1. What will Python return when you execute each of these:

```
print(3-2**-1)
print((3-2)**-1)
print(3-4/8-12*5+16//7)
print(3-4/(8-12*5)+16//7)
print(3-4/8-12*(5+16//7))
print((98%11)//3
```

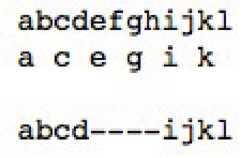
Check by asking Python to do it.

2. Unclear situation. The command `print(2**3**3)` will return an answer, but what is it the answer to?

3. *Integer division.* Instead of computing the quotient a/b of two integers a and $b \neq 0$ as a real number, we identify two integers q and r so that $a = bq+r$, and $0 \leq r < b$. For example, instead of writing $29/8 = 3.625$, we write $29 = 8 \cdot 3 + 5$.

Integer divide 736543/67, i.e., find q and r so that $736543 = 67q + r$, and $0 \leq r < 67$. Likewise, integer divide 9728234 by 1121.

4. Using `print` commands only, create the following output:



```
abcdefghijkl
a c e g i k

abcd----ijkl
```

5. Compute the following expressions: $i(2-i)(75i)$, $(1+i)^5$, $(2-4i)^{-2}$, and $7(2-i)^{4-3i}$.

6. Complete this code, which converts the temperature of C degrees Celsius to the temperature F in Fahrenheit (and calculates it when `C=21`):

```
C=21
F=
print(F)
```

[Find the conversion formula on internet]

7. Which of the following expressions will make Python unhappy, i.e., will cause a syntax error? Give a reason.

(a) `2-(3*(4//5)))`

(b) `2-(3*(4//5))))`

(c) `7.5 = c`

(d) `8 = a+3`

(e) `x = x+1`

(f) `a+=13`

8. Environment Canada uses the following formula to calculate the wind chill index:

$$W = 13.12 + 0.6215T - 11.37v^{0.16} + 0.3965Tv^{0.16}$$

where T is the air temperature (in degrees Celsius) and v be the wind speed, in km/h. Write a code to calculate W.

9. If mysentence="Assume that f(x) is a continuous function." say what each command will do

```
print(mysentence[0], mysentence[4], mysentence[11])
print(mysentence[1], mysentence[-1], mysentence[-11])
print(mysentence[:7])
print(mysentence[:16:3])
print(mysentence[::5])
print(mysentence[:-12])
print(mysentence[-16:])
print(mysentence[-12::2])
print(mysentence[-12::-2])
print(mysentence[-11:-7])
print(mysentence[-7:-11])
```

Check your answers by asking Python to do it.

10. Given the string `words='Python programming language'` state which command generated each output.

(a) `mming languag`

(b) `Ph oaing`

(c) `egaugnal`

(d) `gimropn`

11. Given a string, write a code which prints it in reverse. For example, given the string "Sunshine Today" your code needs to produce "yadoT enihsnuS"

12. If `town1='Halifax NS'` and `town2='Toronto ON'` what is the output of each command:

```
print(Halifax in town1)
print('Halifax' in town1)
print('TorontoON' in town2)
print(town1+town2)
print(town2+town1)
print(town1*3)
print(town2+2*town1)
print(town1+'and'+town2)
print(town1+' and '+town2)
```

13. Given is a string S. Write a code to create a string which consists of S followed by S spelled backward. For instance, if S='sunny' the output should read 'sunnyynnus'.

14. If `Alist=[9,-4,33,2,89,100,36,11]`, what is `Alist[2]`? What is `Alist[2][1]`?

15. If `Blist=[True,False,True,True,False,True,True,True]`, what is `Blist[3]`? What is `Blist[3][1]`?

16. If `Dlist=[[9,-4],[33,2],[89,100],[36,11],[0,6],[1,-2]]`, what is `Dlist[4]`? What is `Dlist[4][1]`?

17. If `Flist=[[True,False,True,False],[False,True],[True,True]]`, what is `Flist[2]`? What is `Flist[1][0]`? What is `Flist[0][1]`?

18. Write the matrix

$$M = \begin{bmatrix} 1 & 21 \\ 3 & 16 \\ 5 & 43 \end{bmatrix}$$

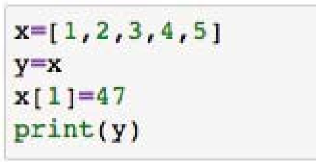
as: (a) a list whose elements are the rows of M, and (b) a list whose elements are the columns of M.

19. Let `Clist=[19,-2,4,3,-2,-2,1,9]`. What is the output of each command?

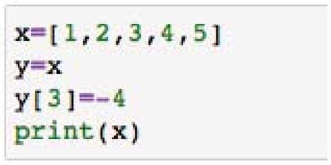
```
print(Clist[4])
print(Clist[8])
print(Clist[:4])
print(Clist[:4:1])
print(Clist[3::1])
print(Clist[3:9:1])
print(Clist[3::2])
print(Clist[3::3])
print(Clist[-4:-4])
print(Clist[-4:-1])
print(Clist[-3:])
print(Clist[:-3])
print(Clist[::-1])
print(Clist[-1:6:-1])
print(Clist[-9:6:-1])
print(Clist[-4:-8:-1])
print(Clist[-3::-2])
print(Clist[3::-2])
print(Clist[:-3:-2])
print(Clist[3::-2])
```

Ask Python to check your answers.

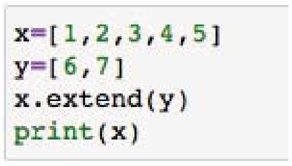
20. In each case, say what the output is, and then check using Python.

