Part 1 — Python

1 Define a function `add(x,y)` that returns the sum, `x+y`. pyz012

2 Change the definition of `add(x,y)` so that if `y` is not given when the function is called, the function just returns `x+1`. Demonstrate it by adding 2 and 3 (using both arguments) and adding 1 to 2 (using just one). pyz013

3 Define a function `add` with two arguments, `x` and `y` that returns the sum `x+y`. The arguments have default values of 3 and 2, respectively. Then call the function with a keyword argument of 5 for `y` without specifying anything for `x` and print the result. pyz014

4 Change the definition of function `add` to include a documentation string. Demonstrate it by printing `add.__doc__`. pyz015

5 The *linear congruential generator* is a pseudorandom number generator. It is defined by the recurrence relation

$$x_{n+1} = (ax_n + c) \mod m, \qquad n \geq 0$$

where the four "magic numbers" are all integers and satisfy the following conditions.

m	the modulus	$m > 0$
a	the multiplier	$0 \leq a < m$
c	the increment	$0 \leq c < m$
x_0	the starting value (seed)	$0 \leq x_0 < m$

Example — $a = 17$, $c = 0$, $m = 100$, $x_0 = 13$ gives

i	0	1	2	3	4	5	6	7	8	9	10
x_i	13	21	57	69	73	41	97	49	33	61	37
i	11	12	13	14	15	16	17	18	19	20	
x_i	29	93	81	77	9	53	1	17	89	$\boxed{13}$	

As this example shows, the sequence is not always "random" for all choices of m, a, c, and x_0 because the sequence always eventually "gets into a loop", i.e., there is ultimately a cycle of numbers that is repeated endlessly. A "good" generator will deliver a large number of different values before it repeats itself. The repeating cycle is called the *period* of the generator. The example has period $p = 20$, i.e., 20 different values before repeating. A useful sequence will of course have a relatively long period. Clearly the maximum possible period is $\leq m$.

The linear congruential generator has full period m if and only if

(1) c and m have no common factors except 1, i.e., they are relatively prime;

(2) $(a-1)$ is a multiple of every prime number that divides m;

(3) $(a-1)$ is a multiple of 4 if m is a multiple of 4.

If $m = 2^k$ (a natural choice for computer calculations) we should take $a = 4d + 1$ where d is a positive integer. This choice obviously satisfies conditions (2) and (3); then (1) is satisfied by taking c as any odd number.

Task — Write a Python function `linear_congruential(a,c,m,seed,n)` to generate a list of n pseudo-random integers using the linear congruential generator as described above. You must include a docstring, some testing code and the output from several example runs (including the example above). pyz106

6 The Chevalier de Méré (1650s) won alot of money gambling on throwing

"at least one 6 after four throws of a single fair six-sided die."

Write Python code to count how many times this event occurs over n trials and print out the proportion of trials in which this event occurs. Use a suitable value of n in order to comment on whether the probability of this event occurring is greater than or less than $\frac{1}{2}$. Use the random seed of 123. pyz122

7 Write a function, `words(S)`, that takes a sentence in a string, `S`, and returns a list of words, each as a string.

Use your function, `words`, in a second function, `sortedwords(S)`, that takes a sentence in a string, `S` and returns a single string holding the words in sorted order, separated by spaces. Hint: refer to the list method `sort`, and the string method, `join`. pyz028

8 You are given a string containing a series of numbers separated by spaces. For example, the string might be assigned as `S = '23.5 34.6 77.9'`. Write code to calculate and print the sum of the numbers. Now write Python code in the form of a function `string2sum(S)`. Do not attempt to read the string in, just assign it. Test your function. pyz085

9 Study different random sampling functions in module `random` of package `numpy` then plot 50 different circles with random coordinates `(x, y)`, random `areas`, and random `colors`, in which `x` and `y` follow a uniform distribution and a normal distribution, respectively, while `areas` and `colors` are drawn from two discrete uniform distributions. Hint: study functions `uniform`, `normal`, and `randint`.

pyz136