/(Mfact*NminusMfact)
print('what is this mystery quantity?!  ')
print(result)
```

[4.] Type in this python script. What does it compute?

```
A=input('Enter A:  ')
x=1.
for i in range (1,10):
    x=x/2.+A/(2.*x)
    print(x)
```

[5.] Type in this python script. What does it compute?

```
A=input('Enter A:  ')
x=1.
for i in range (1,10):
    x=2.*x/3.+A/(3.*x*x)
    print(x)
```

[HW8-12] Write a python script which computes the sixth root of a number.

[6.] This script computes the 'Fibonacci numbers'. Type it in and run it.

```
N=input('Enter N  ')
fib1=1
fib2=1
for i in range (1,N):
    fib3=fib1+fib2
    print(fib3)
    fib1=fib2
    fib2=fib3
```

[HW8-13] Clearly explain the logic of the script in **[6.]**. What are the last two lines doing and why are they needed?

[HW8-14] The Fibonacci sequence appears in the book *Liber Abaci* (1202) by Fibonacci. Fibonacci considered the growth of an idealized (biologically unrealistic) rabbit population, assuming that: a newly born pair of rabbits, one male, one female, are put in a field; rabbits are able to mate at the age of one month so that at the end of its second month a female can produce another pair of rabbits; rabbits never die and a mating pair always produces one new pair (one male, one female) every month from the second month on. The puzzle that Fibonacci posed was: how many pairs will there be in one year?

Explain why your script is answering Fibonacci's puzzle.

According to wikipedia: Fibonacci numbers appear unexpectedly often in mathematics, so much so that there is an entire journal dedicated to their study, the *Fibonacci Quarterly*. Applications of Fibonacci numbers include computer algorithms such as the Fibonacci search technique and the Fibonacci heap data structure, and graphs called Fibonacci cubes used for interconnecting parallel and distributed systems. They also appear in biological settings, such as branching in trees, the arrangement of leaves on a stem, the fruit sprouts of a pineapple, the flowering of an artichoke, an uncurling fern and the arrangement of a pine cone's bracts.

Although named after an Italian mathematician, the sequence had been described earlier in Indian mathematics. More from wikipedia: The Fibonacci sequence appears in Indian mathematics in connection with Sanskrit prosody $\cdots$ In the Sanskrit poetic tradition, there was interest in enumerating all patterns of long (L) syllables of 2 units duration, juxtaposed with short (S) syllables of 1 unit duration. Counting the different patterns of successive L and S with a given total duration results in the Fibonacci numbers: the number of patterns of duration m units is F_{m+1}.

Knowledge of the Fibonacci sequence was also expressed as early as Pingala (c. 450 BC-200 BC). Singh cites Pingala's cryptic formula misrau cha ("the two are mixed") and scholars who interpret it in context as saying that the number of patterns for m beats (F_{m+1}) is obtained by adding one [S] to the F_m cases and one [L] to the F_{m1} cases.