A


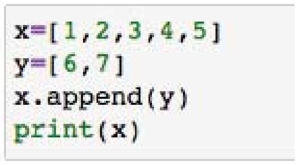
```
x=[1,2,3,4,5]
y=x
x[1]=47
print(y)
```

B


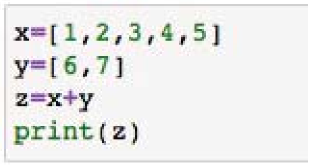
```
x=[1,2,3,4,5]
y=x
y[3]=-4
print(x)
```

C


```
x=[1,2,3,4,5]
y=[6,7]
x.extend(y)
print(x)
```

D


```
x=[1,2,3,4,5]
y=[6,7]
x.append(y)
print(x)
```

E


```
x=[1,2,3,4,5]
y=[6,7]
z=x+y
print(z)
```

21. Write a code that does what .append does for lists. I.e., given a list (call it List1) and a value A, insert the value A at the end of List1.

22. Write a code that does what .insert does for lists. I.e., given a list (call it List2) and a location within the list (call it insert), write a code that inserts an element A into List2 at the location insert.

23. Write a code that does what .pop does for lists. I.e., given a list (call it List3) and a location within the list (call it delete), write a code that removes the element at the location delete from List3.

24. Write the code for this: input is a real number a. Output: if a is negative (smaller than zero), then print 'the number is negative'; if a is equal to zero or smaller than 4, then print 'the number is zero or smaller than 4'; otherwise, print 'the number is larger than or equal to 4'.

25. Write the code for this: input is a positive integer m. Output: if m is divisible by 4 but not divisible by 3, print 'the number is divisible by 4 but not divisible by 3'; if m is divisible by 3 but not divisible by 4, print 'the number is divisible by 3 but not divisible by 4'; if m is divisible by both 3 and 4, print 'the number is divisible by both 3 and 4'; otherwise, print 'the number is not divisible by 3 and not divisible by 4'.

26. Given are three distinct real numbers a, b, and c. Write a code (use if statement(s)) that outputs the largest of the three numbers.

27. Given are three real numbers a, b, and c (not necessarily distinct, so for instance, a and c could be equal). Write a code (use if statement(s)) that outputs the largest of the three numbers.

28. Write a code to determine whether a given year is a leap year or not. Input: a year between 1600 and 2200. Output: Year 1997 is not a leap year. Year 2004 is a leap year. etc.

Note: years divisible by 4 are leap years, except if they are divisible by 100 but not divisible by 400. Thus, 1700, 1800, 1900, 2100 are not leap years, but 2000, 2400 are leap years.

29. You are speeding on a highway and a police officer stops you. Write a code to compute the fine that you might have to pay (call the variable fine), based on the following rules (assume that the speed is a positive integer):

(i) If your speed is 110 km/h or less, the fine is zero.

(ii) If your speed is between 111 and 130 inclusive, the fine is $150.

(iii) If your speed is above 130 km/h, the fine is $350.

(iv) If it is your birthday, your speed can be 5 km/h higher in all cases (use your actual birthday, or make one up)

For instance, if you drive 114 km/h on your birthday, your fine is zero. If you drive 118 km/h on your birthday, your fine $150. If you drive 114 km/h and it's not your birthday, your fine is $150.

30. Given a list (call it `temp`) and a value `A`, count how many times the value `A` appears in the list `temp`.

31. The list `pos` contains positive integers. Determine how many odd numbers are there in `pos`.

32. Write a code to find $\sum_{i=1}^{100} i^2$. Use the formula $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$ to check your output.

33. Write a code to find $\sum_{j=1}^{10{,}000} \frac{1}{j}$.

34. Find the sum of all positive integers from 1 to 10000 which are not divisible by 4 or not divisible by 5 (or is *inclusive or*: thus, "not divisible by 4 or not divisible by 5" includes, for instance, 16, 25, and 27)

35. Create a list of squares of all numbers x, where $1 \leq x \leq 20,000$, and x is divisible by 7.

36. Given are two lists `L1` and `L2` of equal length. Assume that their entries are real numbers. Create a list `L3`, defined as: `L3[i]`$=0$, if `L1[i]` is equal to `L2[i]`; `L3[i]`$=1$, if `L1[i]` is larger than `L2[i]`; and `L3[i]`$=-1$, if `L1[i]` is smaller than `L2[i]`.

37. Given are two lists `L1` and `L2` of equal length whose entries are boolean. Create a list `L3`, defined as: `L3[i] = True` if `L1[i]` and `L2[i]` are both `True` or both `False`; and `L3[i] = False` otherwise.

38. First, create a list of squares of numbers x, where $1 \leq x \leq 100,000$. Then, remove all entries which start with 9 and end with 0 or 1.

39. Write a code which computes the sum $S = \sum_{i^*=1}^{10000} \frac{1}{i^*}$ where i^* is a number which is written without the digit 7, i.e.,

$$S = \frac{1}{1} + \cdots + \frac{1}{6} + \frac{1}{8} + \cdots + \frac{1}{16} + \frac{1}{18} + \cdots + \frac{1}{69} + \frac{1}{80} + \cdots + \frac{1}{10000}$$

40. Given is a list of N real numbers x_i. Compute the mean and standard deviation of this list. Recall that

$$\mu = \frac{\sum x_i}{N} \qquad \text{and} \qquad \sigma^2 = \frac{\sum (x_i - \mu)^2}{N}$$

41. Given a non-empty list of numbers (call in `num`), write a code to find the largest and the smallest numbers in `num`. Then compare with the built-in functions max(`num`) and min(`num`).

42. Write a code to find $\sum_{m=1}^{600} \sum_{n=10}^{450} \frac{m}{n}$.

43. Print "I love math" once, then "I hate math" once; then "I love math" two times, then "I hate math" once; then "I love math" three times, then "I hate math" once; then "I love math" four times, then "I hate math" once, and so on, until a total of 100 lines are printed.

44. Create a list A so what $A[i]$ =sum of digits of i, for $0 \leq i \leq 10000$, i.e.,

$$A = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 1, 2, 3, 4, \ldots, 1]$$

45. Assume that A is a 3×3 matrix entered as a list in Python. I.e., if

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

then `A=[[`a_{11}, a_{12}, a_{13}`],[`a_{21}, a_{22}, a_{23}`],[`a_{31}, a_{32}, a_{33}`]]`. Write a code that generates a matrix B (as a list) whose entries are given by $b_{ij} = a_{ij}^3 - 7$, for $1 \leq i, j \leq 3$.

