NNSA HSAP Computational Laboratory 6 - 2026 Update

Goals: Further Coding Practice

[EX 6-01] Write a code which will print out the first 12 integers and their squares. That is, the output should be:

```
 1      1
 2      4
 3      9
 4     16
 5     25
 6     36
 7     49
 8     64
 9     81
10    100
11    121
12    144
```

[EX 6-02] Write a code which will read in the length and width of a rectangle and print out its area.

[EX 6-03] Write a code which will read in the base and height of a triangle and print out its area.

[EX 6-04] Write a code which will read in the radius of a circle and print out its area and its circumference.

[EX 6-05] This one is very hard! If you get stuck, skip to EX 6-06.

The value of π can be computed from

$$\pi = 4\left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots\right)$$

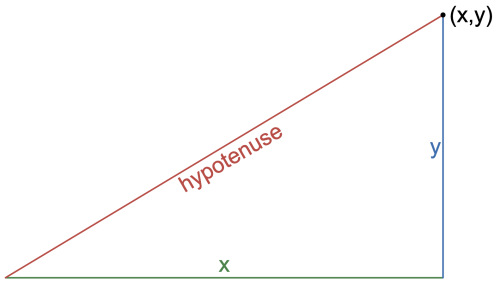
Write a code to do this sum, ending at $\frac{1}{101}$. How close do you get to the right answer $\pi = 3.1415926$? One tricky part is figuring out how to get the sign to alternate.

How close do you get to the right answer if you end at $\frac{1}{1001}$?

[EX 6-06] Write a program which asks the user to give the coordinates (x, y) of a point and then prints out the distance of the point from the origin. It is useful to remember things about how to take a square root from your quadratic equation python code last week:

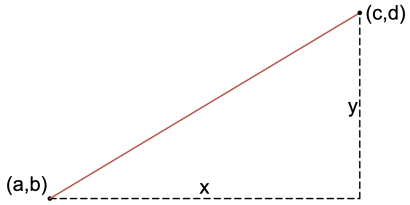
```
# First code to solve the quadratic equation.
# To use the sqrt function we need to import a library

import math

# Read in a,b,c

a = float(input("Enter a: "))
b = float(input("Enter b: "))
c = float(input("Enter c: "))

# Get the first root
root1 = ( -b + math.sqrt(b**2 - 4.0*a*c) ) / (2.0*a)
print("The first root is",root1)

# Get the second root
root2 = ( -b - math.sqrt(b**2 - 4.0*a*c) ) / (2.0*a)
print("The second root is",root2)
```

Remember also this picture below. The distance to the origin is the length of the red line. If the coordinates x and y are the lengths of the sides of the right triangle, what does the Pythagorean theorem tell you about the length of the red line (the hypotenuse)?



[EX 6-07] Write a program which asks the user to give the coordinates of two points (a, b) and (c, d) and then prints out the distance between them.

The picture below might be useful. What is the length of the horizontal dashed line and what is the length of the vertical dashed line? Those are the two sides of the right triangle. You again need the length of the hypotenuse (the red line).



[EX 6-08] You have two lines with equations

$$\begin{aligned} y &= a\,x + b \\ y &= c\,x + d \end{aligned}$$

Write a code which reads in the values of a, b, c, d and figures out where the lines intersect. Remember the math: The lines intesect when they have the same x and y values. Then

$$a\,x + b = c\,x + d$$

Solving for x:

$$x = \frac{d - b}{a - c}$$

You can plug this value of x into the equation for y (either one of them, you'll get the same answer!)

$$y = \frac{ad - bc}{a - c}$$

These are the x and y you want your code to print.

[EX 6-09] This is a continuation of the preceding problem. There is a case when your code from EX 6-08 will do an illegal math operation. When will that happen? When we wrote the code for the quadratic equation an illegal operation also occurred when we tried to compute $\sqrt{b^2 - 4ac}$ for $b^2 - 4ac < 0$. This was a warning that the parabola never crossed the x-axis. What is the illegal operation telling us geometrically here? (Hint: Do two lines *always* have to intersect?)

[EX 6-09] This one is also hard!

Write a code which reads in an integer n. Then if n is odd, it should change n to $3\,n+1$ but if n is even it should change n to $n/2$. Then repeat the process. Stop if you ever get to the value 1. Here's an example of what your code should do:

```
You read in n =  5.
Since n =  5 is  odd    you change n to n = 3*5+1 = 16
Now   n = 16 is even so you change n to n = 16/2  =  8
Now   n =  8 is even so you change n to n =  8/2  =  4
Now   n =  4 is even so you change n to n =  4/2  =  2
Now   n =  2 is even so you change n to n =  2/2  =  1
Since n =  1 the code stops.
```

Here's another (longer) example:

```
You read in n = 7.
Since n =  7 is odd     you change n to n = 3*7+1  = 22.
Now   n = 22 is even so you change n to n = 22/2   = 11.
Now   n = 11 is  odd so you change n to n = 3*11+1 = 34.
Now   n = 34 is even so you change n to n = 34/2   = 17.
Now   n = 17 is  odd so you change n to n = 3*17+1 = 52.
Now   n = 52 is even so you change n to n = 52/2   = 26.
Now   n = 26 is even so you change n to n = 26/2   = 13.
Now   n = 13 is  odd so you change n to n = 3*13+1 = 40.
Now   n = 40 is even so you change n to n = 40/2   = 20.
Now   n = 20 is even so you change n to n = 20/2   = 10.
Now   n = 10 is even so you change n to n = 10/2   = 5.
Now   n =  5 is  odd so you change n to n = 3*5+1  = 16
Now   n = 16 is even so you change n to n = 16/2   =  8
Now   n =  8 is even so you change n to n =  8/2   =  4
Now   n =  4 is even so you change n to n =  4/2   =  2
Now   n =  2 is even so you change n to n =  2/2   =  1
Since n =  1 the code stops.
```

It took a while, but we did end up at $n = 1$ again.

The Collatz conjecture says that no matter what n you start with, you always end up at $n = 1$ eventually.

The Collatz conjecture is unproven!

This is an unsolved problem in Mathematics!!!!