46. Assume that A is a 3×3 matrix entered as a list in Python, as explained in the previous exercise. Write a code that identifies the largest entry in A and counts how many times it appears in A.

47. Assume that A and B are 3×3 matrices defined as lists in Python. Write a code that computes the sum $A + B$ and the value of the matrix expression $5A - 12B$.

48. Define the function is_prime(p) which returns True if p is a prime number and False otherwise. You can assume that p is an integer, $p \geq 1$.

49. Create a list of the first 100 prime numbers.

50. Define the function all_divisors(n), where $n > 0$, which stores all divisors of n (including 1 and n) into a list.

51. Write a function convert(x) which converts the given speed x in km/h into m/s (metres per second) and mph (miles per hour).

52. Write a function Check_Pyth(a,b,c) that checks if the three numbers form the Pythagorean triplet, i.e., can be the sides and the hypothenuse of a right triangle. For instance, Check_Pyth(5,3,4) = True, and Check_Pyth(1,2,3) = False.

53. A 3×3 matrix M is given as a list with three elements, each of which is one row in M; i.e., `M=[[1,2,3],[4,5,6],[7,8,9]]` represents the matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Write a function that computes the transpose of M (call it Mt) and prints it in the same format as M.

54. Given a text (as a string), print the frequency of appearance of each of the vowels (a, e, i, o, u) in the string.

55. Find the sum of all digits used in writing down the numbers from 1 to 2 million (including 1 and including 2 million).

56. Create a list with the following pattern:

$$1, 2, 4, 5, 7, 8, 10, 11, 14, 16, 17, 19, 20, 21, 22, 25, \ldots$$

i.e., write natural numbers in a sequence and remove all numbers which are either divisible by 3 or contain the digit 3. Find the 10,000th number in this list.

57. Let $M = 123456\ldots99979998999910000$ (i.e., M is obtained by writing all numbers from 1 to 10000 next to each other)

(a) How many digits does M have?

(b) How many times does the number 7 appear in M?

58. Let $N = 124567891011121415\ldots282940414244\ldots999799989999$ (i.e., N is obtained by writing all numbers from 1 to 9999 that do not contain digit 3).

(a) How many digits does N have?

(b) Is N divisible by 3?

59. The Fibonacci sequence is defined by $F_1 = 1$, $F_2 = 1$, $F_{n+2} = F_n + F_{n+1}$. Compute the sum

$$S = \sum_{n=2}^{1000} \frac{F_n}{F_{n+1}F_{n-1}}$$

60. The Fibonacci sequence is defined by $F_1 = 1$, $F_2 = 1$, $F_{n+2} = F_n + F_{n+1}$.

(a) Create a list (call it Fib) which contains the first 10,000 Fibonacci numbers, so that Fib[0] $= F_1 = 1$, Fib[1] $= F_2 = 1$, Fib[2] $= F_3 = 2$, and so on.

(b) Create the list L, defined by $L[i] =$ last digit of Fib[i].

(c) Create the list N, defined by $N[i] =$ number of digits of Fib[i].

(d) Create the list D, defined by $D[i]$ = the difference between Fib$[i+1]$ and Fib$[i]$.

(e) Experiment! Think of something fun to do with the Fibonacci sequence (that you would not be able to do using pencil and paper).

61. Assume that a vector $\mathbf{a} = (a_1, a_2, a_3)$ in $\mathbb{R}^3$ is given as a list $[a_1, a_2, a_3]$. Write a code for the function:

(a) dot_orthogonal(v,w) which returns the dot product of v and w, and also a Boolean variable which is set to True if v and w are orthogonal, and False otherwise.

(b) cross_product(v,w) which returns the cross product of v and w.

(c)[optional] projection_vector(v,w) which returns the vector projection of v onto w. (Find a formula in your linear algebra textbook, or derive it.)

62. Compute the sum of all entries in an $m \times n$ matrix M, which is given as a list $M = [r_1, r_2, \ldots r_m]$, where the list r_i is the ith row of a matrix. Test your code on 1×2, 3×1, and 3×2 matrices.

63. Write a function replace_entry(M,find_this,replace_with) which replaces all entries equal to find_this in an $m \times n$ matrix M with the value replace_with.

64. Write a code that finds the largest entry (maximum) in an $m \times n$ matrix M, together with its location (row and column). Note that a maximum entry can appear in several places, and you need to find them all.

65. A 3×3 matrix M is given as a list with three elements, each of which is one row in M; i.e., `M=[[1,2,3],[4,5,6],[7,8,9]]` represents the matrix

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Write a code (i.e., do not use ready-made commands) that

(a) Computes the sum of diagonal terms.

(b) Computes the sum of off-diagonal terms.

(c) Adds two 3×3 matrices M and N.

(d)[optional] Multiplies two 3×3 matrices M and N.

66. Write a code that extracts the main diagonal (as a list) from a square matrix M.

67. Using list comprehension, create the following lists:

(a) L1 $= [1, 3, 5, 7, \ldots, 1001]$

(b) L2 $= [1, 1/3, 1/3^2, 1/3^3, \ldots, 1/3^{43}]$

(c) L3 is the list of the fourth powers of all numbers in a given list, call it M. I.e., if M $= [4, 3, 2.7, 4.5, 7, 17]$ then L3 $= [4^4, 3^4, 2.7^4, 4.5^4, 7^4, 17^4]$.

(d) L4 is the list of the fourth powers of every third number (starting from the first) in a given list, call it M. I.e., if M $= [4, 3, 2.7, 4.5, 7, 17, 2]$ then L4 $= [4^4, 4.5^4, 2^4]$

68. Using list comprehension, create the following lists:

(a) List L1 of all positive integers between (and including) 100 and 1000 which are divisible by 12.

(b) List L2 of all positive integers between (and including) 100 and 1000 which are divisible by 11 and divisible by 12.

(c) List L3 of all positive integers between (and including) 100 and 1000 which are divisible by 11 but not divisible by 12.

69. Given is the list L=['Iron Man', 'Hulk', 'Thor', 'Ant-Man', 'Wasp', 'Hawkeye', 'Quicksilver', 'Scarlet Witch', 'Captain America', 'Goliath', 'Hercules', 'Black Widow'] create the following lists, using list comprehension:

(a) L1 is a list of all names from L which are longer than 6 characters (blank (space) characters count).

(b) L2 is a list of all names from L which start on 'A' or 'B.'

(c) L3 is a list of all names from L which do not contain the string 'as'

(d) L4 is a list of all names from L which end on 'r.'

70. The coefficients of a polynomial $a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$ are given as a list pol=$[a_0, a_1, a_2, \ldots, a_n]$. For instance, $6 - 5x + x^3$ is represented as $[6, -5, 0, 1]$ Write a function evaluate_polynomial(pol,a) which will find the value of the polynomial pol at a (float) value a. Thus, for the above polynomial, the call evaluate_polynomial(pol,2) should return 4.

71. Two prime numbers p and q ($p < q$) are said to form a gap if all numbers between them are composite (i.e., not prime; in other words, there are no prime numbers between p and q). The size of the gap, $q - p - 1$, is the number of composite numbers between p and q.

Identify the largest gap (what it is, how many times it occurs, and where) in the prime numbers up to 10^6. (Use the command discussed in class to create the list of primes that you need.)

72. Given is a list A of length 1000 whose entries are either 0 or 1. Create a list B according to the following rules: The first and the last entries in B are 0. For $i = 1, \dots 998$, the i-th entry in B is defined as follows: if $A[i] = 0$ and at least one of its neighbours (i.e., the numbers to the left and to the right of it) is 0 then the $B[i] = 0$. Otherwise, $B[i] = 1$. If $A[i] = 1$ and at least one of its neighbours is 1 then the $B[i] = 1$. Otherwise, $B[i] = 0$. (Of course, test you code on smaller lists.)

73. A bill was found, stating: 72 turkeys _67.9_ dollars (the first and the last digit have fadded and are illegible). Write a code to figure out the price of one turkey.

74. Write a function convert_distance(d,source,target) that will convert a given distance d (float) from one of the source units (m, km) to one of the target units (inch (in), foot (ft), mile (mi)). For instance, the function call convert_distance(4.5,'km','mi') should convert 4.5 kilometres into miles.

75. Write a function convert_mass(m,source,target) that converts a given mass m (float) from one of the source units (mg, g, kg, metric ton (t), ounce (oz), pound (lb), stone(st)) to one of the target units (same as source). I.e., the call convert_mass(2.38,'t','st') should convert 2.38 metric tons into stones. Consult internet for conversion factors.

Before you start, think a bit about how you plan to do this. There are 7 source and 7 target units, so there are 49 cases (or 42, if you eliminate the "conversion" from a unit into itself). Will you really write 42 if statements?

76. A primitive, but sometimes useful, way of approximately solving an equation is to use the so-called *grid method*. Assume that you need to find all x such that $f(x) = 0$, where x is in some predetermined (or given) interval $[a, b]$. The method goes like this: divide the interval $[a, b]$ into n equally spaced points, so that they are $(b-a)/n$ apart from their neighbours. Check the values of f at these points, and identify all those points at which $|f(x)| < \text{tol} = 10^{-6}$ (or some other, small number). This means that the value of $f(x)$ at these points is very close to zero, so they are potential candidates for approximations of the roots of $f(x)$. (Of course, just because a function has a very small value at some point, does not mean that that point is its root. But it helps if you combine this with IVT.)

Use the grid method to find approximations of solution(s) of

$$f(x) = 3\cos(3.4x) - e^{0.45x} - x + 2.2 = 0$$

on $[-2, 4]$, with $n = 1000$, and then with $n = 10000$. For tolerance tol, use 10^{-6}, and also 10^{-10}. You output should consist of the values of x and the corresponding values $f(x)$.

77. Use the grid method (see the previous exercise) to find approximations of solution(s) of

$$f(x) = x^3 - x^2 - 2x + 3\sin(2x) = 0$$

on $[-1, 3]$, with $n = 1000$, and then with $n = 10000$. For tolerance tol, use 10^{-6}, and also 10^{-10}. You output should consist of the values of x and the corresponding values $f(x)$.

78. Use the bisection method to find approximations (correct to five decimal places) of the solutions of

$$f(x) = 3\cos(3.4x) - e^{0.45x} - x + 2.2 = 0$$

on $[-2, 4]$, and

$$f(x) = x^3 - x^2 - 2x + 3\sin(2x) = 0$$

on $[-1, 3]$.

79. Write a function Newton(f,fprime,x0,tol,max_steps) which produces a list of approximations (starting with x0) of a solution to $f(x) = 0$ using Newton's method. Stop the process when either the number of steps reaches max_steps or the value of f is less than tol.

Recall that Newton's method gives a sequence of approximations

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

which, for a differentiable function, approximate a solution $f(x) = 0$ (if x_0 is close to it).

Test your code by solving

$$f(x) = x^3 - x^2 - 2x + 3\sin(2x) = 0$$

on $[-1, 3]$.

80. (Collatz sequence) Define the function

$$f(n) = \begin{cases} n/2 & \text{if } n \text{ is even} \\ 3n+1 & \text{if } n \text{ is odd} \end{cases}$$

where $n \geq 1$ is an integer. Write a code for the following.

(a) Let $c[N]$ = number of iterations of f needed to reach 1, starting from N. Write a code that places the values of $c[N]$ for $N = 1$ to $N = 100,000$ into a list.

(b) What are the five largest values of $c[N]$ for N between 1 and $100,000$, and for which N do they occur?

(c) How many numbers between 1 and $100,000$ require exactly 113 steps to reach 1?

(d) Identify the longest constant sequence: i.e., find N and k so that $c[N] = c[N+1] = c[N+2] = \cdots = c[N+k]$ and k is the largest number with this property.

(e) Including N and 1, how many numbers in the Collatz sequence from are even and how many are odd? (example: if $N = 10$, then there are 5 even and 2 odd numbers in the Collatz sequence). If correct, your answer will show that there are more even numbers in the sequence. That seems to be the case in general, for any number $N \geq 3$. Find a reason why it is so.

81. Assume that $x \in \mathbb{R}$ represents an angle in radians. Write a function that reduces this angle to an angle in $[0, 2\pi)$. For instance, if $x = 3\pi$, then it reduces to π. The angle of $x = 2\pi$ reduces to 0, and if $x = 9$, then it reduces to $9 - 2\pi$.

82. Write a code for the function cos_series(x) which calculates an approximate value of $\cos x$, where $x \in \mathbb{R}$, based on the formula

$$\cos x \approx \sum_{i=0}^{24} \frac{x^{2i}}{(2i)!}$$

If x is not in $[0, 2\pi)$, reduce it to that interval (see the previous exercise).

83. Write a code for the function cos_series(x,tol=1e-7,max_terms=15) which calculates an approximate value of $\cos x$, where $x \in [0, 2\pi)$.

Use the formula for the approximation of $\cos x$ from the previous exercise and end it, depending on which of the following two events happens first: 15 terms are added, or the next term to be added is less than the tolerance tol.

84. Determine the number of terms in the sum
$$s_n = \sum_{i=1}^{n} \frac{1}{i^2}$$
which are needed to approximate $\pi^2/6$ with an error less than 10^{-6}. (Recall that the above sum approximates $\pi^2/6$.) In other words, find the smallest n so that
$$\left| s_n - \frac{\pi^2}{6} \right| < 10^{-6}.$$

85. Write a code for the function exp__sum(x,n) that, for a given real number x and a positive integer n, calculates the sum
$$\sum_{n=0}^{n} \frac{x^i}{i!}$$
Compare the value of the sum with math.exp(x).

86. Write a code for the function derivative(f,a,h=1e-8) that, for a function $f(x)$ (write a function to define some function, say $f(x) = e^{\sin x}$) approximates the derivative $f'(a)$ at a using the difference quotient $(f(a+h) - f(a))/h$.

87. Write a code which, for a given function $f(x)$, returns an approximation of its derivative $f'(x)$, obtained by using the difference quotient as in the previous exercise.

88. Write a function called single__appearance(numbers) that takes a list and returns a new list (call it single) which contains all elements in numbers that appear only once.

For example: single__appearance([1,2,3,4,5,2,3,4,5,6,2,8]) should return [1,6,8].

89. From a finite subset $S \subseteq \mathbb{R}$ remove all elements which are larger than M.

90. Given is a text as a string: words are separated by a single space, or a comma and single space. Sentences end with a full stop (.) and a space. For example: text='This stuff about Earth is interesting. On Earth this process seems to have taken place quite suddenly about nine hundred million years ago. At that time the Earth was already almost four billion years old, and had been inhabited by some simple lifeforms such as bacteria for much of that time.'

Chop the string text up to form a list of unique words that appear in it.

91. The symmetric difference of sets A and B is the set of all elements which are in the union $A \cup B$, but not in their intersection. Write a function symmetric__difference(A,B) which returns the symmetric difference of two sets A and B.

92. Write a function to perform each of these operations on sets:

(a) complement(A,U); computes the complement of the set $A \subseteq U$. Make use to include options $A = \emptyset$ and $A = U$.

(b) difference(A,B); computes the difference $A \setminus B$.

(c) Partition(X,A,B,C); returns True if A, B and C form a partition of X, i.e., their union is X and they are pairwise disjoint.

93. Create the Cartesian product C of two sets A and B. I.e., C is a set which contains all ordered pairs (a, b) where $a \in A$ and $b \in B$.

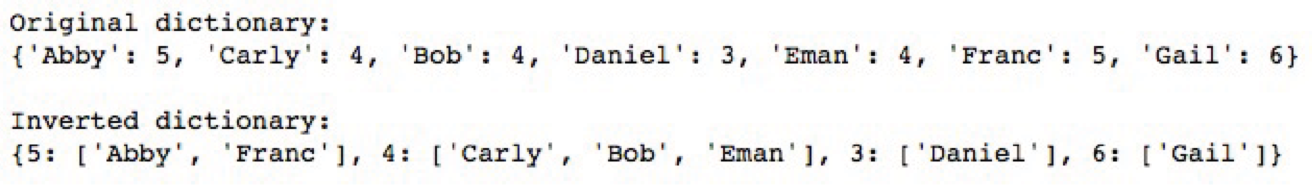
94. Given is a finite set of points $P \subseteq \mathbb{R}^2$ (a point is given as an ordered pair). Create a new set Q which contains all points of P which are in the third quadrant and lie outside of the circle $x^2 + y^2 = 21$. (Assume that a quadrant is closed, i.e., contains the coordinate axes.)

95. Assume that d1 and d2 are two dictionaries with disjoint keys. Write a code (i.e., do not use ready-made commands) to merge the two dictionaries into a new dictionary, called d3.

96. Assume that d1 and d2 are two dictionaries which contain some keys in common. Write a code (i.e., do not use ready-made commands) to merge the two dictionaries into a new dictionary, called d3, in such a way that the entries in d1 and d2 which have the same key are not included in d3.

97. Assume that d1 and d2 are two dictionaries which contain some keys in common. Write a code (i.e., do not use ready-made commands) to merge the two dictionaries into a new dictionary, called d3, in such a way that a key common to both d1 and d2 points to the value which is a list, and which contains both values from d1 and d2.

98. Invert a dictionary, i.e., create a new dictionary with keys and values switched. For all multiple values in the original dictionary, your inverted dictionary should contain a list of all keys from the original directory, as in

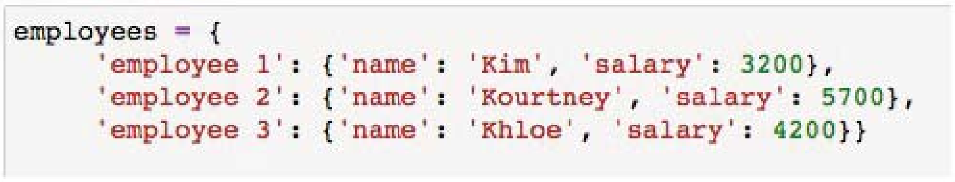


```
Original dictionary:
{'Abby': 5, 'Carly': 4, 'Bob': 4, 'Daniel': 3, 'Eman': 4, 'Franc': 5, 'Gail': 6}

Inverted dictionary:
{5: ['Abby', 'Franc'], 4: ['Carly', 'Bob', 'Eman'], 3: ['Daniel'], 6: ['Gail']}
```

99. Given a dictionary d, create a list of distinct values of d.

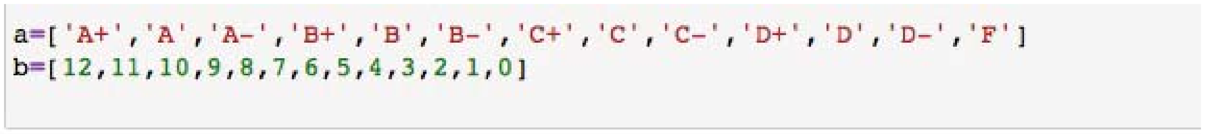
100. As we know, a value in a dictionary can be any Python object, including a dictionary. Here is an example:



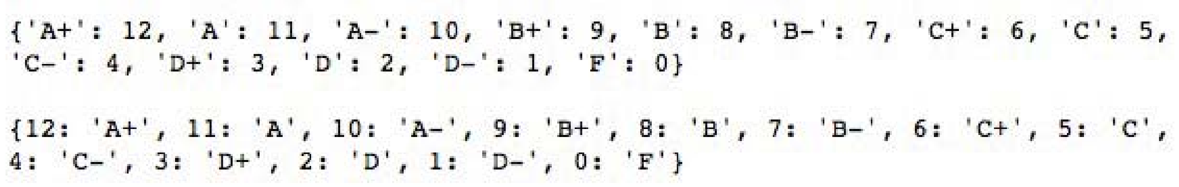
```
employees = {
     'employee 1': {'name': 'Kim', 'salary': 3200},
     'employee 2': {'name': 'Kourtney', 'salary': 5700},
     'employee 3': {'name': 'Khloe', 'salary': 4200}}
```

Write a code to change Khloe's salary to 8800. Write a code to change Kourtney's name to Courtney. Write a code to find the sum of the salaries of the three employees.

101. Given two lists, a and b, create the two dictionaries shown.



```
a=['A+','A','A-','B+','B','B-','C+','C','C-','D+','D','D-','F']
b=[12,11,10,9,8,7,6,5,4,3,2,1,0]
```



```
{'A+': 12, 'A': 11, 'A-': 10, 'B+': 9, 'B': 8, 'B-': 7, 'C+': 6, 'C': 5,
'C-': 4, 'D+': 3, 'D': 2, 'D-': 1, 'F': 0}

{12: 'A+', 11: 'A', 10: 'A-', 9: 'B+', 8: 'B', 7: 'B-', 6: 'C+', 5: 'C',
4: 'C-', 3: 'D+', 2: 'D', 1: 'D-', 0: 'F'}
```

102. Given dictionaries, d1 and d2, create a new dictionary d3 with the following properties:

For each entry (a, b) in d1, if a is not a key of d2 then add (a,b) to d3

For each entry (a, b) in d2, if a is not a key of d1 then add (a,b) to d3

So if d1 is {2:3, 8:19, 6:4, 5:12} and d2 is {2:5, 4:3, 3:9} then the resulting dictionary d3 is {8: 19, 6: 4, 5: 12, 4: 3, 3: 9}.

103. Create a dictionary whose keys are the numbers $1, 1.1, 1.2, 1.3, \ldots, 1.9, 2$ and the values are the squares of the keys.

104. Write a code for the following. Input is a number between 20 and 99. Output is that number spelled out in words. So, input: 21, output: twenty one; input: 68, output: sixty eight.

105. Write a code for the inverse of the previous problem. I.e., input is a number between twenty and ninety nine, spelled in words, and the output is the same number written using digits 0,1,2,...,9.

106. Write a code for the following: Input is a number between 1 and 999. Output is the number spelled out in words. So, input: 15, output: fifteen; input: 608, output: six hundred eight.

EXTRA PRACTICE PROBLEMS, STRINGS AND LISTS (107-111)

107. Write a function rearrange(s) which rearranges elements in a string so that letters come first, followed by digits, in the order as in the given string. For instance, if s='02ab67gj7y8ff', then rearrange(s) should return the string 'abgjyff026778'.

108. Write a function zipper(s1,s2) to concatenate two equal length strings in a 'zipper' pattern, i.e., if s1='ABCDE' and s2='12345' then zipper(s1,s2) should return the string 'A1B2C3D4E5'.

109. Create a function remove(s) where s is a string which contains letters of the alphabet, and at most two of the four brackets/parentheses symbols: (,), [, or]. For instance, s='su]n[', s='ra()in', or s='cloud'. The function remove(s) should return the string without the bracket or parenthesis.

Thus, remove('su]n[')) should return 'sun', remove(']snow')) should return 'snow', remove('cloudy skies')) should return 'cloudy skies', and remove('ra(in')) should return 'rain'.

110. Write a function make_unique(numbers) that takes a list (called numbers) and returns a new list (call it unique) which removes all duplicates; i.e., the list unique contains only one copy of each element which appears in numbers. The list unique should keep the order as in the original lists (meaning that you cannot use set(numbers) command).

For example: make_unique([1,3,22,5,1,3,3,400,1,6,400,6,5,7,5]) should return [1,3,22,5,400,6,7].

111. Write a function called multiple_app that takes a list (call it numbers) and returns a new list (call it multiple) which contains all elements in numbers that appear more than once. The list multiple should contain each element only once.

For example, multiple_app([1,2,33,4,512,2,33,4,512,6,2,8]) should return [2,33,4,512], and not [2,33,4,512,2,33,4,512,2].

112. Create a one-dimensional numpy array

$$[\underbrace{0,0,\ldots,0}_{m},\underbrace{1,1,\ldots,1}_{m},\ldots,\underbrace{20,20,\ldots,20}_{m}]$$

where $m \geq 1$.

113. Write a code that will create the array cos, which contains the values of $\cos x$ for $x = 0, 0.01, 0.02, 0.03, \ldots, 3.14$.

114. Create the numpy array x=[0 0.0.15, 0.3, 0.45 ... 14], and write a code for the function $f(x) = x^3 \sin(1.7x) - 4x^2 \ln(0.43 + x) - 1.9$. Then create the array fofx such that fofx[i] is equal to the value of $f(x)$ at x[i]. (In other words, compute $f(x)$ at all values of the array x).

115. Write code for each of the following; v and w are one-dimensional numpy arrays of shape (n,).

(a) L^p norm of v, defined by

$$\|v\|_p = (|v_1|^p + |v_2|^p + \cdots + |v_n|^p)^{1/p}$$

(b) L^∞ norm of v, defined by

$$\|v\|_\infty = \max(|v_1|, |v_2|, \ldots, |v_n|,)$$

(c) inner product

$$\langle v, w \rangle = \sum_{i=1}^{n} \frac{v_i w_i}{2^i}$$

116. Write a function almost_equal(v,w,tol) where v and w are vectors (one-dimensional arrays of the same length), and the default value for the tolerance tol is 10^{-5}. The function almost_equal(v,w,tol) returns True if the corresponding components in v and w are within tol of each other, i.e., if $|v_i - w_i| <$ tol for all i.

117. Without using np.any and np.all commands, write code for the following:

(a) A is an array of shape (m,n), and its elements are positive integers. Write a function which returns True if all entries in A are even.

(b) B is an array of shape (m,n), and its elements are positive integers. Write a function which returns True if there is an entry in B which is divisible by 11.

(c) C is an array of shape (m,n) and type float. Write a function which returns True if all entries in C are larger than 7 and smaller than or equal to 47.

(d) Answer questions (a) to (c) by using np.any or np.all commands, as appropriate.

118. Given is an array A of shape (m,n), i.e., A is an $m \times n$ matrix.

(a) Create a one-dimensional array sum_row of appropriate size, defined by sum_row[i] = sum of elements in the i-th row of A.

(b) Create a one-dimensional array sum_col of appropriate size, defined by sum_col[i] = sum of the squares of elements in the i-th column of A.

(c) Create a one-dimensional array max_row of appropriate size, defined by max_row[i] = largest element in the i-th row of A.

(d) Create a one-dimensional array avg_col of appropriate size, defined by avg_col[i] = average (mean) of the elements in the i-th column of A.

119. A and B are two-dimensional numpy arrays of shape (m,m); i.e., A and B are square matrices of size m. Write a code for:

(a) L^p norm of A, defined by

$$\|A\|_p = \left(\sum_{i,j=1}^{m} |a_{ij}|^p \right)^{1/p}$$

(b) L^∞ norm of A, defined by

$$\|A\|_\infty = \max_{0 \le i,j \le n} |a_{ij}|,$$

(c) inner product

$$\langle A, B \rangle = \sum_{i,j=1}^{m} a_{ij} b_{ij}$$

(d) product defined by

$$\langle A, B \rangle = \sum_{i,j=1}^{m} \frac{a_{ij} b_{ij}}{2^i 2^j}$$

120. Create each of the following numpy arrays.

(a) An $m \times m$ matrix whose diagonal entries are $1, 2, 3, \ldots, m$, and all off-diagonal entries are zero.

(b) An $m \times m$ matrix (with $m > 2$) whose border entries are 5, and all other entries are 1 (border entries in a matrix are the top and the bottom rows and the first and the last columns).

(c) An $m \times m$ matrix whose diagonal entries, and all entries above the diagonal are 3, and all entries below the diagonal are zero (this is an example of an upper triangular matrix).

(d) An $m \times m$ matrix whose entry at a location (i,j) is the remainder when $i + j$ is divided by 3.

(e) An $m \times m$ matrix whose entry at a location (i,j) is the larger of the two numbers i and j (if i is equal to j then the entry is either i or j).

121. Consider the matrix

$$\mathrm{a} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{bmatrix}$$

Say what each of the following commands does, and then check using Python.

(a) print(a[2,3]) (b) print(a[0,0]) (c) print(a[2:,])
(d) print(a[:,0]) (e) print(a[:,0:2]) (f) print(a[2:4,0:2])
(g) print(a[2:4,1:2]) (h) print(a[::2,]) (i) print(a[0:2,::3])

122. Given is an $m \times m$ matrix A. Use slicing to do the following.

(a) Extract the i-th row of A, $0 \leq i \leq m$.

(b) Extract the j-th column of A, $0 \leq j \leq m$.

(c) Assume that $m = 3$. Extract the 2×2 matrix C_{00} which is obtained from A by removing the first row and the first column.

(d) Assume that $m = 3$. Extract the 2×2 matrix C_{01} which is obtained from A by removing the first row and the second column.

(e) Assume that $m = 3$. Extract the 2×2 matrix C_{02} which is obtained from A by removing the first row and the third column.

123. Write a function which removes the i-th row and the j-th column from a matrix A.

124. Write a function which computes the determinant of a 3×3 matrix using cofactors.

125. Write a function which computes the determinant of a 4×4 matrix using cofactors.

126. Create a numpy array which contains the values of $g(x) = e^{-0.2x}$ for $x = 0, 0.001, 0.002, 0.003, \ldots, 4$.

127. Create a numpy array (float) which contains the values of the function $f(x) = \sin(2x) - 3\cos(x/5)$ at 150 equally spaced values between 0 and $\pi/2$ (150 does not include 0 and $\pi/2$).

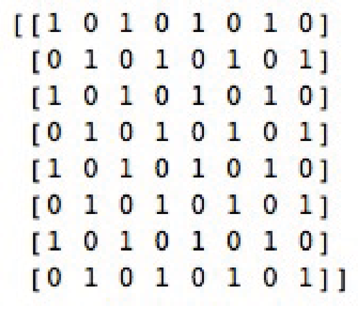
128. Create a numpy array which contains the values of the function $f(x) = \sin(x)$ for $x = 0, \pi/250, 2\pi/250, 3\pi/250, \ldots, 2\pi$.

129. Find the mean of the squares n^2, for $n = 1, 2, \ldots, 100$.

130. Find the sum $\sin(1/1) + \sin(1/3) + \sin(1/5) + \cdots + \sin(1/105)$ using a suitably created numpy array.

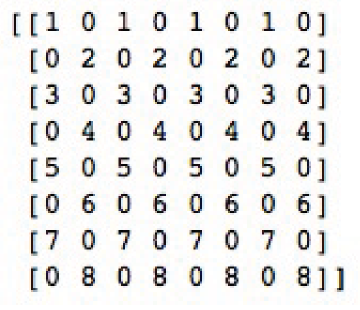
131. Add a border to zeros around a given $m \times n$ numpy array A; i.e., create an $(m+2) \times (n+2)$ array whose first and last rows and first and last columns are zero, and A is inside.

132. Write Python code to generate this 8×8 numpy array:



```
[[1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]
 [1 0 1 0 1 0 1 0]
 [0 1 0 1 0 1 0 1]]
```

133. Write Python code to generate this 8×8 numpy array:



```
[[1 0 1 0 1 0 1 0]
 [0 2 0 2 0 2 0 2]
 [3 0 3 0 3 0 3 0]
 [0 4 0 4 0 4 0 4]
 [5 0 5 0 5 0 5 0]
 [0 6 0 6 0 6 0 6]
 [7 0 7 0 7 0 7 0]
 [0 8 0 8 0 8 0 8]]
```

134. Write a function matrix_ab(m,a,b) which returns an $m \times m$ matrix M (as a numpy array) whose entries are defined by $M_{ii} = a$ and $M_{ij} = b$ when $i \neq j$.

135. Given are two one-dimensional arrays A and B. Create the array C which contains the positions where A and B match.

Thus, if A=[1,2,3,4,5,6,7,8,9] and B=[0,2,0,0,5,4,3,2,9,11,12,14] then C=[1,4,8].

136. Given are two one-dimensional arrays A and B. Create the array C which is the 'Cartesian product' of A and B.

Thus, if A=[1,2] and B=[13,14,15] then C=[[1,13],[2,13],[1,14],[2,14], [1,15],[2,15]].

137. Write a function create__array(A,m,n) which does what reshape does: A is a list of $m \cdot n$ numbers, where m and n are positive integers. The function returns an $m \times n$ numpy array M, where the first m entries in A form the first row of M, the next m entries in A form the second row of M, etc.

138. Given is a list of numbers L. Write a function create_array(L) which creates a square matrix whose diagonal entries are the elements in L, and all other entries are zero. For example, if L=[-4,6,1.2,9], then create_array(L) should return

$$\mathrm{M} = \begin{bmatrix} -4 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 \\ 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 9 \end{bmatrix}$$

139. Given is a numpy array L=np.array([-2,-3,-6,9,0,8,7,6,9]). In each case, state what the output is, and then check by typing code into Jupyter.

(a) print(L>0)

(b) print(L<=0)

(c) print(L%2==0)

(d) print(L**2>0)

(e) print(np.logical__or(-1<L,L>7))

(f) print(np.logical__and(-1<L,L>7))

(g) print(np.logical__not(-1<L)

140. Consider numpy arrays A=np.arange(1,11) and B=np.arange(-6,13,2). In each case, state what the output is, and then check by typing code into Jupyter.

(a) print(A>B)

(b) print(A>B**2)

(c) print(abs(A-B)!=0)

(d) print(A-B==4)

141. Consider numpy arrays A=np.arange(1,11) and B=np.arange(-6,13,2). In each case, state what the output is, and then check by typing code into Jupyter. Add print(A,B) to your code.

(a) A[A> 4]=55

(b) A[B> 4]=55

(c) A[A==B]=333

(d) A[abs(A-B)<=2]=17

142. Consider the numpy array M=np.arange(-5,7).reshape(3,4). In each case, state what the output is, and then check by typing code into Jupyter.

(a) print(M<=2)

(b) print(M**2<=4)

(c) print(np.logical_or(-2<M,M>2))

(d) print(np.logical_not(M>2))

143. Consider the numpy array M=np.arange(-5,7).reshape(3,4). In each case, state what the output is, and then check by typing code into Jupyter.

(a) print(M[M>4])

(b) print(M[M==4])

(c) print(M[2*M-5>0])

(d) print(M[np.exp(M)>1])

144. Consider the numpy array M=np.arange(-5,7).reshape(3,4). In each case, say what the command does, and then check in Jupyter by adding print(M) to the code.

(a) M[M>0]=1

(b) M[M>= 0]=-6

(c) M[M==0]=100

(d) M[M%2==1]=77

145. Given a non-zero $m \times n$ matrix M of type float, create a logical array L, such that

(a) L[i,j]=True if M[i,j] is positive or zero, and False otherwise.

(a) L[i,j]=True if M[i,j] is larger than the mean of all elements in M, and False otherwise.

(c) L[i,j]=True if M[i,j] is the largest term in its row, and False otherwise.

146. Given a non-zero $m \times n$ matrix M of type float.

(a) Create a list of all elements of M which are larger than 4.

(b) Create a list of all elements of M which are larger than the mean of M.

(c) Replace all elements of M which are larger than 5 by zero.

(d) Replace all elements of M which are larger than the mean of M by the mean of M.